Schneier on Security

August 15, 1999

by Bruce Schneier
Founder and CTO
Counterpane Internet Security, Inc.
schneier@schneier.com
http://www.counterpane.com

A free monthly newsletter providing summaries, analyses, insights, and commentaries on cryptography and computer security.

Copyright (c) 1999 by Bruce Schneier

CRYPTO-GRAM now has over 20,000 subscribers!

In this issue:

Back Orifice 2000

Back Orifice is a free remote administration tool for Microsoft Windows. It's also one of the coolest hacking tools ever developed. Originally

released last July, Back Orifice 2000 (BO2K) is the current release of the software. It works on Windows 95, Windows 98, and Windows NT. It is much better written than the original Back Orifice. And it's free, and open source.

There are two parts: a client and a server. The server is installed on the target machine. The client, residing on another machine anywhere on the Internet, can now take control of the server.

This is actually a legitimate requirement. Perfectly respectable programs, like pcAnywhere or Microsoft's own Systems Management Server (SMS), do the same thing. They allow a network administrator to remotely troubleshoot a computer. They allow a remote tech support person to diagnose problems. They are mandatory in many corporate computing environments.

Remote administration tools also have a dark side. If the server is installed on a computer without the knowledge or consent of its owner, the client can effectively "own" the victim's PC.

Back Orifice's difference is primarily marketing spin. Since it is not distributed by a respectable company, it cannot be trusted. Since it was written by hackers, it is evil. Since its malicious uses are talked about more, its benevolent uses are ignored. That's wrong; pcAnywhere is just as much an evil hacking tool as Back Orifice.

Well, not exactly. Back Orifice was designed by a bunch of hackers with fun in mind. Not only can the client perform normal administration functions on the server's computer — upload and download files, delete files, run programs, change configurations, take control of the keyboard and mouse, see whatever is on the server's screen — but it can also do more subversive things: reboot the computer, display arbitrary dialog boxes, turn the microphone or camera on and off, capture keystrokes (and passwords). And there is an extensible plug-in language for others to

write modules. (I'm waiting for someone to write a module that automatically sniffs for, and records, PGP private keys.)

Back Orifice is also designed to hide itself from the server's owner. Unless the server's owner is knowledgeable (and suspicious), he will never know that Back Orifice is running on his computer. (Other remote administration tools, even SMS, also have stealth modes; Back Orifice is just better at it.) Anti-virus software has been updated to detect default Back Orifice configurations, but that will only solve most of the problem. Because Back Orifice is configurable, because it can be downloaded in source form and then recompiled to look different...I doubt that all variants will ever be discovered.

Okay, so who's to blame here? The Cult of the Dead Cow wrote and released Back Orifice. Surely the world is not a safer place because, as CDC's Sir Dystic put it: "every 14-year-old who wants to be a hacker will try it." BO2K's slogan is "show some control," and many will take that imperative seriously. Back Orifice will be used by lots of unethical people to do all sorts of unethical things. And that's not good.

On the other hand, Back Orifice can't do anything until the server portion is installed on some victim's computer. This means that the victim has to commit a security faux pas before anything else can happen. Not that this is very hard: lots of people network their computers to the Internet without adequate protection. An attacker can even ask the victim to install Back Orifice (social engineering might help); the Worm.ExploreZip worm of this spring did exactly that. Still, if the victim is sufficiently vigilant, he can never be attacked by Back Orifice.

But what about Microsoft's computing environment? One of the reasons Back Orifice is so nasty is that Microsoft doesn't design its operating systems to be secure. It never has. Any program that runs in Microsoft Windows 95 and 98 can do anything. In Unix, an attacker would first have to get root privileges. Not in Windows. There's no such thing as limited

privileges, or administrator privileges, or root privileges. Microsoft assumes that anyone who can run a program can reformat the hard drive. This might have made some sense in the age of isolated desktop computers; after all, if you could run a program, you were standing in front of the machine. But on the Internet, this is absurd.

Windows NT was designed as a secure operating system, more or less. There are provisions to make Windows NT a very secure operating system, such as privilege levels in separate user accounts, file permissions, and kernel object access control lists. However, the configuration that makes Windows NT secure is very very far and distant from the default installed configuration. Microsoft admits this. You have to make 300+ security checks and modifications to Windows NT to make it secure in its default configuration. And on top of this, Microsoft assumes that most users have Administrator access to their desktop machines anyway. They only really worry about network security, not host-end security, which is where they are seriously vulnerable to attacks like Back Orifice 2000. Windows NT could be secure, but Microsoft refuses to ship the OS in that condition (presumably they worry that their spiffy animated fading menu bars may be overlooked).

Malicious remote administration tools are a major security risk. What Back Orifice has done is made mainstream computer users aware of the danger. Maybe the world would have been safer had they not demonstrated the danger so graphically, but I am not sure. There are certainly other similar tools in the hacker world — one, called BackDoor-G, has recently been discovered — some developed with much more sinister purposes in mind. And Microsoft only responds to security threats if they are demonstrated. Explain the threat in an academic paper and Microsoft denies it; release a hacking tool like Back Orifice, and suddenly they take the vulnerability seriously.

Back Orifice Home Page:

http://www.bo2k.com/

Commentary:
http://www.zdnet.com/zdnn/stories/news/... http://www.infoworld.com/cgi-bin/displayArchive.pl?/...

Microsoft's Systems Management Server:
http://www.microsoft.com/smsmgmt/techdetails/remote.asp
http://www.cultdeadcow.com/news/pr19990719.html

BackDoor-G:
http://www.zdnet.com/zdnn/stories/news/...

Counterpane — Featured Research

"Notes on the Design and Analysis of the Yarrow Cryptographic Pseudorandom Number Generator"

J. Kelsey, B. Schneier, and N. Ferguson, Sixth Annual Workshop on Selected Areas in Cryptography, Springer Verlag, August 1999, to appear.

We describe the design of Yarrow, a family of cryptographic pseudo-random number generators (PRNG). We describe the concept of a PRNG as a separate cryptographic primitive, and the design principles used to develop Yarrow. We then discuss the ways that PRNGs can fail in practice, which motivates our discussion of the components of Yarrow and how they make Yarrow secure. Next, we define a specific instance of a PRNG in the Yarrow family that makes use of available technology today.

http://www.schneier.com/paper-yarrow.html

News

Major irony alert: President Clinton signs a bill into law using PGP.
http://www.wired.com/news/politics/0,1283,20775,00.html

A new U.K. bill on e-commerce has the nasty provision that police will be able to demand access to encryption keys if they suspect criminal use of the Internet. Those who refuse get a two-year prison sentence.
http://www.wired.com/news/politics/0,1283,20937,00.html
http://techweb.com/news/story/TWB19990726S0010
Text of the bill: http://www.dti.gov.uk/cii/elec/ecbill.html
Foundation for Internet Policy Research commentary on the bill:
http://www.fipr.org/ecommpr.html

The first three chapters of Alan Turing's treatise on the Enigma, retyped from the only known paper copy, are available at:
http://home.cern.ch/~frode/crypto/Turing/index.html

The L0pht has released an anti-sniffer tool. It detects sniffers on networks. Unfortunately, at least one sniffer-detection-resistant sniffer has been released. And the race continues....
http://www.wired.com/news/technology/... L0pht: http://www.l0pht.com/

The Information Society, an academic journal, published a special issue on anonymity and the Internet: vol. 15, no. 2. Actually, there are interesting articles in most of the back issues.
http://www.slis.indiana.edu/TIS/tables_of_contents/...

The Encrypting File System (EFS) built into Microsoft Windows 2000 has been broken.
http://www.ntsecurity.net/forums/2cents/news.asp?... Microsoft claims that it has not, that the attack is predicated on the user doing something wrong: leaving the EFS recovery key on the machine.
http://www.microsoft.com/security/bulletins/... The author's reply:
http://www.ntsecurity.net/forums/2cents/... I reserve judgment, not having studied EFS, the attack, or Microsoft's response.

In late May, Janet Reno wrote to German Federal Secretary of Justice Herta Daubler-Gmelin, asking her to control the distribution of encryption software over the Internet.
http://www.heise.de/tp/deutsch/inhalt/te/5117/2.html

There's another version of Melissa floating around. This one uses the ".all" extensions in Microsoft Outlook to crash systems. Clever idea, actually.
http://www.computerworld.com/home/print.nsf/all/...

This rather impressive espionage device is being sold as a home consumer item:
http://www.x10.com/home/offer.cgi?!ZDX30,../...

There has been considerable hoo-hah over a U.S. government plan to monitor private networks for intrusion, and invade a lot of privacy in the process. (This will all be at the consent of the various companies, so warrants are not required.) It's called Fidnet, for Federal Intrusion Detection Network.
http://www12.nytimes.com/library/tech/99/07/biztech/...
http://www.zdnet.com/zdnn/stories/news/...
http://www.sjmercury.com/svtech/news/indepth/docs/...
http://techweb.com/wire/story/TWB19990729S0013
http://www.fcw.com/pubs/fcw/1999/0726/...
http://www.infoworld.com/cgi-bin/displayStory.pl?... EPIC's "Critical Infrastructure Protection and the Endangerment of Civil Liberties" http://www.epic.org/security/infowar/epic-cip.html Copy of the White House plan, and commentary:
http://www.cdt.org/security/fidnet/

The House Appropriations Committee has approved a $36 billion budget for the departments of Justice, Commerce and State, but included language specifically barring any spending on FIDNET.
http://www.techweb.com/wire/story/reuters/...

And the U.S. government backpedals.
http://www.fcw.com/pubs/fcw/1999/0802/...

AOL has been hit by an ingenious social engineering attack. This hoax message, masquerading as a hoax warning, fools users into giving up account and credit card information.
http://www.zdnet.com/zdnn/stories/news/...

The FBI is preventing CMI Communications, a Canadian company, from offering satellite phone service in the U.S. because the FBI can't eavesdrop on the calls.
http://www.nationalpost.com/financialpost.asp?...

California adopted a new digital signature law, allowing brokerages to use signed e-mail for contracts.
http://www.computerworld.com/home/news.nsf/all/...

The case against Kevin Mitnick has finally been dropped.
http://www.msnbc.com/news/178825.asp

Congressman Porter Goss (R-Fla) wants to offer a tax break to companies that develop encryption products that enable key recovery or other methods of giving the government access to the encryption keys.
http://www.wired.com/news/politics/0,1283,21014,00.html

A new Excel vulnerability allows a malicious spreadsheet to execute arbitrary code without the user's permission.
http://www.securityportal.com/list-archive/bugtraq/...
http://www.zdnet.com/zdnn/stories/news/...
http://officeupdate.microsoft.com/Articles/mdac_typ.htm

The Ontario Information and Privacy Commissioner has published a pamphlet that recommends that anyone using e-mail learn to understand and use encryption.

http://www.ipc.on.ca/Web_site.ups/MATTERS/SUM_PAP/...

And one last Microsoft item. To help salvage their reputation, Microsoft put a server running a beta of Windows 2000 outside its firewall and dared hackers to break in. The problem was that the server couldn't stay up long enough for anyone to even try.
http://www.zdnet.com/zdnn/stories/news/...
http://www.windows2000test.com/ [dead link as of 2000-02-18]

Counterpane Systems News

Counterpane Systems has changed its name to Counterpane Internet Security, Inc. We have received venture-capital funding from Accel Partners and Bessemer Ventures, and are in the process of creating a series of service offerings in the managed security area. Anyone interested in working for Counterpane in the Bay Area should contact me immediately. Watch this space for more details. This is going to be the coolest security company you've ever seen.

PasswordSafe wins PC Magazine editors choice award:
http://www.zdnet.com/pcmag/stories/reviews/...

Bruce Schneier profiled on guru.com:
http://www.guru.com/channel/tech/portrait/P66.jhtml

Microsoft PPTP's vulnerability discussed:
http://www.zdnet.com/sr/stories/news/...

Bruce Schneier will be speaking at the Scandinavian Network Expo, in the evening on 14 September and then on 15 September
http://www.networkstelecom.com/index_eng.html [link dead as of 2000-04-28; try http://www.networkstelecom.com/engelsk/engelsk.htm]
http://www.firedoor.se/bruce/bruce.var

NIST AES News

AES is the Advanced Encryption Standard, the encryption algorithm that will eventually replace DES. In 1997, the U.S. government (NIST, actually), solicited candidate algorithms for this standard. By June 1998 (the submission deadline), NIST received fifteen submissions. NIST asked for comments on these algorithms, with the intention of pruning the list to five finalists. NIST held an AES conference in Rome in April (this was the second AES conference, the first was the previous August in California), the comment deadline was in June, and last Monday NIST announced the finalists.

They are:

Mars, submitted by a large team at IBM.
RC6, from RSA Data Security (including Ron Rivest) Rijndael, from a team of excellent Belgian cryptographers
Serpent, by three very respected cryptographers, Ross Anderson, Eli Biham, and Lars Knudsen
Twofish, from Counterpane Systems, including myself

NIST didn't just announce the five finalists. They published a 52-page report explaining their rationale — why they chose the algorithms they did and why they did not chose the algorithms they didn't — and it is worth reading to peek at their decision process. It's at http://csrc.nist.gov/encryption/aes/round2/...

The next step is to choose among the finalists. NIST is again soliciting comments on the algorithms, and there will be a third AES Candidate Conference in New York in April 2000, held in conjunction with the 7th Fast Software Encryption workshop. Comments are due by 15 May 2000, and then NIST will propose a standard. The AES will then go through the formal government approvals process and become a Federal Information Processing Standard (FIPS), and presumably will become the standard

encryption algorithm for all sorts of international applications. Expect all this to happen by the summer of 2001; the government moves slowly.

Cryptographers are busily analyzing the submissions for security. It's tempting to think of the process as a big demolition derby: everyone submits their algorithms and then attacks all the others...the last one standing wins. Really, it won't be like that.

At the end of the analysis period, I don't expect serious weaknesses to be found in any of the finalists. The winner will be chosen based on other factors: performance, flexibility, suitability.

This means that we need your input into this process. I know you're not cryptographers, and you won't be able to comment on the mathematics of the various submissions. But you can comment on your encryption requirements, and whether the algorithms will suit your needs.

AES will have to work in a variety of current and future applications, doing all sorts of different encryption tasks: 32-bit microprocessors, 64-bit microprocessors, small 8-bit smart cards, DSPs, FPGAs, custom ASICs, and everything else we can't even imagine yet.

Choosing a single algorithm for all these applications is not easy, but that's what we have to do. It might make more sense to have a family of algorithms, each tuned to a particular application, but there will be only one AES. And when AES becomes a standard, customers will want their encryption products to be "buzzword compliant." They'll demand it in hardware, in desktop computer software, on smart cards, in electronic-commerce terminals, and other places we never thought it would be used. Anything we pick for AES has to work in all those applications.

So how do you comment? NIST is accepting formal comments either on paper or by email. See http://www.nist.gov/aes for instructions. Be sure to identify who you represent and what cryptography interests you have.

Remember, AES is going to be your cryptography standard for the 21st century. We need your help.

NIST Round 2 page:
http://csrc.nist.gov/encryption/aes/round2/round2.htm

FSE 2000:
http://www.counterpane.com/fse.html

Performance comparison of AES candidates:
http://www.schneier.com/paper-aes-performance.html

A version of this essay appears at:
http://www.zdnet.com/zdtv/cybercrime/features/story/...

The Doghouse: HPUX and the UNIX Crypt Algorithm

Here is a comparison of the Solaris and HPUX man pages for the UNIX "crypt" encryption function. Same algorithm, different interpretations, different conclusion.

According to the Solaris 2.6 Crypt man page, "crypt implements a one-rotor machine designed along the lines of the German Enigma, but with a 256-element rotor. Methods of attack on such machines are widely known, thus crypt provides minimal security."

According to the HPUX10.20 man page, "crypt implements a one-rotor machine designed along the lines of the German Enigma, but with a 256-element rotor. Methods of attack on such machines are known, but not widely; moreover the amount of work required is likely to be large."

Reading the HPUX man page, you get the impression that crypt offers adequate protection for your files. It is a sad statement when cryptographic algorithms that are broken as homework for cryptography

students are put forward as a means to protect data by a mainstream OS vendor.

Web-Based Encrypted E-Mail

The idea is enticing. Just as you can log onto Hotmail with your browser to send and receive e-mail, there are Web sites you can log on to to send and receive encrypted e-mail. HushMail, ZipLip, YNN-mail, ZixMail. No software to download and install...it just works.

But how well?

HushMail <http://www.hushmail.com> is basically a PGP or S/MIME-like e-mail application that uses Java (although oddly enough, HushMail is not compatible with either). The sender logs onto the HushMail Web site, and encrypts messages using a Java applet that is automatically downloaded onto his machine. Both the sender and receiver need to have HushMail accounts for this to work. Accounts can be anonymous.
The algorithms are 1024-bit ElGamal for key exchange and signatures, and Blowfish for bulk encryption. But everyone's private key is stored on the HushMail server, protected in a passphrase. This means that one weak link is likely to be the passphrase; it's the only protection you have against someone who has legal or illegal access to the HushMail server. (The current beta — August 99 — doesn't let you change your passphrase, although they promise the feature in the future.)

Another weak link is the Java applet. When you download it, you have no idea if it is the correct applet. Yes, the source code is public, but that doesn't help when you are at a public Internet terminal trying to encrypt or decrypt private e-mail. A Trojaned Java applet can do all sorts of damage, and there is no way to know. Sure, you use an SSL connection between your computer and the HushMail server, but if you don't actually check the details of the received certificate, you have no idea who you are

connected to. HushMail is considering writing something to verify the applet automatically, but then how do you trust the verifier?

This is actually a major problem. The applet can be signed, but who signed it? Even if you check the certificate, the typical browser permits a dozen different PKI roots by default, and any one of them can issue a forged certificate. This means you have to trust them all. And you have to trust that a Trojan didn't drop a phony certificate into your browser. Note that a downloaded verifier can never solve this problem; it just turns the "how do I trust the applet" question into "how do I trust the verifier."

And a third possible weakness is the location of the HushMail servers. Although the company is based in Anguilla, the servers are located in Canada. Presumably Canada is more susceptible to legal attacks. And remember that the security depends on the physical protection of the HushMail server.

All in all, though, HushMail seems like a reasonable implementation of the idea. The company seems clued; they have a reasonably informative Web site, and respond promptly to security questions.

ZipLip <https://www.ziplip.com/zlplus/home.jsp> is different. Both parties do not need an account to communicate. The sender logs onto the ZipLip Web site and, using SSL, sends a message to someone else. ZipLip then sends the recipient a message telling him that your message is waiting. The recipient then logs onto ZipLip to receive the message. Encryption, outside the two SSL connections, is completely optional.

ZipLip won't identify the encryption algorithm used, which is enough to discount them without further analysis. But they do something even stupider; they allow the sender to create an encryption key and then give the recipient a "hint" so that he can guess it. ZipLip's own Web site suggests: "The name of the project we're working on," or "The restaurant where we had dinner last night." Maybe there are 100,000 restaurants, so

that's a 17-bit key.

The threats here are serious. Both the sender and receiver need to verify their SSL connections, otherwise there is no security. The ZipLip server is a major attack target, both because many messages will not be encrypted, and because those that are will have keys weakened by the requirement that both parties remember them.

On the plus side, ZipLip claims a policy of deleting all mail 24 hours after delivery, which provides a level of lawyer-proofing that HushMail does not have...if they implement it properly.

YNN-mail <http://www.ynnmail.com> is barely worth this paragraph. They encrypt stored messages with a 40-bit key, and don't use SSL when you sign up and send them a long-term password. Snake-oil if I've ever seen it.

And I just heard of another, ZixMail <http://www.zixmail.com/> [link moved to http://www.zixit.com/]. I didn't have time to examine it in depth, but the FAQ — look at their wishy-washy comments on encryption — makes it sound like real snake oil, too.

Web-based encrypted e-mail is less secure than PGP-encrypted e-mail (or S/MIME e-mail) for a few reasons. One, the constant interaction between the communicants and the server leaves more opportunity for man-in-the-middle attacks, Trojan horses, etc. Two, SSL-based authentication is more vulnerable to spoofing, since almost no one ever bothers to check the details of received certificates and there is no revocation mechanism in place. And three, there are some very attractive attack targets: servers with large collections of secret e-mail and potential decryption keys. Certainly Web-based encrypted e-mail is better than unencrypted e-mail, but I'd stick with PGP or S/MIME if possible.

This essay was written with input from Fred Wamsley.

A version of this essay appears at:
http://www.zdnet.com/zdnn/stories/comment/...

> **From: "Couvares, Peter F." <peter.couvares@tdstelecom.com>**
> **Subject: Crypto-Hacking**

> For all it's worth, it looks like you were beaten to the punch — I can find at least four prior uses of "crypto-hacking" or "cryptohacking".

> Google turned up the following, among others:

> http://cc2.gamestats.com/wwwboard/messages/894.html [dead link as of 2000-02-18]
> http://www.hotwired.com/talk/club/special/...

> All of them seem to use it to mean hacking a system that employs cryptography rather than hacking cryptography itself, however — your definition is a more useful contribution to the vocabulary.

> **From: John Savard**
> **Subject: Cluelessness Alert. I'm not so sure.**

> I certainly do agree that the military can safely allow public information to be stored on Web sites on commercial hosts. However, I have noted that a lot of military sites are actually on U.S. Government-owned machines in the .mil domain.

> And it is difficult, particularly using common commercially-available operating systems and Internet hosting software, to maintain the kind of impregnable security needed for any system that also contains sensitive information.

> There are ways of making an Internet server essentially immune to most kinds of hacking. Macintosh servers, not having a CLI, appear to

> be quite secure. But there are other techniques, most of which require custom software and even custom hardware.

> For example, to take an idea from the telephone company, how about a computer with two CPUs. CPU number 1 is connected to the hard drive containing the software for the computer, and has read-write access to all of RAM. CPU number 2 is the one connected to the network. It has read-only access to the chunk of memory from which it runs programs. But it has read-write memory for storing data, and read-only access to a hard drive containing the Web site it is to present to the Internet. If it also has data to store, it gets write access to a hard drive for that purpose. The access is determined by *hardwired connections*, not by operating system privileges which can be subverted.

> In most operating systems, either the Microsoft ones or the Unix clones, networking is part of the operating system, and the TCP/IP connection to the Internet is part of that network. It has to be explicitly limited in its privileges, and if someone gets Administrator privileges/root access, that can be overturned. That shouldn't happen, but any bug in the OS is a possible back door.

> Now, suppose instead that the OS didn't even HAVE networking in it. The port connected to the Internet was something the OS didn't even know about, and everything that port did was under the control of one unprivileged *applications program*. Even if the OS didn't even have security — say it was MS-DOS — with precautions against such attacks as buffer overrun, an applications program with narrowly focussed capabilities could be quite secure.

> If one doesn't go to these kinds of lengths, though, while it is true that constant vigilance and the use of more conventional security methods (i.e. firewalls) can give "pretty good" security, I think the Pentagon is entirely justified in taking the attitude that the kind of *ironclad* security they need just isn't available if one connects to the Internet.

> I'm quite sure that the NSA or whoever could come up with a "super-firewall" that could act as a public Web-site host, and yet be updated from within a highly sensitive computer network, with safety. But it would take technologies like the two-CPU sketch above, which just aren't available off the shelf. And it's off-the-shelf technologies that have been used for much of the military's Internet presence.

> So while it is true there is a way for the military to stay on-line and maintain security, it is also true that that is not immediately available. Taking some Web sites off-line until the vulnerabilities can be remedied isn't a silly policy, even if there may be some individual examples of cluelessness where sites involving no exposure are taken down.

> **From: dragon@revealed.net Subject: Re: Major cluelessness alert**

> I just read your blurb on the Army's consideration of pulling off of the net, and I felt I had to comment. In particular, I disagree with the page which you felt had "a good analysis of this idiotic idea".

> While I agree that a simple knee-jerk reaction to shut off the Internet connection just because X company did so is not prudent, I do believe that, in an organization with an educated security staff, there is a place for a temporary shut-down of the connection. In particular, I was involved in making this decision for one of the companies I work with, and we were concerned with two points: 1) since Melissa was propagating via e-mail with little human intervention, we decided to cut off access until we had gotten enough control on our internal population to not propagate to our business partners in the way that other large companies had done to us, and 2) to give our admins the breathing room to be able to rationally understand what the impact on our production systems were and to implement the updates/fixes that were coming to us from our suppliers.

> I don't know how anyone can say that it's idiotic to disconnect from the

> Internet when in the face of an attack which is both significant in scope and relatively unknown in implementation. Yes, it could be considered to be paranoid, xenophobic, and reactionary, and it's true that it is not necessarily any safer to be connected on any other day, but to deny a security staff the ability to raise the drawbridge until the immediate threat is at least understood hoodwinks us to the point that we won't really be able to function.

> Finally, I have to say that I agree with at least a part of the military's decision to pull back. The one thing that they mentioned was that they were attempting to correct the positioning of sensitive data. There is a lot of information, military or otherwise, that has no place on the public Internet. The running joke in our department is that the only secure computer is one that is powered off, melted into slag, encased in concrete, and buried at the bottom of the ocean. Your own writings show that not even cryptography is completely reliable due to advances in mathematics and side-channel attacks. There are many, many circumstances where the sensitivity and criticality of data demands location on a network that is air-gap protected from others, whether those other networks are the public Internet, less-secure Intranets, or private WANs connecting to suppliers and dealers. The real idiocy is placing data which needs to be kept secure on machines which are accessible via public, or near-public, channels.

> **From: Jon Williams <dragon@revealed.net> Subject: Cracking Encrypted ZIP files**

> Regarding encrypted ZIP file cracking:

> While brute forcing the password may work most of the time for most people and take less time, there is also a known-plaintext attack, which only requires 13 known bytes. Check out http://www.unix-ag.uni-kl.de/~conrad/krypto/... for a whitepaper describing the attack and working software. I've successfully used this.

> **From: “David Brownell” <david-b@pacbell.net> Subject: SSL at Wells Fargo**

> Wells Fargo’s on-line banking site is still using SSL v2 ... doesn’t support browsers configured to use more secure versions (v3, TLS) and has even rejected SSL v2 connections that don’t use RC2 (deprecated). I’m sure you understand the SSLv2/RC2 issues, even when 128-bit keys are in use; they’re just not as strong as other protocols/ciphers, at least for the front-door sorts of attacks that were NOT your point.

> The “simple” bungle on their site, however, is that if you’ve adopted a policy that you’re not going to use SSLv2 for “secure” transactions, the Wells Fargo site says to you that your browser isn’t secure enough, and you need to get a 128-bit browser. Doesn’t say “you must enable an obsolescent version with a dubious cipher” ... which it could say, very easily. It says something completely wrong.

> That was a useful collection of basic bungles. Don’t forget the other type, using an HTTPS page that’s got sensitive data in query params for its URL, and an http://... link that’ll cause that sensitive data to be logged in what are usually insecure logfiles. (No current examples handy — but if you see one of those, it’s classic!)

> **From: David Crick <dacrick@cwcom.net> Subject: SSL at BT**

> British Telecom (BT) are another company with worrying views on Internet security. You’d think with their image and standing that they could do better.

> Their e-services Web page [www.bthome.com/e_services/index_sh.html] allows home users to check and amend various account details and services.

> But despite the spread of strong crypto Web-browsers [www.opera.com] and security upgrades for IE, Windows and Netscape [www.replay.com], BT only chose to use 40-bit SSL.

> This is accompanied by the following endorsement and warning:

> "When ordering goods and services make sure the Web site you are using uses a 'Secure Socket Layer (SSL)' session. The BT Shop – At Home uses such sessions from the moment you start to place an order."

> Also: "If you are still uneasy about using the Web to order on-line then you should use an alternative method of ordering."

> Hardly inspiring, is it?

> It also makes one dubious about their "Secure Site Programme":

> "Trustwise Secure Sites use a BT Secure Server certificate to establish proof of identity of the owner of the Web site and enable secure communication between the Web site and visitors to that site.

> "BT carefully checks the identity of the organization that owns the Web site and verifies that the Web site is registered to that organization.

> The BT Trustwise Secure Site Programme allows you to learn more about the Web sites you visit before you submit any sensitive or confidential information."

> Again, I could only find 40-bit SSL in operation, despite the "Trustwise" logo [e.g. see http://www.bt.com/Talk/].

> **From: Ross Anderson <Ross.Anderson@cl.cam.ac.uk> Subject: AES**

> NIST has just announced that the finalists in the Advanced Encryption

> Standard competition are MARS, RC6, Rijndael, Serpent and Twofish. That makes three U.S. algorithms, one Belgian, and one which I developed in collaboration with colleagues in Israel and Norway.

> It may be of interest that, under the export controls on intangibles which England's DTI pushed in their recent White Paper and which they are now trying to have adopted as an EU regulation, I would have needed a personal export licence from the DTI in order to do this work.

> It seems somewhat unlikely that a licence would have been granted. Arms exporters complain to me that DTI officials are notorious for blocking licences to punish them for such 'offences' as complaining about the licensing process. So perhaps I would have not done the work; perhaps I'd have defied the law and now be involved in a huge test case in the European Court; perhaps I'd have emigrated; perhaps we'd just not do research in collaboration with foreigners. Who knows?

CRYPTO-GRAM is a free monthly newsletter providing summaries, analyses, insights, and commentaries on cryptography and computer security.

To subscribe, visit http://www.schneier.com/crypto-gram.html or send a blank message to crypto-gram-subscribe@chaparraltree.com. To unsubscribe, visit http://www.schneier.com/crypto-gram-faq.html. Back issues are available at http://www.schneier.com.

Please feel free to forward CRYPTO-GRAM to colleagues and friends who will find it valuable. Permission is granted to reprint CRYPTO-GRAM, as long as it is reprinted in its entirety.

CRYPTO-GRAM is written by Bruce Schneier. Schneier is founder and CTO of Counterpane Internet Security Inc., the author of "Applied Cryptography," and an inventor of the Blowfish, Twofish, and Yarrow

algorithms. He served on the board of the International Association for Cryptologic Research, EPIC, and VTW. He is a frequent writer and lecturer on cryptography.

Counterpane Internet Security, Inc. is a venture-funded company bringing innovative managed security solutions to the enterprise.
http://www.counterpane.com/