;login:

THE USENIX MAGAZINE

December 2003 • volume 28 • number 6

Focus Issue: Security

Guest Editor: Rik Farrow

inside:

USENIX

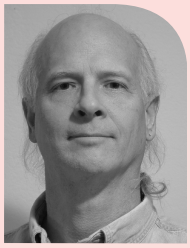
The Advanced Computing Systems Association

in this issue

by Rik Farrow

Rik Farrow provides UNIX and Internet security consulting and training. He is the author of *UNIX System Security* and *System Administrator's Guide to System V*.

rik@spirit.com



Welcome to the sixth security special edition of *;login:*. Since 1998, I have invited security researchers, lawyers, and other professionals to write articles for these editions. I have also received proposals each year from those who had simply heard about the security edition and wanted to contribute to it. If you want to submit a proposal for the next security issue, May 2004 would not be too early.

I am especially excited about the articles submitted this year. We start off with perhaps the last security paper from Tom Perrine, who worked at San Diego Supercomputer Center until his move to Sony. Tom and Devin Kowatch used the abundant disk and computing capabilities at SDSC to show why crypt-based passwords should never be used anymore. Even without clusters of computers, the time required to crack a single crypt password is too short to believe.

Chris Wysopal writes about why security researchers can find exploitable bugs that software vendors miss. For example, when Microsoft announced the first patch for the RPC vulnerability (MS03-026), security researchers immediately found several other problems, resulting in a second patch from Microsoft.

Jason Damron has researched techniques for application fingerprinting. In his article, Jason shows how various versions of Apache httpd servers can be distinguished just through their responses (ignoring the "Server:" line, if present).

Ed Balas, of Indiana University and the Honeynet Project, explains how the Generation II honeynets can capture encrypted data – unencrypted. His clever ploy uses a kernel module roughly based on the Adore rootkit.

Markus Jakobsson and Filippo Menczer report on a new technique for using email forms for denial of service. Their article includes appropriate remedies that need to be applied to any site offering email forms.

Mudge digs into the realm of tunneling, using protocols in ways they had never been intended for. In particular, Mudge explains how to detect tunneling, focusing on HTTP.

Abe Singer, also of SDSC, writes about how SDSC has survived for many years, very successfully, without firewalls. Abe points out the most important areas for anyone who wants to improve security at a site, especially for sites that demand a large degree of openness.

Renaud Deraison and Ron Gula write about using Nessus. Rather than just offering a rehash of online documents, Renaud and Ron provide hints for making your use of Nessus more efficient, as well as customizing the .nessusrc file.

Dario Forte explains how he and an international team of law enforcement personnel tracked down a slick and aggressive group of computer criminals. According to Dario, this group used custom rootkits to control critical systems, but it was a more common tool that helped to bring the organization down.

Ryan Russell writes about patching Windows systems. Ryan explains why this is one of the most critical security issues that exists today, and why he considers agent technology the right way to go.

Finally, Erin Kenneally has the entire legal component of this issue to herself. Erin has been researching the problems surrounding the use of computer logs as evidence. She includes examples of the use of logs and compares these uses with the current US legal

standards for the acceptance of evidence. Erin has also been working with others to create the "secure audit log" model, designed to follow the standards set by the rest of Western law for evidence admissibility.

As always, summaries from the annual Security Symposium are included. As I spent most of the symposium in the basement, running the Ask the Experts track, I personally was very glad to be able to read the summaries, and I want to thank all of the summarizers (again).

I believe that we all enjoyed a wonderful period, roughly coinciding with the growth in popularity of the Internet and going up to the dot-com crash that ended an era of "irrational exuberance," to quote an overquoted Federal Reserve Board chairman. During the Internet runup, technology specialists, including programmers and network and system administrators, were highly valued as the producers of new wealth. Once the bubble popped, the belief in technology as a source of wealth largely vanished as well. Many organizations decided that they no longer needed the services of highly paid specialists. After all, the systems had been installed, and the traffic was flowing over the network, wasn't it?

But we are still in the early stages of computer science and security. Most of our applications have terrible designs, our protocols are flawed, and security is an afterthought. Running networks and systems is still an art, not yet a science. And security? I often believe that we would do better by starting over, with simpler operating systems, cleaner and better documented interfaces, and network protocols based on the lessons we have learned. But how likely is this?

CyberInsecurity

On Thursday, September 25, Dan Geer was fired from his job as CTO of @Stake. Geer is the primary author of the paper "CyberInsecurity: The Cost of Monopoly" (*http://www.ccianet.org/papers/cyberinsecurity.pdf*), in which he is critical of Microsoft, one of @Stake's largest clients. In this paper, co-authored by six other well-known names in security, Geer discusses some issues that are already well known, such as the danger of having a single operating system, a monoculture, that provides fertile ground for worms and viruses. Geer also points out that the "edges" of the Internet are where the fastest growth occurs, with the total computing power of the Internet doubling every 10 months. And the users of these edge systems are not computer security experts. They are home users, and small business owners, who are buying new computers with a Microsoft operating system pre-installed.

Geer also cites the complexity of Microsoft's operating systems. According to Microsoft's own figures, the NT code base has grown at 35% a year, and Internet Explorer at 220% a year. Software experts often describe complexity as proportional to the square of code size. Thus, NT grows by 80% in complexity each year, while IE grows an astounding 380%. Based on the growth in complexity, Microsoft products will have between 15 and 35 times the number of flaws as other operating systems now, with increased disparity in the future.

Microsoft products will have between 15 and 35 times the number of flaws as other operating systems

Geer goes on to write than not all of this complexity is accidental. While Microsoft has begun to employ best programming practices on its internal code, the interfaces provided to application writers are undocumented. Geer uses the proprietary Exchange interface, designed to work with Outlook, as an example.

Microsoft programmers must provide clear modularization and code interfaces on internal code, but the external interfaces are designed to be complex, so that they can-

Signing does not make code secure, but only serves to associate the code to the signer of that code.

not easily be duplicated. The goal of this design is to be anti-competitive. The side effect of this practice means the code that faces the network and all local programs is unnecessarily complex. And that complexity means that there will be more bugs, including many exploitable holes.

Geer et al. do not mention what I consider to be an equally important issue. Back in 1998, when USENIX and Microsoft were co-sponsoring Windows NT conferences in Seattle, Microsoft speakers talked at length about the goal of supporting mobile code. COM was being phased out in favor of DCOM, the Distributed Common Object Model, and would form the basis for future Microsoft operating systems. In this model, code is sent to clients, or servers (SOAP), seamlessly merges with running software, and runs with the privileges of this software. To make this code secure, it would be signed using Authenticode.

Signing does not make code secure, but only serves to associate the code with the signer of that code. In 2002, an excellent example of this appeared when the Microsoft MSN Chat control, an ActiveX object, was discovered to have a buffer overflow vulnerability (*http://www.cert.org/advisories/CA-2002-13.html*). Because the object had been signed by Microsoft, it was trusted by the operating system. And because of the design of Windows, a system without an MSN Chat object could be tricked into downloading the vulnerable object through the use of email containing HTML.

Most Windows viruses spread via email, although other mechanisms, such as Web pages and file shares, are also used. Email- and Web-spread viruses rely on the features of IE to interpret and execute code, making the work of virus writers much easier. Users of IE can deactivate ActiveScripting and prevent the spread of viruses that do not require human interaction, such as SoBig. But doing so also disables all plug-ins. That's right, Flash, sound, the Adobe Acrobat reader, all get disabled as well, to make IE secure (*http://www.spirit.com/Network/ net0502.html* and *http://www.kb.cert.org/vuls/id/25249#solution*).

Ethics

Geer was fired from his position at @Stake the day his paper was published. I found myself defending Geer, whom I know through USENIX, as some people considered what he had done to have been unethical. That is, as an employee of a company which had Microsoft as a frequent client (*http://www.atstake.com/research/reports/eval_ms_ibm/objectives.html*), Geer should not have done anything that would harm the company by which he was employed. He would also have been under NDA (Non-Disclosure Agreements) that might have prevented him from publishing anything harmful about an @Stake client.

I believe that the historical standards for complying with unethical orders come from the Nuremberg trials. During these trials, people who had committed human experimentation and genocide argued that they did not make these decisions themselves, but were acting under orders. The Nuremberg trials produced the standard that a soldier or actor of the state is compelled to disobey unethical orders, even if this will be considered treason.

At this point, you may be wondering what the Holocaust has to do with Microsoft or with Geer's ethics. How can I compare something that involved the torture and death of millions to a computer monopoly, or to the well-being of @Stake and its investors? While certainly not anywhere near the same scale of offense, what Geer did involved a similar ethical decision, one that may impact Geer's ability to get work as an executive

in the computer industry. And will the Microsoft monopoly cause millions of deaths? Hardly. But no deaths?

For the sake of discussion, let's consider a specialist in computer security who signs an NDA as part of the process of becoming employed. Note that consultants also sign NDAs before working with a company. Let's present our security person, whom I will call Barnaby, with three different ethical conundrums.

In the first, Barnaby has gone to work for a modest sized, but thriving, software business. As soon as Barnaby gets settled, he discovers that part of his job will be to protect the company's intellectual property. The problem is that this intellectual property was acquired when an executive left another company and brought along the code base from his old company. Should Barnaby comply with the NDA?

Let's raise the stakes. After Barnaby leaves his old employer in disgust, he goes to work for a better-paying job at a pharmaceutical company. Barnaby plows into his work, and it is only after working for months that he discovers the company's big secret. The flagship drug, used for weight loss, has been found to be lethal and is killing, on average, 18 people a year. Should Barnaby comply with his NDA? Keep in mind that disclosing this information would have dire effects on his company's bottom line and on the stock prices so dear to those with options.

In the final example, Barnaby has become an executive in a company that provides code and product reviews for software developers. His company has in the past provided glowing white papers for a software company that has a dominant industry position. This software has become an important element in the monitoring of medical equipment used in ICUs, in nuclear power plants, in the stock market, traffic control, and in many other areas where mistakes can have serious, even deadly, consequences. And Barnaby has discovered that the code in question is nowhere near secure or robust enough to be used in many of the applications where it has been installed. Just the widespread use of this software poses a threat to national security.

Should Barnaby give up his career, possibly violating his NDAs, because he strongly feels that supporting his company's position is not only unethical but may result in death, and certainly will result in considerable economic losses?

Dan Geer is not Barnaby. But in case you believe that this is wholly fiction, remember that the great August 2003 blackout in the US involved a First Energy control facility where the operators lost their consoles during the critical hour when the cascading shutdown occurred – starting on their portion of the grid. Note that this was during the spread of MSBlaster. Also recall that operators of a First Energy nuclear power plant, fortunately shut down for maintenance, lost their consoles when the Windows systems they were using to run X Window servers crashed during the Slammer worm attack. Note that firewalls should have prevented either of these attacks from getting into the networks – but in each case these worms penetrated even supposedly isolated networks.

I find myself standing and applauding Dan Geer for his courage, integrity, and strength. Geer et al. do suggest ways in which Microsoft could modify its strategies to increase the security of all networks. Microsoft would have to give up its monopoly position to do so, and that would be a bad thing for Microsoft, its employees, and its stockholders. But if the alternative is more huge economic hits with each new worm (an estimated $30 billion in August 2003 alone), and potentially the deaths of many (how many people might die during a blackout?), then I think the course that must be taken is crystal clear.

;login:

EDITORIAL STAFF

EDITOR:
Rob Kolstad *kolstad@usenix.org*

CONTRIBUTING EDITOR:
Tina Darmohray *tmd@usenix.org*

MANAGING EDITOR:
Alain Hénon *ah@usenix.org*

COPY EDITOR:
Steve Gilmartin

PROOFREADER:
jel *jel@usenix.org*

TYPESETTER:
Festina Lente

MEMBERSHIP, PUBLICATIONS, AND CONFERENCES

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
Phone: 510 528 8649
FAX: 510 548 5738
Email: *office@usenix.org*
login@usenix.org
conference@usenix.org
WWW: *http://www.usenix.org*