COMPUTER NEWS from the

SEPTEMBER 2014 Volume 2 NO. 9

As found on the web and other sources

Just when you thought everything was fine!

ALERT Serious Security Flaw in USB Drives

Category: Gadgets from "askbobrankin.com".

Undetectable malware can be hidden in any USB flash drive, according to security researchers Karsten Nohl and Jakob Lell. This is very bad news for home users who pass around USB drives, and for corporate IT managers who may have to ban the popular devices from business networks. Read on to learn more about the USB devices, and what you need to do...

Is Malware Lurking in Your USB Gadget?

To demonstrate the vulnerability of USB drives, the researchers wrote some proof-of-concept malware (which we can only hope no one copies) called BadUSB. It is a collection of malicious apps that can modify any software installed from a USB drive on a target computer; completely take over control of an infected PC; and even redirect users' Internet traffic.

Erasing or reformatting the USB drive does not destroy the malware, which hides in the USB device's firmware that controls the drive's basic functions. This previously unknown vulnerability is part of the USB standard's design; as such, it can't be eliminated without re-engineering every USB device.

"These problems can't be patched," says Nohl. "We're exploiting the very way that USB was designed." Noll and Lell plan to demonstrate their code at the BlackHat 2014 conference to be held on August 7th. That will either shine a bright light on the problem, or spawn a cottage industry of hacking USB devices, or both.

We've long known that using USB flash drives can be dangerous, because a virus can be stored as a file on the drive. But any decent anti-virus tool will catch that type of thing. However, standard anti-virus scans can't see or touch the firmware that controls a USB drive's basic input/output functions. Security pros would have to reverse-engineer the firmware of a USB device and know what to look for in order to detect this threat. That would require some specialized expertise and equipment to analyze firmware.

It's Not Just Your Flash Drive...

But wait, the news gets worse: it's not just USB flash drives that are vulnerable. Any USB device, from a mouse or keyboard to a digital camera or smartphone charger, contains firmware with the same exploitable vulnerability. While such devices aren't shared among users as promiscuously as USB flash drives are, it's very possible to pick up an infection from anything that plugs into a USB port.

As Noll says, you must "treat USB devices like hypodermic needles that can't be shared among users." This drags safe computing, safe sex and drug use into the same murky metaphor pool. But it's a real problem that shouldn't be ignored.

The BadUSB demo malware suite can do a lot of evil tricks. It can sneak Trojan software past anti-malware defenses. It can imitate a USB keyboard and execute any commands on the target PC. It can hijack Internet traffic and change DNS settings to redirect a user's outbound traffic to any server it pleases. If planted on a phone or other USB device with an Internet connection, it can eavesdrop on a user's communications.

There is currently no way to ensure that your USB device's firmware is clean of such malware. There are no digitally signed versions of USB firmware that can serve as certified "clean" standards.

The only defense against the USB attack vector is to jealously guard your USB devices. Don't plug them into any port that is not a trusted device, say the experts. But following that protocol will drastically reduce the usefulness and convenience of USB devices.

For example, you can't safely plug your flash drive or phone charging cable into a friend's computer, unless you are 100% certain that person's computer is virus-free. (Plugging into a USB port on a PUBLIC computer has never been safe.) Neither can you trust a flash drive, mouse, keyboard or digital camera that you've borrowed, bought used, or that has been used by someone who is not diligent about security. Presumably, USB devices purchased new will be safe.

What's The Solution?

USB device manufacturers will have to step up and address this problem. One solution is to implement "code signing," an encrypted digital certificate that certifies a firmware package was clean when it left the factory and has not been altered. But first, we'll have to convince OEMs that this is their problem, not just ours. And that solution will only fix the problem for new USB gadgets, not the untold millions already in circulation.

Nohl told Wired magazine that he contacted an unnamed USB drive maker and described his team's findings. The vendor repeatedly denied that it was possible. Wired contacted the USB Implementers Forum, a trade organization that manages the USB standard. Its spokesperson responded with this statement:

"Consumers should always ensure their devices are from a trusted source and that only trusted sources interact with their devices," she wrote. "Consumers safeguard their personal belongings and the same effort should be applied to protect themselves when it comes to technology." In other words, it's your problem and no concern of the people who sold it to you. That will not sit well with consumers.

Let me reiterate... any USB device (flash drive, external hard drive, smartphone, digital camera, mouse, keyboard, etc.) that has been plugged into an untrusted computer should be treated with suspicion -- much like a used hypodermic needle. Further, erasing, formatting, or using anti-virus tools will not remove malicious code from the firmware of USB devices. And there is no known method at this time to scan USB devices to see if they are clean.

So on a practical level, what should you do? I think it's important to recognize that this vulnerability is new, and (as far as we know) it hasn't been exploited yet. So it seems likely to me that we don't have USB gadgets with infected firmware in circulation, for now. My advice is that if you use USB devices, do so with this threat in mind from now on. A 32 GB flash drive sells for about $15. If you have a flash drive that's been connected to unknown or public computers, you might want to discard it.

Read more:
http://askbobrankin.com/alert_serious_security_flaw_in_usb_drives.html#ixzz39kJ2Fsur

---

Now something for the technologically inclined to ponder!

Why we're moving to the IPv6 network protocol

By Doug Spindler

For decades, all devices connected to the Internet have had an IPv4 network address. But the IPv4 address space is nearly tapped out.

To provide more headroom for new devices, we're all transitioning — in fits and starts — to IPv6. Here why that's important.

For most PC users, the transition from IPv4 to IPv6 will happen under the hood. But advanced PC users and administrators will want to understand the ramifications of the IPv6 rollout.

Adoption of the Internet outstrips technology

On Jan. 1, 1983, the ARPANET — which we now call the Internet — began using the now ubiquitous Transmission Control Protocol/Internet Protocol (TCP/IP) Version 4, most often referred to as simply **IPv4.** (IPv1, 2, and 3 were experimental; they were never used on the public Net. IPv4 completely replaced Network Control Protocol.)

Every device connected to the Internet has an IPv4 address, and for the past 31 years, the Net has worked fine with 3,706,452,992 ***public*** IPv4 addresses. (Some IPv4 addresses are reserved and designated as ***private*.**) But the unprecedented growth of the Internet — and all the devices attached to it — has highlighted the limitations of IPv4. It won't be long before all IPv4 addresses are assigned and there'll be no way to attach new devices to the Internet. We'll know that day has arrived only when the last IPv4 address is handed out. But we're getting there much faster than anyone predicted.

About 20 years ago, computer scientists foresaw that we would run out of IP addresses, and they began work on IP Version 6. (IPv5 never made it out of the lab.) Not knowing whether IPv4's end would be sooner or later, Microsoft added IPv6 support to Windows Server 2000 and Windows Professional 2000. That was 14 years ago. (Apple, Linux, and router vendors did the same.)

As computer scientists worked on IPv6, others started looking for ways to conserve and reclaim IPv4 addresses using Network Address Translation (NAT) in routers. But this was only a temporary fix. With IPv6 not yet mainstream, a typical supply/demand marketplace developed for IPv4 address space — and it was a seller's market. Two years ago, Microsoft purchased 666,624 IP addresses for U.S. $7.5 million. As reported in a MaximumPC story, one of the last assets sold by the defunct Borders Group was a block of 65,536 IPv4 addresses. Software vendor Cerner purchased them for $786,432, or $12 per address.

The depletion of IP addresses is a worldwide problem. Who gets blocks of IPv4 addresses is managed by the Internet Assigned Numbers Authority (IANA). On June 10, 2014, IANA announced shortages of IPv4 address space for Latin America and the Caribbean. Of IANA's five regions, three have depleted their IPv4 pools.

Soon after, Microsoft announced that it had run out of U.S. IP addresses for its Azure cloud service. As reported in a PC Pro story, the company had to borrow addresses from its Latin America pool. In a blog post, Microsoft senior program manager Ganesh Srinivasan stated:

> "IPv4 address space has been fully assigned in the United States, meaning there is no additional IPv4 address space available. This requires Microsoft to use the IPv4 address space available to us globally for the addressing of new services. The result is that we will have to use IPv4 address space assigned to a non-US region to address services which may be in a US region. It is not possible to transfer registration because the IP space is allocated to the registration authorities by Internet Assigned Numbers Authority."

(The blog was later updated to say that Microsoft actually did have some IPv4 space in the U.S. but the situation was "dynamic.")

To be clear, this is ***not*** a reason to avoid or abandon Microsoft's Azure service. But it tells us that IPv6 is in our immediate future.

Since 2000, Microsoft has supported IPv6 in all of its operating system. (There's a common misconception that IPv6 can be removed or disabled. It can't be removed, but it can be disabled. Because IPv6 is an integral part of the Windows, Microsoft states that disabling it will probably break some Windows components or cause erratic performance. The company believes there is no valid reason to disable IPv6 and strongly advises against it.)

IPv6's address space provides vast room to grow

We're running out of IPv4 addresses because of IPv4's 32-bit structure. (It's analogous to the RAM and hard-drive limitations of a 32-bit PC.) To say that IPv6's address space is vast is an understatement. Based on a 128-bit format, Version 6 supports 340 ***undecillion*** addresses (2 to the power of 128, or 340,282,366,920,938,000,000,000,000,000,000,000,000).

Users see IPv6 addresses as hexadecimal values (0 through 9 plus A through F) broken into eight blocks, separated by colons. A typical IPv6 address looks like **2002:1a4d:9eda:a95f:2973:0a0b:4791:023d.** (There are rules for shortening addresses that contain blocks of zeroes, so you might see an address formatted as something like 2001::1.)

Just as with IPv4, IPv6 addresses can be assigned (leased) by a DHCP server, manually assigned, or automatically assigned. The Windows IPv6 network properties dialog box (see Figure 1) looks similar to the IPv4's, though you'll notice that the subnet setting looks a bit different. Whereas IPv4's setting is called **Subnet mask,** IPv6's is labeled **Subnet prefix length.** Most of the time you'll use a subnet prefix length of 64.

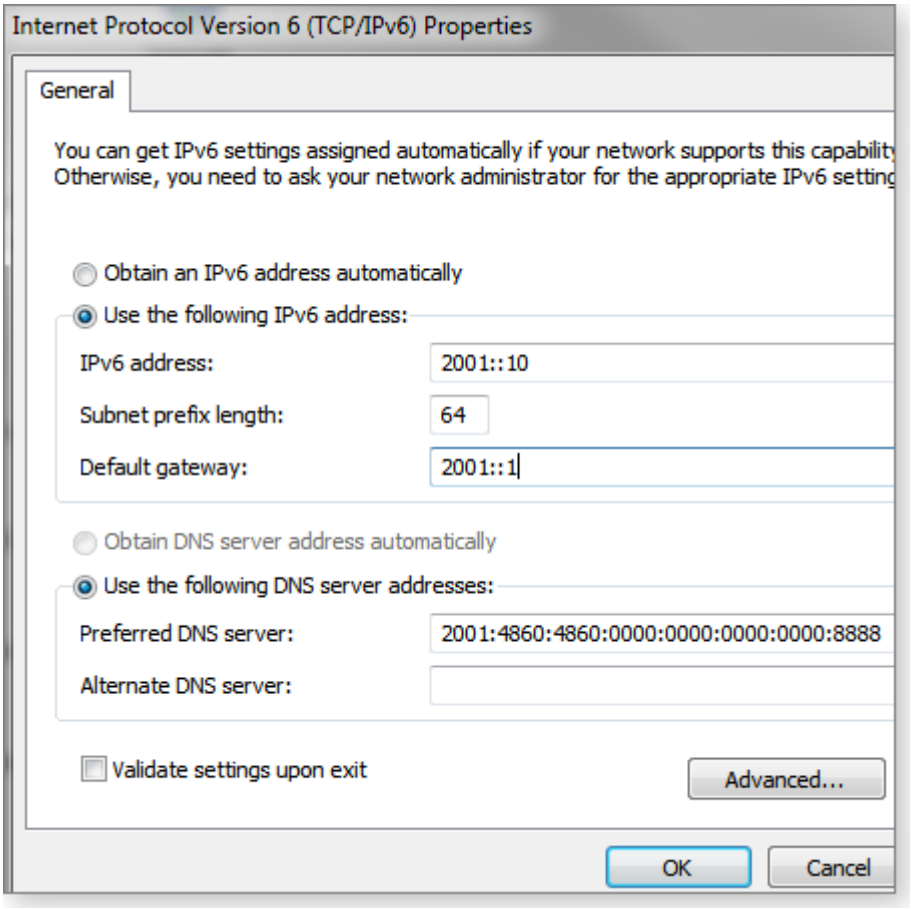


Figure 1. Typical entries for a static IPv6 address

The settings for IPv6 are very similar to those of IPv4, with the exception of the subnet. With both protocols, the subnet setting defines the number of possible addresses that exist only on the local network. A typical IPv4 subnet is 255.255.255.0 — which is a long way of saying that there can be up to 256 separate addresses on the local network. A PC might have a local address of 10.0.0.9 or 10.0.0.255.

The subnet size in IPv6 is shown as a simple integer. For example, Figure 1 shows a subnet length of 64 — the default in Windows 7. With a 64-bit subnet mask, the first 64 bits of the address is for the network and the second 64 bits is for the specific machine (host).

For the IPv6 address shown in Figure 1, I used shorthand notation — **2001::10** — to eliminate the zeroes. The full address is **2001:0000:0000:0000:0000:0000:0010.** If I'm setting up static addresses for other machines on my network, the addresses might be 2001::10, 2001::11, 2001::12, etc. Your home router probably doesn't support IPv6, so you can either leave the address box blank or enter **2001::1** into the box. (An IBM information page has more on IPv4 and IPv6 address formats. A link at the bottom of the page takes you to a quick discussion on subnets. Also, check out the ipv6now.com "IPv6 Prefix Primer" [downloaded PDF].)

To confirm your settings, open up a command window, type "ipconfig" at the prompt, and click Enter. Figure 2 shows my sample settings.

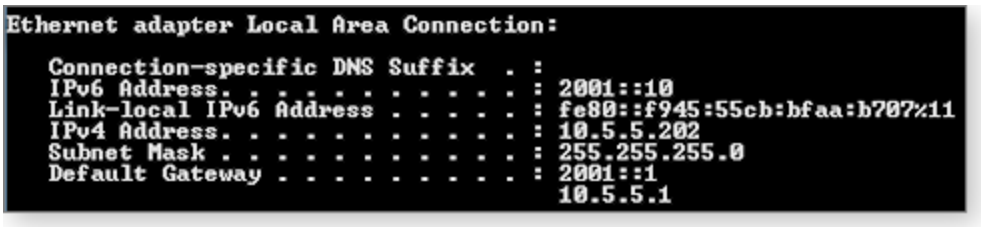


Figure 2. Typing "ipconfig" into a command prompt window confirmed my IPv6 settings.

Figure 2 shows two IPv6 addresses. The first, **IPv6 Address,** is the global address — the one used to communicate with other hosts on the Internet and other network devices. **Link-local IPv6 address** is automatically generated when your computer boots, and it's used for communications on the local network. Link-local traffic passes through local hubs and switches but is not forwarded outside the local-area network by routers. (All Link-local addresses begin with **fe80.**)

You can use the command-prompt command **ping** to test IPv4 and IPv6 connections between different machines on your local network. Get the IPv4 and IPv6 addresses for another local device, and then ping those addresses from a command window. Try with IPv4 and then IPv6. (For a list of ping options, enter **ping /?** at a command prompt, as shown in Figure 3.)

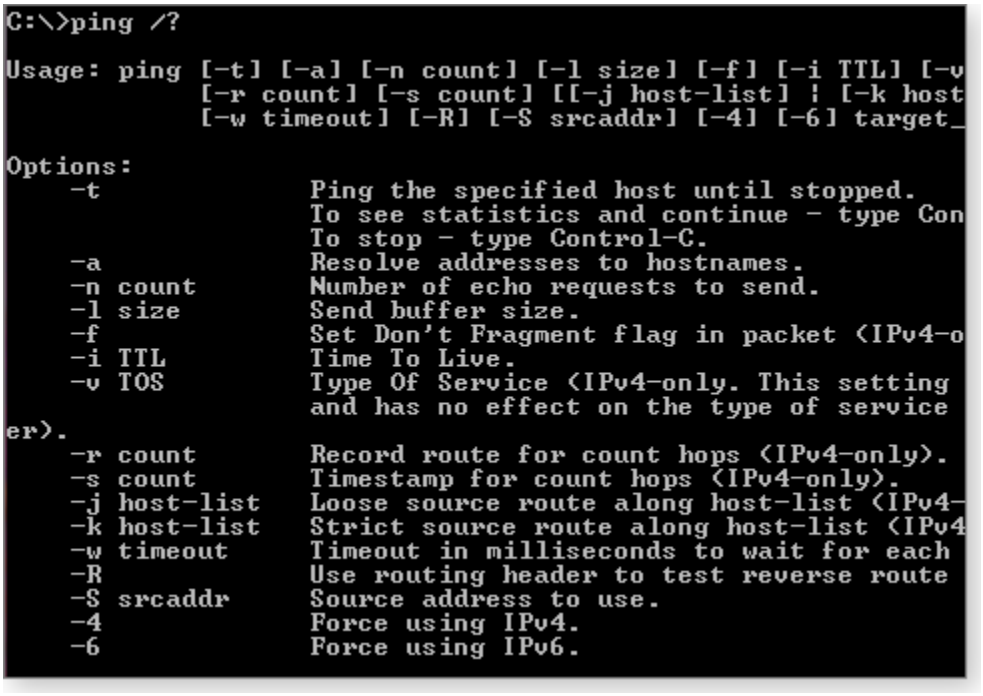


Figure 3. The *ping* command's list of optional switches

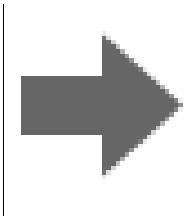
Pinging will show that IPv4 and IPv6 communications are completely separate. For example, try pinging another machine's IPv6 address, using the **-4** option. The ping won't find the other machine. Now ping another machine, using its Link-local address.

Here's another exercise, using the KAME Project: Open a Web browser and connect to kame.net. It's actually two websites, one for IPv4 and the other for IPv6. But they look identical. If you see a swimming turtle, it means you've connected to the IPv6 website; if the turtle isn't swimming, you've connected to the IPv4 site. (Also try it with different browsers.) Here's the point: When connecting to a website, a DNS server will correctly resolve your request, whether the site is based on IPv4 or on IPv4 — no user intervention required. By default, Microsoft will try IPv6 first, and then try one of the transitioning technologies (more on that below) and finally IPv4. (That's something to keep in mind when troubleshooting network communications.)

What if your local machine and a website have only IPv6 addresses and your ISP is only IPv4? Microsoft has three IPv4 and IPv6 transitioning technologies: **6to4, ISATAP** (Intra-Site Automatic Tunnel Addressing Protocol), and **Teredo.** (Home users might be most familiar with Teredo.) All three use different little tricks to insert IPv6 traffic into an IPv4 packets — so they can be sent trough an IPv4 network. Once the packets reach their destination, they're unpacked and read as IPv6 packets. Type **ipconfig** or **ipconfig /all** on Win7 or Win 8 to see one or more of these transitional adapters listed — though they require changing a setting to become active. (Be aware that in rare instances, attackers are using the transitioning adapters as attack vectors.)

For home and small businesses, the Teredo technology should be of interest because many home routers already support it. For more on Teredo, see its Wikipedia page and the MS TechNet overview.

This article should give you a starting point. For more information, I suggest checking out Hurricane Electric (site). The organization has been supporting IPv6 for years, and it offers free online IPv6 training and certification (more info). Hurricane Electric also offers alternative transitioning technology — a tunnel broker (more info) — for home and small-business computers. The tunnel broker is relatively easy to set up and, like Microsoft's products, installs a virtual network adapter to encapsulate IPv6 traffic in IPv4 packets.) Give it a try and see whether you can connect to the swimming turtle at the KAME Project.



Is this stuff for real?

The Virgin Mobile division of Sprint, in its noble effort to provide more affordable ways for the poor to remain poor, will introduce a $12 per month wireless Internet service that lets you connect only to Facebook, Instagram, Pinterest, or Twitter, and you can choose only one. But for only $10/month more you can waste time on all four social media networks.

Two months ago, the U. S. government relaxed restrictions on satellite imaging, permitting high-altitude photos of objects smaller than 50 cm (1.64 feet). Now imaging company DigitalGlobe has launched a satellite camera capable of rendering objects as small as 10 inches. So it's still safe for most men to sunbathe in the nude.

As they say in LOONEY TOONS "That's that's all folks!"
