# ;login:

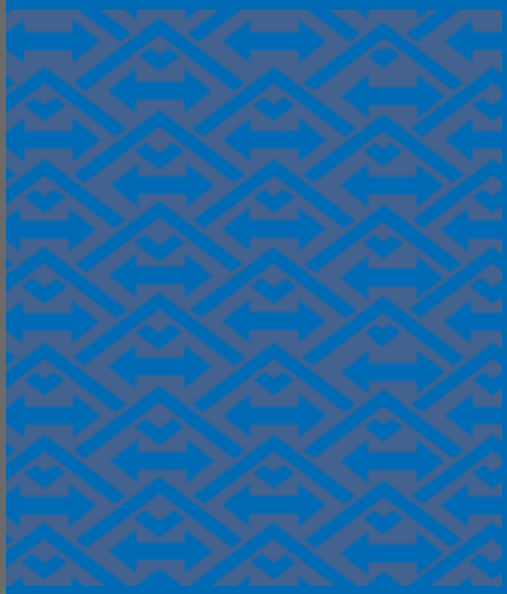
THE MAGAZINE OF USENIX & SAGE

December 2000 • volume 25 • number 8


inside:

SECURITY:

THE NETWORK POLICE BLOTTER




USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

# the network police blotter

**by Marcus J. Ranum**

Marcus J. Ranum is CTO of Network Flight Recorder, Inc. He's the author of several security products and of a book on computer security (with Dan Geer and Avi Rubin) and is a part-time sysadmin.

*<mjr@nfr.net>*

The first year of the twenty-first century is ending. I guess that's not a particularly big deal unless you're an Arthur Clarke fan. Did you notice that in *2001*, the HAL 9000 computer suffered a security incident, when it was apparently subverted by an alien trojan-horse program? I guess firewalls don't work in the sci-fi future. Then again, I'm not convinced anymore that they work now. My impression of computer security in sci-fi is that it's either ignored as if it's a solved problem or it's a plot device in which the aliens'/bad guys'/good guys' systems are penetrated using an unsophisticated trick like guessing a password. With a bit of luck, computer security will somehow improve without requiring a complete parallel upgrade of Homo Sapiens V1.0.

I've long maintained *Ranum's Law,* "You can't solve social problems with software" but never projected into the future far enough. The implication is that computer-security problems will remain the same as they are today, no matter what we do – because people, not computers, cause problems. Have you ever seen a computer hack itself? It takes a human to do something that stupid.

In order to make the next great leap in computer security, we need to change our attitudes and address the problem socially, not technologically. I've been really happy to see the beginnings of a sea change in opinion about hacking – recently, a number of companies have been jumping loudly on the "we don't hire hackers" bandwagon. Next, I'm guessing we'll see wider recognition of the fact that a lot of the "grey hat" hacking going on consists either of ego-driven attempts to count coup on unpopular vendors or stealthy attempts at marketing security services. Getting a clear picture of things is important to making progress – especially with social problems.

Even the best technology is not going to help unless it's wisely and correctly used. As I was pondering computer security in sci-fi, I had an amusing mental picture of Capt. James T. Kirk telling the ship's computer to do something and having the computer reply, "Password." We've known passwords were an obsolete technology for a very long time now, but we're still using them – so why will the future be any different? It'll only be different if we make it so.

Are there any good things on the horizon in the short term? Well, I'm seeing one trend that I like, and a recent item supporting it from none other than Microsoft. The current trend of security is "penetrate and patch" – find a bug, fix a bug, find a bug, fix a bug. This is also known as "kludging your way into heaven," and it would work except that heaven doesn't approve of kludgy code. A few years ago I was drinking too much beer in some hotel bar at some conference, and predicted that eventually we'd see software that would update itself automatically: instead of reading BugTraq and seeing that (oh, joy!) you have to rush and upgrade your Web server because 10,000 script kiddies were just given a tool that breaks it, you wake up in the morning with a nice email from your Web server informing you that last night it upgraded itself and it hopes you're enjoying your coffee. The antivirus software vendors figured this out a while ago – having your defense system update itself automatically is a good thing. Why can't my operating system do that also? And my Web server? And my firewall? In fact, that's what this new thing from Microsoft does: it checks to make sure your copy of your Web-server software is up-to-date with security patches. It's a start, anyhow. It's called HFCheck. If it works, I bet you we'll see more tools like it.

One of the biggest problems with patches is getting the user to actually install them. I'm hoping that eventually they won't have to. I'd like to be able to install a piece of software and, when I install it, tell it, "You're mission-critical. Notify me immediately if you need to be updated but don't do anything until I tell you to." Or, "You're security-critical. If you need to be updated, just do it automatically or shut yourself down and call for help." There are downsides to this concept, of course. One of them is that it further legitimizes the "penetrate and patch" model – it just speeds up our ability to patch. The other is that it may further reduce the role of software testing. If a vendor can just throw patches out one after another, what's the point in getting a release "right" anymore? I suspect software testing is already a casualty of "Internet time," though. With the full-disclosure crowd insisting on patches with a subweek turnaround, it's no wonder that everyone is being forced to run beta-test software. Perhaps our future piece of software won't simply be told, "upgrade yourself if you need to." We might tell it, "upgrade yourself only to improve reliability; don't upgrade yourself to add features without asking me first; upgrade or halt to fix security flaws." This would move us to a model of software development in which everyone is running the "code du jour." Is this a fair price to pay? I think we're already paying the price but aren't yet reaping the benefits.

One of the biggest problems with patches is getting the user to actually install them.

Incidentally, I believe the big breakthrough in self-upgrading software will come when some business visionary points out that the self-upgrade cycle represents a perfect opportunity for a vendor to favorably "touch" its customers at a regular interval. If the whole Application Service Provider model works out, we may even see self-upgrading software as an eventual implementation of "rental" software – you get to run the latest and greatest version of whatever it is as long as you're paying a subscription fee monthly or annually. Microsoft has already pioneered this approach with its developer tools, which presently self-upgrade in the form of a steady stream of CD-ROMs sent in the mail. Indeed, the operating systems of the future might offer load-on-demand application services. Imagine for a second if your operating system came on a 1.4M floppy disk and installed itself in 20 seconds. Then, the first time you try to send email it presents you with a list of mail systems, their subscription prices, and so forth, and lets you pick and go. Software might look like signing up for satellite TV: you'd buy a number of "points" with your platform package that could be "spent" to subscribe to a browser application, a word processor, a mailer, a security audit service, and perhaps an archival backup service. This implies that system administration will need to be a "solved" problem within the next ten years (no more Windows installation questions, please) – is anyone out there listening?

## Some Feedback

My recent articles about hacking and full disclosure have certainly touched a few nerves. I asked for feedback/rebuttals and have gotten more than I can print or even paraphrase here. About 70% of the messages I got were actually supportive; it's good to know I'm not completely in la-la land. So, some feedback.

Jeff Root writes:

> The short version: what you describe is a fantasy version of the future. It will never come to pass.
>
> The longer version: what you propose (i.e.: launching lawyers against full-disclosure sites) is emotionally satisfying, but is analogous to ridding the kitchen of cockroach-

> es by turning on the lights. It appears to work; there are never any visible critters, but in fact their actual number does not decrease.

Analogies are always dangerous, since they can be extended beyond their intended bounds, often with ridiculous results. What I propose is actually more analogous to putting all the food away in the kitchen when it's not in use, and spraying a little insecticide in all the obvious roach-breeding spots. It appears to work, there are never any visible critters except for a few dead ones, and their actual number does decrease. Sometimes it decreases enough that the roach population becomes manageable.

A. Nonymous writes:

> We must know our enemy. This is a fact of life everytime good guys try to combat bad guys/things: in war, medicine, biology, earthquake . . . I learned much more on BugTraq than in all vendor bulletins combined. We (white hats) must have access to as much information as possible in order to better prepare our defenses. This implies free access to technical descriptions of the holes, probably also exploits to test our defenses, and freedom to reverse-engineer the programs that we use so that we can protect them before the vendors do . . .

"Know your enemy" is an ancient maxim of warfare – I'm sure Sun Tzu wasn't the first person to say it. But, in wartime, there are stiff penalties for "aiding and abetting" the enemy – which, in my opinion, is what a lot of "grey hats" are doing when they release tools to script kiddies. We "white hats" must have access to as much information as we need to strengthen our systems – as far as I am concerned it is enough that a vendor tells me to "upgrade XYZ pronto!" I revere curiosity as the root of all human learning, but sometimes it's best left unsatisfied. We white hats already know about buffer overruns and how to avoid writing them into our code – it's not intellectually stimulating to be drowned in a sea of reports of buffer overruns in versions of software you don't run.

A. Nonymous continues:

> In crimes (and cybercrimes are primarily crimes) the intention is more important than the objects used. Some people carry guns but don't kill; some commit murders with ropes or knives but we don't ban these objects. Releasing an exploit program is like selling a knife: it's morally OK and the real difference is in the mind of the end user.

"Toolz don't hack Web sites, script kiddies do!" – perhaps the NRA will adopt the slogan one day, but I doubt it. There are two important issues at play here:

- what is legal
- what is moral.

Lawyers exist to address (some might say "muddy") the first of those. The second is one that individuals must resolve in their own hearts with whatever guidance they deem appropriate. I believe that there are some serious ethical lapses in the security field today – I don't believe that what's going on is necessarily illegal. Yet.

In the parts of the world that have not fallen into anarchy, the ownership and distribution of guns are regulated. You're right that ropes aren't. They aren't for two very good reasons:

- It is much easier to kill someone with a gun than a rope or a knife.
- Ropes and knives have a wider variety of benign uses than guns.

For the record, I am a gun owner and enjoy shooting paper targets. I'm not happy about having to register my guns; unfortunately, a lot of bad apples have already spoiled that barrel for the rest of us. I'm *very* unhappy that irresponsible individuals have made suspects out of those of us who do act responsibly. Part of my sense of responsibility includes recognizing that if some kid got his hands on one of my guns and did something stupid with it, I'd feel terrible (and probably spend a lot of money on legal bills). Consequently I'm very careful to keep them unloaded and locked in a very secure storage facility. See any parallels here?

Societies choose to regulate such things to a greater or lesser degree based on the trade-off of utility to lethality in the hands of an unpremeditating or inexperienced killer. I think that society may make a similar determination with hacking tools. Releasing an exploit program is not like selling a knife; it's like handing out loaded submachine guns and claiming that the end result is not your responsibility. I don't buy that, and neither would most juries.

Releasing an exploit program is not like selling a knife; it's like handing out loaded submachine guns and claiming that the end result is not your responsibility.

By the way, in many parts of the US you must be over 18 to buy certain configurations of knives. The seller can get in a lot of trouble for selling a double-edged combat boot knife to a minor. You can also get in a lot of trouble for carrying one on your person, whether you intend to use it or not; be especially careful about airport metal detectors.

With power comes responsibility. Many of the nifty-keen Internet technologies we build are powerful tools that will help shape the future of global information. Those of us who build those tools – or build tools that might be harmful if abused – must accept responsibility for the consequences of our actions.

Harald Nordgard-Hansen writes:

> I often find one thing that really annoys me. That is all the vendors that tend to brush away or downplay the severity of security holes with a statement along the lines of "no exploits exist for this hole" or "has not been observed in the wild." There seems to be an implicit assumption that until someone writes an exploit and distributes it, the hole is not really serious and there is no reason to allocate resources in the organization to fix it.

Downplaying the severity of a hole is a double-edged sword, indeed. On one hand, if vendor make the hole sound minor, nobody may install the patch. On the other hand, if they make the hole sound major, they may trigger a panic. Remember back in the days when something like the Michaelangelo virus was a big deal? I suspect that people are becoming so jaded by the relentless flood of bugs that it'll be hard to raise their blood pressure anymore. I wonder sometimes if the vendors are afraid that someday they may be held liable (like, say, for example, tire manufacturers?) for failures in their products that result in end-user damage. In fact, I'm *sure* they're afraid of the idea, which is one reason they're trying to head the threat off with UCITA.

This is one of the big changes that has to happen. Vendors have to own responsibility for taking security bugs seriously and fixing them in an appropriate and safe manner. Some of them have gotten away with dodging that responsibility for a long time. I think that era has come to an end, though.

The issue of making vendors take a bug seriously until there's an exploit in the wild is a serious one. By now it should be obvious that once a bug is found, someone's going to do a press release about it, so it's just a matter of time until there's an exploit. However, I don't think responsible security practice is to *force* their hand unduly by threatening them with an exploit, the way many "grey hats" do. Let's suppose someone discovered

that a certain brand of tires failed catastrophically if they were driven over marshmallows. Responsible practice would not be to scatter marshmallows all over the streets "to get people to take the problem seriously."

Adam Shostack writes:

> my system got hacked
> lamerz get slammed in juvenile court
> like fish in the sea

Last time I checked, CNN said that "mafiaboy" (allegedly the "brains" behind the denial-of-service attacks against amazon.com and CNN) had 64 new charges being levelled at him by prosecutors. Think of it as evolution in action.

## Up Next

Next column I promise I'll try to lighten up a little bit. For the next contest, I'll accept nominations for "practicality prize" winners. These are to be real-life, incredibly great things that are overlooked and perhaps underappreciated. Let's give them their day of glory! Best suggestions will get a nifty windbreaker. Some sample favorite practical things I love: those nitrogen doodads in the cans of Guinness – what a brilliant idea. And those Velcro cable ties. And . . . Email to <*mjr@nfr.net*>; Subject: "practicality prize."