Have a Cocktail: Computer Security Today

From: “Marcus J. Ranum” mjr@nfr.net

An Experiment in Comparison

Let’s play pretend!

Imagine for a minute that there’s a city that has severe problems with urban vandalism: 15% of the residences and buildings in that city are covered with spray-painted graffiti, and 6% of them have been fire-bombed, broken into, or otherwise damaged. Criminals in the city are fearless; some of them stand on the street corner giving out cans of spray-paint, sledgehammers, and unlighted molotov cocktails to anyone who wants to join in the fun. The police force in our mythical city is hopelessly under-staffed, and is so overwhelmed by the level of vandalism that they can typically only get involved in cases in which a major business is being damaged. Owners of residences that are attacked are written off and (albeit regretfully) told they have to take care of themselves. Into this city come security companies, offering their services to businesses and homeowners alike, saying they will help prevent vandalism against their customers for a fee. Business for the security companies is so good they are struggling to hire enough people. The situation is desperate: only the wealthy can afford security or can get the attention of the police, and the number of criminals is growing every day.

That’s an ugly scenario, in our imaginary little town, and I’m sure most of us wouldn’t want to live or start a business there. But it gets worse! Suddenly, some of the security companies set up booths on street corners, where they give out free cans of spray paint and molotov cocktails just like the criminals. In fact, their molotov cocktails are professionally designed, and sport a snappy corporate logo – they work better than the homemade ones some of the criminals used. Whenever one of these security companies decides to give away a new lot of weapons, they advertise it on the local TV and radio stations. Some sponsor parties at their booths complete with DJs and bikini-clad girls, and explain to press that they are doing it:
“*To educate the community about how poor a job the police force is doing, and to show them why they have to work **harder** to protect themselves against criminal activity*.”

Lastly, because of their manpower shortage, the security companies start aggressively recruiting criminals, graffiti artists, and fire-bombers. They explain to their customers that they hire ex-criminals because the ex-criminals understand how graffiti artists and vandals work and are as well (if not better) trained than the local police in dealing with the problem. They explain that an “ethical fire-bomber” can come look at your business, tell you where *he’d* throw a firebomb, so you can be prepared, and that you’ll be safer as a result. Some of the security companies hold classes in how to throw firebombs, so people can learn how to think like fire-bombers themselves.

Back to Reality

Obviously, this is a ridiculous and contrived story! Or is it? In fact, it represents *exactly* what is going on today in Internet computer security. Over the last three years, the trend has been toward what many call “full disclosure” in security problems, and a lot of security companies have jumped on that bandwagon with a vengeance. The doctrine of full disclosure argues that the vendors of security-critical software would not be sufficiently responsive in fixing bugs if they were not *forced* to fix their bugs by having

them publicly exposed. Further, proponents of full disclosure argue that the bad guys may already know of the security bug, so keeping it secret and trying to manage it quietly may be playing into their hands by keeping critical information from the good guys. I believe that a big piece of the problem is that the computer security community has intellectually distanced itself from its *real* constituents: the users of the Internet. Arguing about the fine points of who discovers what, first, ignores the fact that the average Internet user doesn't want to have to worry about security. It's not interesting to them. They don't particularly care about security, and have better things to do their time than to constantly patch their system against security bugs. It's largely academic to them, until someone breaks into their system and ruins their whole day.

Let's look at a recent case that attracted a lot of attention: Brown Orifice. A security researcher discovered a vulnerability in Netscape's Java implementation, which would allow a hacker to install a mini web-server within a user's browser. Once this mini web-server was operating, the hacker could view and delete files within the user's system, invisibly over the network. The person discovering the bug put up a web site, with example code for how the bug operates, and posted the information to security mailing lists. Somewhere in this process, Netscape was notified of the bug, but, obviously, it was too late for them to effectively react to protect or notify their users. Within days of the bug's being revealed, thousands of users' systems were compromised. Who stood to benefit from this situation? Obviously, the way the bug was disclosed didn't help Netscape. Obviously, the way the bug was disclosed didn't help the thousands of innocent users whose systems were compromised. That really leaves only two classes of people who stood to gain: the bad guys who now had a useful new weapon, and the security researcher who revealed the bug. Last time I checked his web site, it said Netscape is giving him $1,000 for his assistance and that his site has had over 1,000,000 hits "thanks for your support." He's gone from being *a nobody* to being *a somebody* for a short while. I wonder if that's a big consolation to the thousands of people who have been negatively affected by his discovery. Why didn't he just report the bug to Netscape and wait for them to fix it like a responsible security practitioner would?

One of the first denial of service tools[1] was written by a fellow who, at that time, worked for a security products company that offered intrusion detection systems to monitor attackers activities. It's an unintentional irony that a few weeks after the tool, SYN flooding, was published in the hacker journal "Phrack" his employer released a version of their intrusion detection system which would alert victims to the attack. This kind of activity raises more questions than it answers. Never mind the basic ethics of the situation, companies that employ such people are running a substantial risk of being caught looking like the proverbial dentist that has a candy jar in his waiting room. Nor are these isolated instances! A very popular penetration and hacking tool, Back Orifice, is primarily authored by an employee of a well-known security consultancy. An engineer who used to work for a company that builds operating system integrity checkers left his job there and now assists in developing hacking tools that evade operating system integrity checkers. The list goes on, and if anyone was able to document the depth and breadth of the problem, it would take your breath away.

---

[1] Denial of service attacks are attacks intended to render a system useless without necessarily breaking into it. In other words, they don't benefit the attacker, they simply hurt the victim. Denial of service attacks earlier this year were responsible for interruption of service attacks at Ebay, Amazon.com, and CNN.com. Internet businesses have lost hundreds of millions of dollars in lost revenues due to denial of service attacks in the last 4 years.

Those are some obvious cases of unethical behavior. What about the less obvious cases? What about the web sites that disingenuously claim to be helping promote security awareness, which also have downloadable archives of hacking, attack and penetration tools? There are literally dozens of such sites, ranging from venture-funded security portals down to privately maintained personal sites. Many of them warn users that the tools are only there for educational purposes and it's the responsibility of the person downloading them to use them responsibly. This is not unlike our fictional friends who pass out their free spray paint and molotov cocktails while asking the vandals to please exercise responsibility in their use.

Many conferences today offer "hacking" classes to their attendees. People who attend these classes are given a hands on introduction to the tools and techniques of hackers, along with explanation of how to defend against those same attacks. The sponsors of such classes proffer the excuse that "the bad guys already know all this stuff" and "most of these holes are all fixed, already" but conveniently ignore the question of what kind of message this sends to the community. Their message is "hacking is cool, hacking is fun, hacking is OK." When you're living in a town that's constantly being vandalized by spray paint, you're not going to be impressed if someone starts holding "how-to-graffiti art" classes on your street corner. The "hacking" classes are extremely popular, and the conferences and instructors are making a great deal of money teaching these dubious skills.

Set a Thief to Catch a Thief

Perhaps the most pernicious piece of nonsense being foisted on users by some security practitioners is the notion that "it takes a hacker to build good security." I'm not sure who started it, but my guess is it was probably someone who had a vested interest in seeing hackers employed as security experts! The argument goes something like this:

- Hackers are able to find flaws in many systems
- Clearly they are smart
- Clearly they have highly developed analytical capabilities
- That intelligence and analytical ability can be put to good use making sure systems are secure

I'm biassed, clearly, since I've been spending 13 years building computer security products without the "benefit" of a hacking background, but I believe the "ethical hacker" argument has some fundamental flaws. The first, and most critical flaw is simply that:

The Huns knew how to sack Rome; they didn't know how to build a Rome.

Skills at destroying something do not necessarily imply the skills necessary to build something. Indeed, the skills needed to destroy something are a subset of the skills necessary to build it. As with real insects, there are a huge number of individual computer bugs – but these bugs can be categorized into phyla just like real insects. Moths and Butterflies are both lepidoptera, and share many features in common; someone can understand a great deal about moths and that knowledge will generalize pretty effectively to knowledge about butterflies (but not ants). One popular form of attack that is being exploited on the Internet today is the *buffer overrun*, a class of software mistake that is often a cause of vulnerability. For someone who is building a security critical networked application, knowledge of how to prevent buffer overruns is necessary expertise. However, it isn't useful to know about the thousands of individual buffer overruns in other applications – because presumably someone who is building applications is *avoiding* making those mistakes. The hacker's skills lie in searching for

those mistakes, but the security-competent application designer's skills lie in avoiding those mistakes in the first place! Of course, a security-competent application designer might still make the mistake, and a hacker will find it – but so could another security-competent application designer. Really, this argument builds a case not for having hackers launch "penetration tests" against applications, but rather for having security-competent application designers check out eachother's work in a collegial manner. In most development shops, this is known as a "code review" – one programmer does a sanity check on another programmer's workmanship. It's a basic part of software engineering. The fact that many organizations omit code review and quality assurance doesn't mean we need more hackers, it means we need more code review and quality assurance! The current Internet software development environment is to write code fast and toss "beta test" versions to live users. Not surprisingly, many of these "beta test" applications are rife with security holes. To return to the early analogy, the hacker is like an insect collector, who knows how to find and categorize insects, while the security-competent application designer not only knows how to find and categorize insects, but how they work, in depth.

"Ex-"hackers are being widely used as analysts in penetration tests against existing networks. Usually, what they do is search for a catalog of well-known flaws, and identify them if they are present. Indeed, there are automated tools (which cost a lot less than hiring a hacker) that do the same thing; they simply test against an exhaustive list of known holes and total them up. It's a lot like having an exterminator come and check for termites in the foundations of your house: they'll look in the usual places for the usual signs. When they find a flaw, it's easy to fix it. But is that the way to build a secure system in the first place? It turns out that the way to build a secure system is *to design it from the beginning to be secure*. This requires a lot of fundamental knowledge about security, system design, policies, procedures, and analyzing requirements. I don't know about you, but I'd rather have someone taking a top-down view of building my critical systems, rather than a "hunt and peck" strategy focused on finding individual flaws.

My experience as a security products builder for 13 years is that there are loads of smart hackers out there, but they're not the people who are building worthwhile security systems. Why? Because their knowledge of security is perforce detail-oriented and scattershot; they have encyclopedic knowledge of flaws but lack the fundamental understanding of security necessary to build complete solutions. Your typical skilled hacker would make a pretty good security analyst if he spent a couple of years studying the fundamentals. The Huns were smart, too, but they didn't have civil engineering skills, or the logistical skills necessary to construct large cities. That's one reason that the remaining Hunnish artifacts in archaeology are relatively small and barbarous when compared with the structural integrity and beauty of the Roman structures that have lasted to this day.

What* is *a responsible security practitioner?

A recently-announced Neilsen survey indicates that there are 136 *million* Internet users in the US today, out of a worldwide Internet population of 300 million users in 20 countries. A significant percentage of those people, at any time, are at risk from computer security-related attacks. The problem is only going to get more important as the population of users continues to increase.

Therefore, I believe there is only one practical definition of responsibility for security practitioners and companies:

The responsible security practitioner will always work to protect the immediate interests of end user security.

That means that companies purporting to be improving security will not be releasing attack tools, inventing new attack techniques, and training people in how to hack. Nor will they be hiring hackers, or tolerating employees who are playing both sides of the fence. Let's face it, one of the main reasons that the "gray hat" hackers[2] release their exploits is to gain attention and ego gratification. Many gray hats work for security companies, indeed, many security companies primary marketing vehicle is the activities of their gray hat hackers:
Responsible security practitioners promote themselves by their positive contributions, not the amount of damage they could do or have done.

What Can We Do?

The first thing we can do to improve the situation is to stop tolerating it. This process has already begun. An increasing number of computer security firms have sworn off hiring hackers. My company, Network Flight Recorder, remains one of the small handful of firms that has a policy of never hiring even "ex"-hackers; it's grounds for immediate termination if an employee is found to have ever been involved in hacking. Many customers are now asking consultants and consulting firms to not send "ex"-hackers to their sites; I've even heard of customers requesting background checks on consultants.

The most important thing you can do to secure your network and improve the state of security is to support pro-active security practices. Unfortunately, many people don't take security seriously until it is too late; get staff – especially developers and network managers – involved in the security process as quickly as possible. Make sure staff are adequately trained in security design. There are a number of excellent professional organizations that promote not only good security but good computing/administration practices and research. The USENIX association has done a great deal to improve the level of security and systems knowledge in the community, along with its sister organization SAGE (The System Administrator's Guild) which focusses on the craft of system and network management.

In the next 5 years, the Internet will touch virtually all aspects of our lives, whether we want it to, or not. Recent news (as of this writing) a major wire service's web site was hacked, and they had to contact 10,000 customers advising them to cancel their credit cards because their account information had been stolen. Whether you want it to be or not, Internet security is a part of your life. Ask yourself what kind of behavior you expect from the security products companies that service you, and if you see a kid reaching for a can of spray paint, encourage him to find an appropriate canvas.

About the author:

Marcus J. Ranum is Chief Technology Officer of Network Flight Recorder, Inc. Involved with computer security for over 13 years, he is widely recognized as a security visionary,

---

[2] "White Hat" is a term used to describe computer security professionals that do not and have never dabbled in cybercrime and hacking. "Black Hat" refers to active hackers. "Gray Hat" is a term often used to describe hackers who want to have their cake and eat it to: they incite and encourage cybercrime but claim their hands are clean because they, themselves, never actually do anything wrong.

consultant, instructor and architect of practical security systems. He’s the author of several security products, a book on computer security (with Dan Geer and Avi Rubin), and a part-time system administrator. He is a frequent speaker at conferences, and spends a lot of his time in airports and hotels.