

Indecent Exposure—The Dangers of Outside Networks
May 27, 2006



Feed address for Podcast subscription:
<http://feeds.feedburner.com/EdZollarsTaxUpdate>
Home page for Podcast: <http://ezollars.libsyn.com>
©2006 Edward K. Zollars, CPA

The TaxUpdate podcast is intended for tax professionals and is not designed for those not skilled in independent tax research. All readers and listeners are expected to do their own research to confirm items raised in this presentation before relying upon the positions presented.

The Podcast and this document may be reproduced freely so long as no fee is charged for the use of this document. Such prohibited use would include using this podcast or document as part of a CPE presentation for which a fee is charged.

This podcast is sponsored by Leimberg Information Services, located on the web at <http://www.leimbergservices.com>. Leimberg Information Services offers email newsletters on tax related matters, as well as access to a library of useful information to tax practitioners that subscribe to their services.

Security of Internet Data

This week we'll take a brief break from tax specific topics to deal with the issue of the security of data, especially when we take our laptops out on the road and make use of wireless hotspots and hotel high speed networks that have become very prevalent. These connection mechanisms contain risks that might not be obvious to many users.

Much of this week's discussion originated from podcasts that have been created over the past few months by Steve Gibson and Leo LaPorte on the *Security Now* podcasts that I've recommended as a source of information for those interested in these topics. However, I thought this topic was important enough, due to our obligations to our clients to preserve the confidential nature of their information, to discuss here with a suggestion you go and retrieve their more detailed podcasts, which can be found at:

<http://www.grc.com/SecurityNow.htm>

Episode 29 is the one that contains most of this information.

A very useful, and more detailed, outline of the risk we're covering this week can be found at:

<http://www.grc.com/nat/arp.htm>

Trust and the Local Area Network

When you hook your computer up to an ethernet network, you may not be aware that your system implicitly trusts (and has to trust) virtually all of other devices on that network in order to function. A key reason for this is that every device on your network ends up with two addresses—it's assigned "IP address" which can and does change as your machine logs into different networks, and the unique MAC address your machine has (which never changes).

As well, depending on how the network gets data to the machines on it, you may find that everything going to and from your machine also is seen by every other machine on the network.

Network Addressing

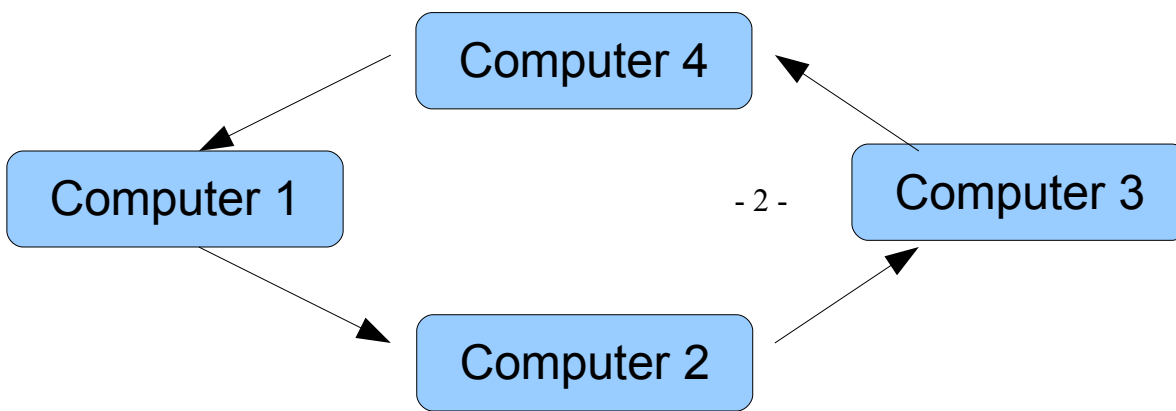
Why two addresses? Well, because the Internet address (IP address) is a hierarchical address that helps systems direct data to where it needs to go, while the specific unique MAC address assigned to each device makes sure that your machine is uniquely identified. By associating the currently assigned IP address to your unique MAC address, we make sure that data can be sent from any computer connected to the Internet and make it directly to your machine.

Your IP address is that "dotted" address you see referred to from time to time. For instance, my machine right now reports an address of 192.168.1.109—the address my home router assigned to the machine when I logged back into it today. Earlier in the day it had a very different address assigned to it when I was logged into Verizon Wireless's EVDO network while attending the Phoenix Tax Workshop's meeting.

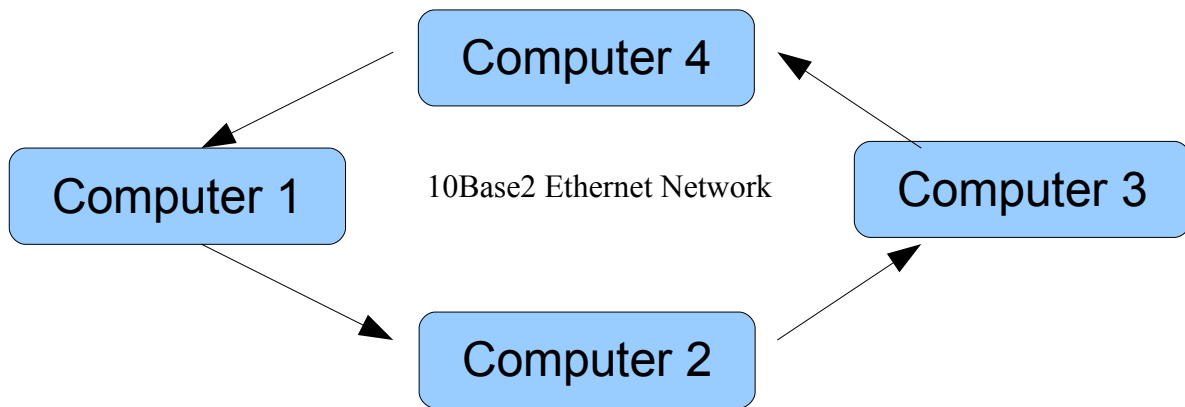
Each segment of the address gives information about how to move the data in the direction of this machine to get it to the local network. Once it's on the local network, my MAC address is used to finally get the data into my machine.

Ethernet

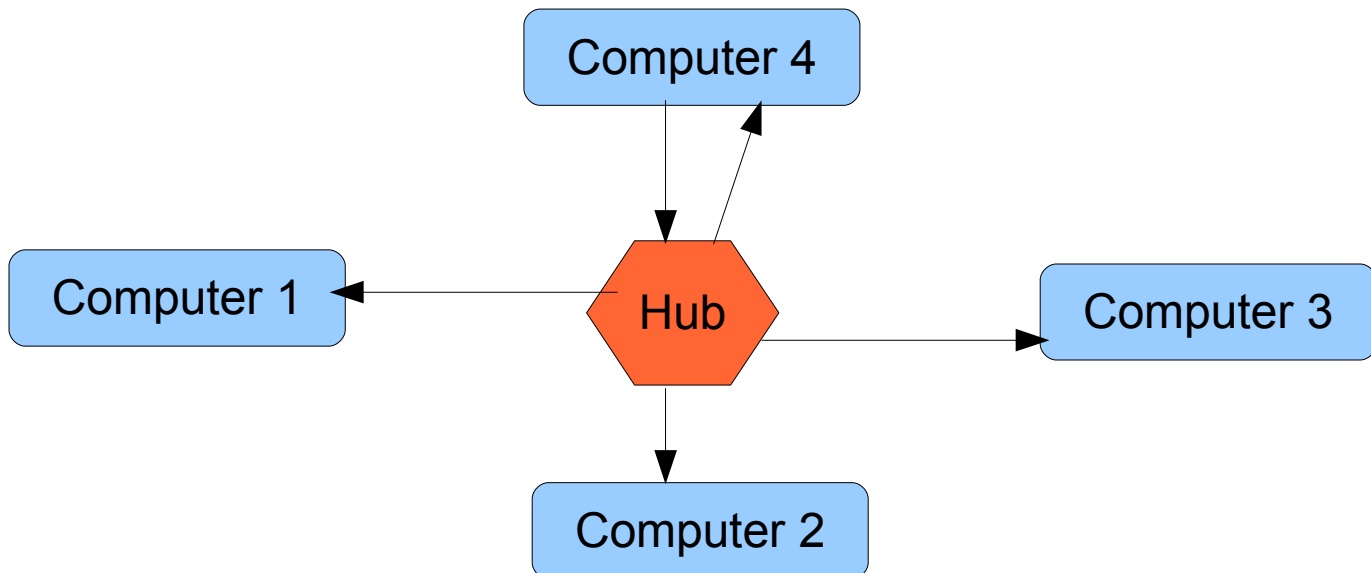
On the local network, originally all data going through the network was sent to all machines on the network, with each machine checking to see if it was the machine that



was supposed to grab the data by looking at its MAC address. Originally the wiring made that a given—machines were put along one long wire over which data passed (the old 10Base-2 coaxial cable you all may fondly remember). That wiring had another problem—if any part of that wire was damaged, your network went dead (kind of like the old Christmas tree light strings where when one light goes out the entire string goes dead).



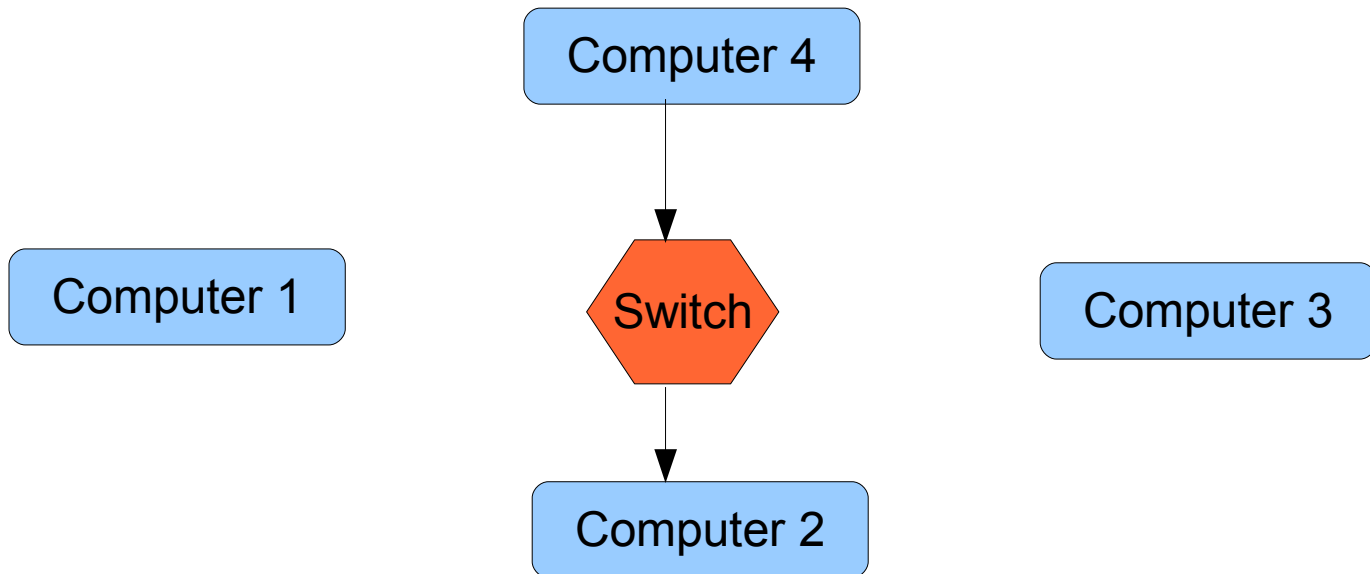
Eventually we went to a more reliable system—each machine running a separate wire to a central hub where the data would be broadcast. While that could have allowed the data also to be targeted (we no longer had to send data past everyone), in the initial implementations (known as hubs) that wasn't done—it was simpler to retransmit everything since that required no intelligence in that central location.



Now if one wire breaks, the rest of the network continues to function. But even though computer 4 may have been sending information to Computer 2, all four computers on the network still had the information pass through their cards. That certainly seems to pose

certain risks for data, as well as causing a lot of “noise” for the individual network adapter card to have to sort through.

As intelligence became cheaper to add to devices, wired hubs began to be replaced by wired switch. A switch is similar to a hub except it keeps a table of addresses and sends the data only down the wire for the address of the network card that the data is bound for. So the same conversation for a switch looks like this:



The data bound for computer 2 only went to computer 2, and computers 1 and 3 saw none of it.

Exposures

Well, surely everyone is using switches now so there’s not an issue. Well, not really.

First, Steve Gibson reported on *Security Now* that another security expert he knows has tested hotel networks as she has traveled, and found that nearly half of the high speed hotel networks she has used have wired the room networks using hubs, generally because when the hotels wired up they were cheaper than switches—and they are still working, so no need is seen to change them over.

Second, if you think about it a wireless network can’t really function as a switch—every machine on the network (and any device in radio range) can see every transmission.

Ah, but wireless network transmissions can be scrambled using either WEP or, for newer systems, WPA. Well that’s true—except that every machine on the network has the key and can still look at every piece of traffic going over the network. As well, WEP (the older “security” system for encrypting wireless transmissions) has significant design problems that means that anyone who is interested and does some minimal work

searching on Google will get software that can quickly break the WEP encryption on those wireless networks (WPA is generally secure *if* you have sufficiently strong password—but that still doesn't help for those who are on the network already).

So I'll Stick with Wired Switched Networks...

Well, that one doesn't work either, due to the item described in the article referenced from Steve Gibson's GRC site—the “man in the middle” attack. Steve's article describes the matter in detail, but essentially our problem is the mapping of IP addresses to MAC addresses is handled very informally on our ethernet networks. Machines send out their own mapping, and each machine retains a table of mappings.

Thus, a “bad guy” on the network can send your machine a message claiming to be the gateway to the internet for the network (which will exist on each third party network you hook to). It can then tell the gateway machine that it is your IP address—so all your outbound traffic will be routed through the bad guy's machine. That machine logs your data and then sends it on to the real gateway.

Similarly, when traffic arrives for your machine at the gateway, it will send the data to the bad guy who logs it and then sends it on to you. And, worst of all, in most cases no one will really be aware this is happening. The problem is that the parties are not authenticating that the machine they are talking to is the one they should be. The only real clue would be that if you signed onto a secure website, your browser will likely initially complain about the certificate (the bad guy has to fake that—it's an authentication device). However, if you accept the faked certificate (and most users would do just that) then even your secured transmission is now actually not secured.

Software exists on the internet to “help” this process along for those who wish to do such attacks on networks. Such software has additional “features” that it claims, including being able to break the encryption on a Windows Remote Desktop session running through it.

Solutions?

The only real solution to this is use a method that truly authenticates the machines you are talking with. The best solution is to use VPN software of some sort, either to connect back to your home LAN or a third party VPN to get you out to the Internet.

A VPN creates a “tunnel” that contains what looks like random noise to any machine that intercepts the traffic. That encrypted traffic is only decoded on each end of the transmission. By using a VPN you assure the traffic finally comes out at a location that is confirmed in the protocol.

Setting up a VPN goes beyond what we'll cover today in detail, though there are a number of commercial and open source options for setting up such a structure on your

network. Note, though, that some networks you get on won't work with certain VPN structures, though that's becoming less of an issue given the use of VPNs.

If you just need to set up a simple VPN to get back to a machine, one option is to use the freely available Hamachi service (<http://www.hamachi.cc>). That system helps solve one of the major problems of establishing a VPN back to your home network—getting the router to forward the traffic properly—but using a third party server only to initiate the conversation. Clients are available for Windows, Linux and, recently, MacIntosh OSX Tiger. Using Hamachi you can connect back to shares on your local machine, or use a program like RealVNC to remotely use your computer. You can also configure it to use Windows Remote Desktop, but it's not as simple to do that (due to architectural issues in Windows XP, Hamachi has to run as a service for this to work).

If you simply want to connect back to your machine, GotoMyPC from Citrix is a good option that's relatively easy to set up and will be secure so long as you remember that little bit about what to do if you get a complaint about the certificate. You can find it at <http://www.gotomypc.com>. The offering is commercial, but can't be beat for “quick and easy” solution that is a lot simpler and more secure than using an “exposed” Windows Remote Desktop or RealVNC client. However, it only works with Windows machines.

If you are primarily concerned with getting safely to the Internet to check your mail or the like, a couple of solutions I've worked with are PublicVPN, found at <http://www.publicvpn.com> and Hotspot VPN found at <http://www.hotspotvpn.com>. Both are “pay to play” solutions, but they will give you an authenticated tunnel to a known location. Hotspot, because it uses the same https technology used on secured websites, will also tend to work in situations where other VPNs fail. Both are relatively simple to set up, though I had a few issues with each.