

[Full-Disclosure] w32.frethem.k@mm and good reading

Source: <http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2002-07/0350.html>

From: Nathan Fain (full-disclosure@lists.netsys.com)

Date: 07/16/02

From: full-disclosure@lists.netsys.com (Nathan Fain)

Date: Tue, 16 Jul 2002 13:28:05 +0300

This method of obfuscation applies If you are only protecting a static website that doesnt deal with any security critical data (credit info, shopping, etc.) and defacement of your site is a primary concern. Script kiddies deface websites (with few exceptions). Script kiddies run OS fingerprint scan's or other scans to find their target. Yes, script kiddies will be quite fooled by this method. Otherwise you are only obfuscating your own perception of security. IP stacks are secure in their own right (i haven't heard of anyone gaining remote access by exploiting the IP stack). And this is about all you change keeping with an older OS itself.

The idea of keeping older versions of services you required (ie Apache) has little application as well. It applies to the concern of stability.

Once **proven** stable (left to your own interpretation) one should switch to the later version. Reason being that at some point the developers will stop looking at code in older version all together. So even if you have the oldest version of apache with the latest patches available for it, you will likely have wide vulnerabilities in functions that are used in the code or underling libraries. And in such a scenario, while you might have stopped script kiddies, you have left the door wide open for anyone determined to get in your system.

The article applies to those whose primary concern is perhaps defacement alone

> *By Robin Miller, NewsForge.com*

>

>> *Posted: 06/06/2002 at 12:10 GMT*

>> *[724.gif] Here's an interesting way to secure an Internet-connected computer against intruders: Make sure the operating system and software it runs are so old that current hacking tools won't work on it. This was suggested by Brian Aker, one of the programmers who works on Linux.com, NewsForge, Slashdot, and other OSDN sites; he runs several servers of his own that host a number of small non-profit sites in the Seattle area. "I have one box still running a version of*

>> *Solaris that's so old none of the script kiddies can figure it out,"*
>> *Brian says. "They tend to focus on the latest and greatest, and don't*
>> *have the slightest idea how to handle my old Sun box."*
>> *Brian points out that some of the most secure Department of Defense*
>> *Web sites -- ones that don't make headlines by getting cracked all the*
>> *time -- run old versions of Mac OS and the venerable WebSTAR server*
>> *suite. "[Mac is] a great operating system for that application," he*
>> *says. "No scripting or remote capability at all, so there's no way for*
>> *them to get in."*
>> *Not only that, the hacker/cracker crowd is fixating, as usual, on the*
>> *latest versions of everything, like Windows 2K/XP, Mac OS X, the most*
>> *recent Linux kernels and BSDs, the newest Solaris, and so on. What fun*
>> *is there in breaking into a system running something so ancient only a*
>> *dad would even consider using it? There's also an obscurity factor to*
>> *consider here, and not the one proprietary software advocates usually*
>> *trot out when discussing security issues.*
>> *True "security through obscurity"*
>> *Most Web site takedowns and system intrusions make use of known*
>> *vulnerabilities in a particular operating system or server software*
>> *package. These vulnerabilities are typically discovered, a little at a*
>> *time, by thousands of bad hackers who poke and prod at systems,*
>> *port-scanning and probing them, sharing the information they gain from*
>> *their (mostly failed) attempts with each other. A million monkeys with*
>> *Internet connections may not reproduce any Shakespeare plays -- they*
>> *need to use old-fashioned typewriters to do that -- but they sure as*
>> *bleep are going to find vulnerabilities in any host they contact*
>> *sooner or later simply by sheer weight of numbers, especially if the*
>> *operating system or software they attack is popular enough that they*
>> *have many instances of it out there to look and poke at. It doesn't*
>> *matter whether the operating system and server software under attack*
>> *is proprietary or Open Source. Sooner or later, with enough monkeys*
>> *scratching at it, every single chink or opening can be discovered and*
>> *exploited.*
>> *Imagine a custom operating system used by only a few servers, running*
>> *server software so oddball that cracking lessons learned on mainstream*
>> *servers don't apply to it at all. Or imagine running a DOS variant or*
>> *an OS like AIX that has never been widely used for Net-attached*
>> *servers but is adequate for handing out simple Web pages and receiving*
>> *responses through online forms and handling email, which are the*
>> *primary tasks performed on most publicly-accessible servers.*
>> *Now imagine your local script kiddie trying to crack a box running an*
>> *operating system and server software he's never seen before, about*
>> *which no information is available in the usual online hacker hangouts.*
>> *Chances are, he's going to move on to an easier target.*
>> *This is security through obscurity at its finest. Even if the custom*
>> *operating system and server software are Open Source, low-level*
>> *attackers aren't going to bother poring over the code thoroughly*
>> *enough to find its vulnerabilities, and those few who have the skill*
>> *level needed almost certainly have better things to do with their time*
>> *-- like work -- and won't bother.*
>> *Really dumb stuff*

>> *Never forget, most intrusions and defacements exploit really stupid administrator or user mistakes, like using "password" as the password for remote access or running all kinds of unnecessary services that create security holes so big a whale could dive through them. These lapses have nothing to do with the operating system or software being used. No operating system or application ever written is immune to user stupidity. Some just take more stupidity to botch than others, you might say. But that's enough about that. Let's go back to talking about old operating systems.*

>> *Age before beauty*

>> *One advantage of mature software is that lots of people have already tried to crack it and lots of patches have been written. A smart sysadmin like Brian, running an ancient version of Solaris, has kept up with security updates over the years and has installed all of them he has found. What some people might sneer at as "obsolete" software, others might call "carefully tested" or "proven." Indeed, Debian Linux users often point to the fact that Debian's stable branch does not include the latest kernel or software as one of its great strengths; Debian lets others explore the latest and greatest -- and fall victim to the latest and greatest exploits -- before all the kinks are worked out to the Debian maintainers' satisfaction.*

>> *Note that an awful lot of servers out there are still running on Red Hat 6.1 or 6.2, not Red Hat 7.x, and that it takes a long time for the latest version of Apache to trickle out into the world full-strength. Because these programs have zero licensing cost attached to updates, why would so many sysadmins keep using old versions when new ones no doubt offer more and slicker features? Obviously, those sysadmins have the same outlook as delivery truck fleet managers who refuse to buy a new model during its first year or two in production. They prefer to wait until all the kinks are worked out and all the defects and maintenance tricks have been discovered and applied by early adopters before jumping from the tried and true into something new.*

>> *This is sane behavior for a conservative business manager whether she is running a fleet of Web servers or a fleet of trucks -- or even a fleet of Web servers for a trucking company. But it may be even more sane to hold on to the same servers and trucks even when others sneer at them as being old, even if new versions are smoother and easier to administer or drive. Quite simply, once you have worked with a piece of software or a truck for a number of years, you know its quirks inside and out. When it acts up in a subtle way someone not used to it might not even notice, long experience with it can point an observant sysadmin or mechanic straight to a problem, thereby saving downtime and repair costs.*

>> *Because "Total Cost of Ownership" is the big management buzz phrase that cuts across all business areas, and anything new requires a learning curve, sometimes it is best to just keep on using the old whatever as long as it does its job reasonably well.*

>> *At some point -- hopefully before Microsoft stops supporting it -- Windows NT may be reasonably secure against most common exploits. If nothing else, by that time there will be hundreds of thousands of sysadmins who have learned how to secure it as hard as possible, even*

>> *if they had to learn some lessons the hard way -- by getting cracked.*
>> *At the same time, the script kiddies and malicious hackers who ran*
>> *roughshod over NT servers when they first appeared have aged. Most of*
>> *them probably have jobs and responsibilities by now, and aren't*
>> *getting their kicks playing in other people's systems but are busily*
>> *securing ones they run themselves.*
>> *The next generation of bad-kid hackers probably won't mess much with*
>> *NT -- or pre-X Mac OS or Linux pre-2.5 kernels or Apache pre-2.x or*
>> *any of the other operating systems and server applications their*
>> *fathers or older siblings ran "back in the day," while those same*
>> *fathers and older siblings will have piled up endless experience*
>> *securing those old, now-obscure programs, making them harder targets*
>> *than the latest stuff.*
>> *You never read about this kind of "security through obscurity," which*
>> *can just as correctly be called "security through obsolescence."*
>> *Despite this lack of publicity, it may be as effective a tactic as any*
>> *other, and it can be implemented without spending a dime.*
>> © Newsforge. All rights reserved
>
>
>

> *Full-Disclosure - We believe in it.*
> *Full-Disclosure@lists.netsys.com*
> *http://lists.netsys.com/mailman/listinfo/full-disclosure*
>