How to Obscure Any URL

How Spammers And Scammers Hide and Confuse

Last Updated Sunday, 13 January 2002

***NOTICE:** the IP address of this site has changed of late, and I've been unable to set aside time for the rather **large** task of revising this page. Its numerous links to the old IP address won't work. It'll be updated soon!*

Since this page was first written in 1999, Internet Explorer and Netscape have both begun dealing with URLs differently, particularly in versions 6 and above. Some of the examples here will no longer work with those browser versions.

The URL (Universal Resource Locator) of the page you are now viewing is http://www.pc-help.org/obscure.htm.

It is also http://3468664375@3468664375/o%62s%63ur%65%2e%68t%6D. Go ahead and click on that link. It'll take you right back to this very page.

The weird-looking address above takes advantage of several things many people don't know about the structure of a valid URL.

There's a little more to Internet addressing than commonly meets the eye; there are conventions which allow for some interesting variations in how an Internet address is expressed.

These tricks are known to the spammers and scammers, and they're used freely in unsolicited mails. You'll also see them in ad-related URLs and occasionally on web pages where the writer hopes to avoid recognition of a linked address for whatever reason. Now, I'm making these tricks known to *you*. Read on, and you'll soon be very hard to fool.

(Note: Depending on your browser type and its version, some of the oddly-formatted URLs on this page may not work. Also if you're on a LAN and using a proxy [gateway] for Internet access, many of them are unlikely to work. Also, fear not; this page does not *exploit the "Dotless IP Address" vulnerability of some IE versions.)*

How It's Done

Here it is again: http://3468664375@3468664375/o%62s%63ur%65%2e%68t%6D

First take note of the "@" symbol that appears amid all those numbers. In actual fact, everything between "**http://**" and "@" is completely irrelevant! Just about **anything** can go in there and it makes no difference whatsoever to the final result. Here are two examples:

http://doesn'tmatter@www.pc-help.org/obscure.htm

http://!$^&*()_+`-={}|[]:;@www.pc-help.org/obscure.htm

Go ahead and use the links. If they work at all with your browser, you'll be back to this page again.

This feature is actually used for authentication. If a login name and/or password is required to access a web page, it can be included here and login will be automatic.

Example: **http://username:password@www.whatever.com/secret/eyesonly.htm**

But if the page requires no authentication, the authentication text is in effect ignored by both browser and server.

This presents interesting possibilities for confusing the unsuspecting user. How about this one:

http://www.playboy.com@3468664375/obscure.htm

If you didn't know better, you might think this page were at playboy.com!

By the way, the @ symbol can be represented by its hex code %40 to further confuse things; this works for the IE browser, but not for Netscape. *(Thanks to The Webskulker for this.)*

All right, so what about that long number after the "@"? How does **3468664375** get you to **www.pc-help.org**?

In actual fact, the two are **equivalent to one another**. This takes a little explaining so follow me carefully here.

The first thing you need to know (most Net users know this), is that Internet names translate to numbers called IP addresses. An IP address is normally seen in "dotted decimal" format. **www.pc-help.org** translates to **206.191.158.55**. So of course, this page's address can be expressed as: http://206.191.158.55/obscure.htm.

Numeric IP addresses are generally unrecognizable to people, and not easily rememberd. That's why we use names for network locations in the first place.

Merely using an IP address, in its usual dotted-decimal format, in place of the name is commonly done and can be quite effective at leaving the human reader in the dark about which website he's visiting.

But there are *other ways* to express *that same number*. The alternate formats are:

- "dword" - meaning **double word** because it consists essentially of two binary "words" of 16 bits; but it is expressed in decimal (base 10);
- "octal", meaning it's expressed in base 8; and
- "hexadecimal" hexa=6 + deci=10 (base 16).

The dword equivalent of **206.191.158.55** is **3468664375**. Its octal and hexadecimal equivalents are also illustrated below.

Why obscure names in the first place? Most often it's because by publicly-available registration records, *the owners of domain names can often be identified*. Even if the owner isn't traceable by that record, his service provider *is*. The last thing any scammer or spammer wants is to be found by his victims, or to have his service provider alerted to his abuses. Although the use of obscured URLs is far from their only means of avoiding retribution, it's been a favorite.

Below, I explain how you can get an IP address for any name, how to convert a dotted-decimal IP address to the dword format, and how the octal and hex formats work. If you know how it's *done*, you will also know how it's *undone*.

Okay, so what about the rest of the URL? Let's look yet again at that weird address I first showed you:

http://3468664375@3468664375/o%62s%63ur%65%2e%68t%6D

It's beginning to make ***some*** sense, isn't it? But what's all that gibberish on the right? Here's how that works:

Individual characters of a URL's path and filename can be represented by their **numbers** in **hexadecimal** form. Each hex number is preceded by a "%" symbol to identify the following two numbers/letters as a hexadecimal representation of the character. The practical use for this is to make it possible to include spaces and unusual characters in a URL. But it works for **all** characters and can render perfectly readable text into a complete hash.

In my example, I have interspersed hex representations with the real letters of the URL. It simply spells out "/obscure.htm" in the final analysis:

/ o %62 s %63 ur %65 %2e %68 t %6D
/ o b s c ur e . h t m

The letters used in the hex numbers can be either upper or lower case. The "slashes" in the address cannot be represented in hex; nor can the IP address be rendered this particular way. But everything else can be.

Hexadecimal Character Codes

Hex character codes are simply the hexadecimal (base 16) numbers for the ASCII character set; that is, the number-to-letter representations which comprise virtually all computer text.

To find the numeric value for an ASCII character, I often use a little batchfile I wrote for the purpose years ago; and then if I want the hex equivalent I usually do the math in my head. It just requires familiarity with the multiples of 16 up to 256.

For most people, the conversion is probably best done with a chart. The best ASCII-to-hex chart I have ever seen is on the website of Jim Price: http://www.jimprice.com/jim-asc.htm. Jim explains the ASCII character set wonderfully well, and provides a wealth of handy charts.

I can't improve on Jim's excellent work! Print out Jim's ASCII-to-hex chart and you're in business. If Jim's site ever disappears, let me know and I'll do a chart of my own.

IP Addresses

IP addresses are most commonly written in the dotted-decimal format. A dotted-decimal IP number normally has 4 numeric segments, each separated by a period. The numbers must range from 0 to 255.

Translation of a network name to its IP address is usually done in the background by your network software, invisible to the user. Given a name, your browser queries a **name server**, a machine somewhere on the Net which performs this basic network addressing function; it thereby obtains the numeric IP address and then uses that address to direct its requests to the right computer, somewhere out there on the Net.

There is a standard utility which allows the user to perform these name server lookups directly and see the results. It's called **NSLOOKUP**.

A wide variety of nslookup utilites is available on the Net, often for free download. Some provide a graphical interface under Windows, but the original and most basic nslookup is run from a textual command line. One such command-line utility is included in my free Network Tracer. Please download it if you're interested.

Place **NSLOOKUP.EXE** in your Windows directory and you can use it from a DOS window. A simple nslookup query is structured as follows:

nslookup [name or IP address] [name server]

A name server has to be specified if you're using Windows 9x/ME, either by name or IP address. Find out the address of your

ISP's Primary DNS Server -- it can usually be found in your Dial-Up Networking setup or in the documents provided for setup of your Internet connection.

If you're using XP or NT, the name server need not be specified.

A valid query for my ISP's web server address would be:

nslookup www.nwi.net [name server]

Here's what that command puts out in response, with my comments:

```
nslookup www.nwi.net 198.41.0.196 <-- Here's the command you typed in

Server:  ns.netsol.com          <-- The name server you utilized

Address:  198.41.0.196          <-- The responding name server's IP address


Non-authoritative answer:       <-- Some other name server is the source of the data

Name:    sundance.nwinternet.com  <-- The "real" name of www.nwi.net

Address:  206.159.40.2          <-- What you came here for: the IP address of www.nwi.net

Aliases:  www.nwi.net           <-- www.nwi.net is an alias -- not the

                                    primary name given to that address, but a valid one.
```

It's a powerful utility; it can find names for known addresses, addresses for known names, and a variety of other information relevant to an Internet address. But doing some of the fancier things with **NSLOOKUP** is difficult if you're not already technically savvy. For the technically inclined, there is a manual; and several examples of its use can be found in TRACE.BAT, the primary component of my Network Tracer.

If you're determined to avoid the DOS command line, and want a tool that will do most of the thinking for you, I recommend NetScanTools, a reasonably-priced network utility toolbox. It's available as a 30-day shareware demo and a bargain at just $25. NetScanTools is not merely an address-lookup utility; it can do a great many things. For a Windows user trying to comprehend the nuts and bolts of the Net, it's a whole world of discovery.

You can also do your name server lookups with a web browser. There are nslookup "gateways" scattered around on the Web. One such gateway is at: http://www.lasaltech.com/cgi-bin/nslookup Another is: http://www.interlog.com/~patrick/cgi/nslookup.cgi

A Variation on Dotted-Decimal IPs

If you're using Internet Explorer, this address should work (It doesn't work with at least some versions of Netscape): http://462.447.414.311/obscure.htm

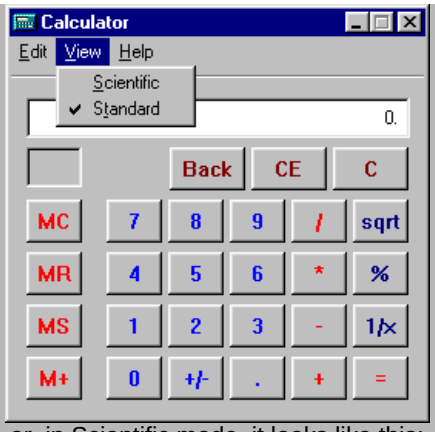
Normally, the four IP numbers in a standard dotted-decimal address will all be between 0 and 255. In fact they must translate to an 8-bit binary number (ones and zeroes), which can represent a quantity no higher than 255.

But the way this number is handled by some software often allows for a value higher than 255. The program uses only the 8 right-hand digits of the binary number, and will drop the rest if the number is too large.

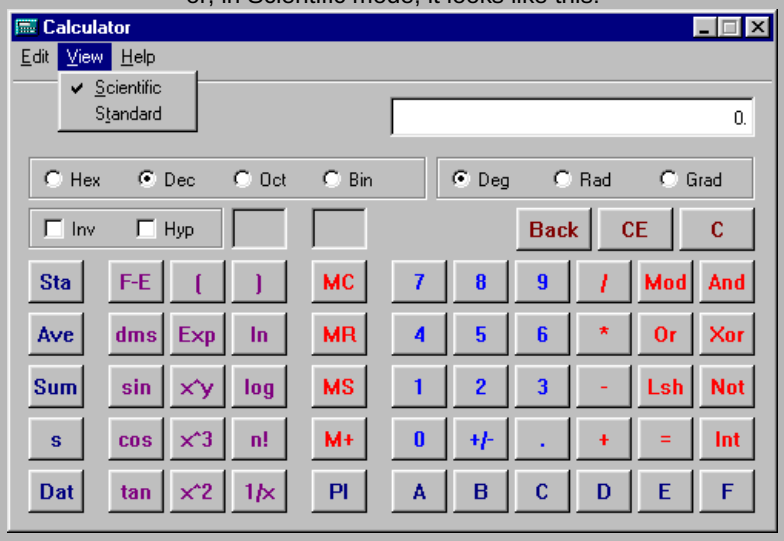
This means you can add multiples of 256 to any or all of the 4 segments of an IP address, and it will often still work. In my tests, it was limited to 3 digits per number; values over 999 didn't work.

Converting An IP Address to Dword Format

I could create a math lesson about this, and tell you all about bits and bytes and base 16. But it's not really necessary. Anyone with a Windows system has a handy calculator that makes it simple to convert decimal numbers to hex, and to find the dword equivalent of any dotted-decimal IP number. You should find it by selecting Start ... Programs ... Accessories ... Calculator. It will look like this:



or, in Scientific mode, it looks like this:



I suggest Scientific mode for this purpose.

Start with an IP address. In this example we'll use 206.191.158.55. Enter the following keystrokes into the calculator exactly as shown:

206 * 256 + 191 = * 256 + 158 = * 256 + 55 =

The dword equivalent of the IP address will be the result. In this case, 3468664375.

Now, there is a further step that can make this address even*more* obscure. You can add to this dword number, **any multiple of the quantity 4294967296** (256^4) -- and it will still work. This is because when the sum is converted to its basic digital form, the *last 8 hexadecimal digits will remain the same*. Everything to the left of those 8 hex digits is discarded by the IP software and therefore irrelevant.

Thus, the following URLs will also lead to this page:

http://7763631671/obscure.htm
http://16353566263/obscure.htm
http://235396898359/obscure.htm

There now exist a handful of utilities that will do dword (and other) conversions of IP addresses and URLs. When time permits, I'll be sure to list them on this page. Meanwhile, there's a handy script on Matthias Fichtner's website which will quickly convert any IP address to its dword value and vice-versa: http://www.fichtner.net/tools/ip2dword/.

PING

The PING utility that's in every Windows system can decipher dword IPs. In fact, it deals with **every method of expressing an IP address** that's described on this page. *(My thanks to Steven, who pointed this out on the NTBugTraq list.)*

Just open a DOS window and type:

ping [IPAddress]

PING will then do its usual job, in which it contacts the remote system (if any) at that address and gauges its response times. In the process, it displays the ordinary dotted-decimal equivalent of the IP address you entered.

Octal IP Addresses

As if all this weren't enough, an IP address can also be represented in **octal** form -- base 8.

The URL for this page with its IP address in octal form looks like this: http://0316.0277.0236.067/obscure.htm
Go ahead, try it. You'll be right back here once again..

Note the leading zeroes. They're necessary to convey to your browser the fact that this is an octal number.

Any number of leading zeroes can be added to any or all of the numbers in the address. For example:
http://00000000316.000277.00000236.00000000067/obscure.htm

Naturally, arbitrary authentication text can also be added to an octal address. Example: http://www.sleazy-ad.com@00000000316.000277.00000236.00000000067/obscure.htm

Octal numbers are easily derived with the Windows calculator in Scientific mode. Enter a decimal number, then select the "Oct" button at upper left. The octal number will appear. The reverse operation translates octal to decimal.

(Those who find all this unwieldy can always use the handy URLomatic at www.samspade.org. It will reveal the dotted-decimal IP address of a dword- or octal-formatted URL, as well as to decode hex character codes. This link to the URLomatic will completely decipher my original example address. Many thanks to Dan Renner of R&B Computerhelp).

Hexadecimal IP Addresses

You thought that was all? Well, so did I, until one Daniel Doèekal informed me otherwise. There is yet *another* obscure way to express an IP address.

Starting with the method outlined above, you can readily calculate the *hexadecimal* number for 206.191.158. 55. In Scientific mode, calculate the dword value. Then select the Hex button. The resulting hexadecimal number (CEBF9E37) can be expressed as an IP address in this manner: 0xCE.0xBF.0x9E.0x37

The "0x" designates each number as a hex quantity.

The dots can be omitted, and the entire hex number preceded by 0x:0xCEBF9E37

And, additional arbitrary hex digits can be added to the left of the "real" number0x9A3F0800CEBF9E37

Some browsers (Netscape 3.x and 4.x for instance) won't work with hex IPs; but for IE users (prior to version 6), this page's URL can be:
http://0xCE.0xBF.0x9E.0x37/obscure.htm
or:
http://0xCeBF9e37/obscure.htm
or:
http://0x9A3F0800CEBF9E37/obscure.htm

It's Not Over Yet

Ah, you thought you had it all nailed down? Well, it's mix-and-match time!

Believe it or not, the following URL, which uses hex, decimal *and* octal numbers in the IP address, actually *works*:
http://0xCE.191.0236.0x37/obscure.htm

This mixed-format address also works with bogus authentication text: http://spam-world.net@0xCE.191.0236.0x37/obscure.htm

For Netscape users, omitting the hex is necessary; but decimal and octal can be mixed: http://spam-world.net@0316.191.0236.067/obscure.htm

Thankfully, the wonderful URLomatic at samspade.org deals with these mixed-up IPs just fine.

Also, don't forget PING. It will also decode these mixed-format addresses.

Not Dotless, But Less Dots

A variation on the dword IP address is one where a *portion* of the IP address is similarly converted. This only works with the rightmost two or three numbers, not the leftmost.

Let's start again with 206.191.158.55. Leaving the "206" as it is, we do the same calculation with the last three numbers:

191 * 256 + 158 = * 256 + 55 = 12557879

Now we have: http://206.12557879/obscure.htm

158 * 256 + 55 = 40503

Which results in: http://206.191.40503/obscure.htm

Now extend this same concept to hex and octal numbers:

http://0xCE.0xBF9E37/obscure.htm
http://0xCE.0xBF.0x9E37/obscure.htm
http://0316.057717067/obscure.htm
http://0316.0277.0117067/obscure.htm

Why not mix and match?

http://0316.0xBF9E37/obscure.htm
http://206.0277.0x9E37/obscure.htm

Furthermore...

Internet Explorer (versions prior to 6) will allow the characters of the IP address itself, in any format, to be expressed as hex-coded characters.

Example: http://%334%368%366%34%33%375/obscure.htm

The same can be done with the domain name: http://%70%43%2d%68%45%6c%50%2e%6f%52%67/obscure.htm

In Sum

URLs can be obscured at least three ways:

1. Meaningless or deceptive text can be added after "**http://**" and before an "@" symbol.
2. The domain name can be expressed as an IP address, in dotted-decimal, dword, octal *or* hexadecimal format; and all of these formats have variants.
3. Characters in the URL can expressed as hexadecimal (base 16) numbers.

An Increasingly Common Exception

As IP address space becomes more valuable, web hosting services increasingly use systems that place *many* websites at *one* IP address. The server differentiates between sites by means of the domain name portion of the URL. Sites on such a server cannot be addressed using the IP address alone.

Some Notes on Compatibility

I've been getting a lot of feedback about this page from people who are running various browsers and proxies. So far, reports and my own rather limited tests seem to indicate that:

- hex-coded IPs and values over 255 in dotted-decimal IPs don't work with Netscape;
- most, perhaps all of the dword-coded IPs don't work with some versions of IE; this could be an effect of the MS patch for the "dotless IP" exploit.
- Later IE versions seem to reject any hex-coded IP that's not broken up by dots as in my first example above;
- Opera 3.60 doesn't allow non-dotted hexadecimal IPs.
- Netscape won't allow the following characters in the authentication text: /?
- IE won't allow the following characters in the authentication text: /\#
  and it exhibits problems or inconsistencies with: %'"<>
- MS-Proxy reportedly rejects almost any IP address that's not in dotted-decimal IP format, as may some other proxies. Reports indicate that most proxies handle them all just fine.

If you notice anything more you think I should know about URL formats or the behavior of some particular software, feel free to drop me a note.
