NEOHAPSIS - Peace of Mind Through Integrity and Insight

Messages sorted by: [date] [thread] [subject] [author]

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Moderators -- I know the open/full disclosure debate has been kicked numerous times, but I think this one is worth putting through. It is in response to the following:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/colum
ns/security/noarch.asp

>It's Time to End Information Anarchy

It's time to end insecure coding practices and insecure systems management.

>Code Red. Lion. Sadmind. Ramen. Nimda. In the past year, computer worms
>with these names have attacked computer networks around the world,
>causing billions of dollars of damage. They paralyzed computer
>networks, destroyed data, and in some cases left infected computers
>vulnerable to future attacks. The people who wrote them have been
>rightly condemned as criminals. But they needed help to devastate our
>networks. And we in the security community gave it to
>them.

Worms and virus' have been created long before "security research" was fashionable. Code Red, Nimda and a few of the more recent worms were

made possible not by the research that discovered the vulnerability they exploited but by the lack of awareness and training by system administrators who did not patch their systems.

>It's high time the security community stopped providing blueprints for
>building these weapons. And it's high time computer users insisted
>that the security community live up to its obligation to protect them.
>We can and should discuss security vulnerabilities, but we should be
>smart, prudent, and responsible in the way we do
>it.

Working with vendors to release a patch/fix is the responsible thing to do. That being said, in the past vendors have had to be literally forced to release a patch with the threat of proof of concept code. If a proper security aware culture is promoted within the certification processes and by all vendors, the release of exploit code along with a patch would be trivial as system administrators would patch critical systems. This is of course assuming that the patch is properly tested and actually works. Without the existence of exploit code, how do we ensure that the patches actually work?
Trust our vendor? I don't think so, vendors have proven that they bow to stock price and the so called market pressure and will continue to do this over and above security needs. Multiple vendors, not just Microsoft, have also proved that they will not completely research the issue themselves and release insufficient patches.

>First, let's state the obvious. All of these worms made use of security
>flaws in the systems they attacked, and if there hadn't been security
>vulnerabilities in WindowsR, Linux, and SolarisR, none of them could
>have been written. This is a true statement, but it doesn't bring us
>any closer to a solution. While the industry can and should deliver
>more secure products, it's unrealistic to expect that we will ever
>achieve perfection. All non-trivial software contains bugs, and modern

>software systems are anything but trivial. Indeed, they are among the >most complex things humanity has ever developed. Security >vulnerabilities are here to stay.

Correct, if there were no flaws in the operating systems the worms would not exist. But, on the other hand, if system administrators are properly trained and security aware the worms may have existed but they would have failed. If a security issue is discussed in the public, someone somewhere will be able to extrapolate enough information to generate exploit code. Consulting organizations create exploit code, scary underground hacking groups do it -- so why not as a security researcher do it? Regardless, if a vulnerability is discussed in an open format, the exploit code will be created.

Not discussing vulnerability information is not an option either.
The best option is to have secure software, in absence of secure software, we need proper patch management and proper training. If you look at the MCSE training programs of the past (NT 3.51, early 4.0 days) you can literally count the number of times the work security is mentioned on one hand. To me this is what makes worms like Code Red successful.

>If we can't eliminate all security vulnerabilities, then it becomes all >the more critical that we handle them carefully and responsibly when >they're found. Yet much of the security community handles them in a way

>that fairly guarantees their use, by following a practice that's best >described as information anarchy. This is the practice of deliberately >publishing explicit, step-by-step instructions for exploiting security

>vulnerabilities, without regard for how the information may be used.

I hate to repeat myself, but it is impossible to discuss a vulnerability

without giving enough information that would allow someone else to re-discover the problem and use it. How useful are advisories or vulnerability discussions that say: "There is an issue in Win2K that will allow me to execute commands in system context remotely. We recommend that you disable the spooler service or install this patch" there is no valuable information for someone who wants to understand a problem in these types of discussions. Those who do not wish to understand the problems are, in my opinion naive and should not be giving their vendor complete trust and control over their systems.

Look at the flack Novell recently received over their release of the Padlock patch. They released a patch, with no information just an urgent message to install it at once. Who in their right mind would install this without asking questions?

>The relationship between information anarchy and the recent spate of >worms is undeniable. Every one of these worms exploited >vulnerabilities for which step-by-step exploit instructions had been >widely published. But the evidence is more far conclusive than that. >Not only do the worms exploit the same vulnerabilities, they do so >using the same techniques as were published - in some cases even going

>so far as to use the same file names and identical exploit code. This >is not a coincidence. Clearly, the publication of exploit details >about the vulnerabilities contributed to their >use as weapons.

Yes, but not only was step by step vulnerability information published, but step by step patch information was published to even more sources than the vulnerability information was.

>Providing a recipe for exploiting a vulnerability doesn't aid >administrators in protecting their networks. In the vast majority of

>cases, the only way to protect against a security vulnerability is to
>apply a fix that changes the system behavior and eliminates the
>vulnerability; in other cases, systems can be protected through
>administrative procedures. But regardless of whether the remediation
>takes the form of a patch or a workaround, an administrator doesn't
>need to know how a vulnerability works in order to understand how to
>protect against it, any more than a
>person needs to know how to cause headache in order to take an
>aspirin.

But what about the patches that don't work? Or the ones that cause additional problems? Without the ability to test a system after it has been patched system administrators are defenseless. It isn't Information Anarchy, it is common sense that you test a system to be 110% sure that it is patched. You are right, I don't need to know what causes a headache to take an aspirin, but I do need to know that the aspirin will work.

>Likewise, if information anarchy is intended to spur users into
>defending their systems, the worms themselves conclusively show that
>it fails to do this. Long before the worms were built, vendors had
>delivered security patches that eliminated the vulnerabilities. In
>some cases, the fixes were available in multiple forms - singleton
>patches, cumulative patches, service packs, and so forth - as much as
>a year in advance. Yet when these worms tore through the user
>community, it was clear that few people had applied these fixes.

Who is to blame for this? Patches are not installed because system administrators are not taught the importance of it. Instead they are told that the patch is not regression tested and that they should wait for a proper service pack.

>Finally, information anarchy threatens to undo much of the progress
>made in recent years with regard to encouraging vendors to openly

>address security vulnerabilities. At the end of the day, a vendor's
>paramount responsibility is to its customers, not to a self-described
>security community. If openly addressing vulnerabilities inevitably
>leads to those vulnerabilities being exploited, vendors will have no
>choice but to find other ways to
>protect their customers.

It is "Information Anarchy" as you put it that has forced vendors to begin addressing security issues. I have said many times that Microsoft as an organization has done a lot to address security issues compared to that past, but there is still a way to go and Microsoft is not the only vendor out there generating insecure software. There are still multiple vendors in this day and age that would rather ignore security issues than spend the money to fix them.

>This is not a call to stop discussing vulnerabilities. Instead, it is a
>call for security professionals to draw a line beyond which we
>recognize that we are simply putting other people at risk. By analogy,
>this isn't a call for people for give up freedom of speech; only that
>they stop yelling "fire" in a crowded movie house.
>

Repeating myself once again, there is no way that a vulnerability can be discussed properly without letting enough information out that would allow someone else to discover the issue. Merely saying that there is an issue with a specific service will cause multiple people and groups to begin looking at that service to find the issue.
"Fire" needs to be yelled when there is a fire.

>Some security professionals go the extra mile and develop tools that
>assist users in diagnosing their systems and determining whether they
>are affected by a particular vulnerability. This too can be done
>responsibly. In many cases, it's possible to build a tool that
>performs non-destructive testing and can only be used by a legitimate

>system administrator. In other cases, the specifics of the
>vulnerability make it impossible to limit how the tool could be used -
>but in cases like these, a decent regard for the well-being of the
>user community suggests that it would better to not build the tool
>than to release it and see it misused.

A tool that "non-destructively" tests for a vulnerability can be easily re-engineered to exploit the vulnerability. Not building the tools means that everyone should trust that the patch does what it is supposed to do. Lets look at the extreme of this, a malicious user manages to find his way into a patch repository of a vendor. This user replaces working patches with ones that do nothing. The system administrator does his duty and installs the patches. This admin has no way of knowing that the patch actually does nothing or worse -- backdoors his system further. You cannot tell me that there have been no incidents were a software vendor has been compromised and you cannot guarantee that there never will be again.

>Ending information anarchy will not end the threat of worms. Ethics and

>intelligence aren't a package deal, and some of the malicious people
>who write worms are quite smart. Even in the best of conditions, it
>will still be possible to write worms. But the state of affairs today
>allows even relative novices to build highly destructive malware. It's
>simply indefensible for the security community to continue arming
>cybercriminals. We can at least raise
>the bar.

I agree with this comment. Right now we have nothing but a bunch of low-skilled script kiddies using tools that are pre-made for them. Hell, some of them even use the commercial vulnerability scanners as their tool.

But, not releasing complete vulnerability information will not stop the

more skilled people. It is unfortunate, but there is a large group of very skilled people who would love to do nothing more than code their own exploits and release them to the lower skilled population. There is nothing we can do to stop this, so why not try and capitalize off of it by learning from the code and even creating our own that we can easily footprint. By forcing this information into closed room discussions, we are blinding security managers who, because of the open discussion and tool repositories, know exactly what footprints the various exploits leave and know exactly what to tune their log watching or IDS' for.

>This issue is larger than just the security community. All computer
>users have a stake in this issue, and all of us can help ensure that
>vulnerabilities are handled responsibly. Companies can adopt corporate
>policies regarding how their IT departments will handle any security
>vulnerabilities they find. Customers who are considering hiring
>security consultants can ask them what their policies are regarding
>information anarchy, and make an informed buying decision based on the

>answer. And security professionals
>only need to exercise some self-restraint.

I agree with working with the vendor. There only valid reason for releasing proof of concept code is a vendor that does not cooperate. But, without the threat of this, many vendors can simply ignore important issues.

>For its part, Microsoft will be working with other industry leaders
>over the course of the coming months, to build an industry-wide
>consensus on this issue. We'll provide additional information as this
>effort moves forward, and will ask for our customers' support in
>encouraging its adoption. It's time for the security community to get
>on the right side of this issue.

I look forward to this process and hope that I am involved in some way.

Regards;

Steve Manzuik
Moderator - VulnWatch
www.vulnwatch.org

-----BEGIN PGP SIGNATURE-----
Version: PGPfreeware 7.0.3 for non-commercial use
<http://www.pgp.com>

iQA/AwUBO84YWQQ+sDtVIYbcEQIGewCgkzPhyaxDWZTahPnMNFQbUi
WX52EAoI5l
xXinuEj2ZWr96BIRuBieObWk
=9AxH
-----END PGP SIGNATURE-----

- **Messages sorted by:** [date] [thread] [subject] [author]