;login:

THE MAGAZINE OF USENIX & SAGE

October 2001 • Volume 26 • Number 6

inside:

CONFERENCE REPORTS

13th Annual Computer Security Incident Handling Conference (FIRST)

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

conference reports

This issue's reports are on the13th Annual Computer Security Incident Handling Conference (FIRST)

Our thanks to the Summarizer:
Anne Bennett

13th Annual Computer Security Incident Handling Conference (FIRST)

Toulouse, France

June 17–22, 2001

Summarized by Anne Bennett

The Forum of Incident Response and Security Teams (FIRST) is a global organization whose aim is to facilitate the sharing of security-related information and to foster cooperation in the effective prevention and detection of, and recovery from, computer security incidents. It holds several technical colloquia each year which are open to members only, and one annual conference which is open to all.

TUTORIALS

Legal and Operational Issues Affecting Evidence Preservation and Recovery in Intrusion Cases

Byron Collie, Wells Fargo Services Company, USA; Steve Romig, Ohio State University, USA

Computer forensics is defined as the process of identifying, preserving, analyzing, and presenting digital evidence in a manner that is acceptable in legal proceedings. A number of computer and network intrusions (and other crimes which make use of computers in some way) cannot be successfully prosecuted because the evidence has been lost, destroyed, or mishandled.

Planning for correct incident handling includes not only acquiring relevant tools and making sure that staff know how to use them, but, possibly more importantly, putting in place the organizational structure which will make it possible for people to act quickly: for example, identifying who has the authority to release log information, start network monitoring, or make the decision to investigate or to prosecute.

Computer evidence is volatile and fragile; as soon as an incident is suspected, immediate action must be taken to preserve the evidence.

It can be hard to decide whether it is best to shut down the computer gracefully (risking booby traps which may have been inserted into the shutdown sequence), just kill the power (risking losing valuable data as well as compromising the integrity of the file systems), unplug it from the net (risking a possible "dead man switch" inserted by the attacker which could delete all data if the network becomes unreachable), or leave it running (risking further damage or liability to other parties). Just don't reboot: that's the worst choice of all, as it is likely that the intruder has installed programs that will start at boot time, and /tmp will be cleared and other information may be overwritten.

If possible, acquire the volatile evidence, such as the list of open network connections (with netstat), the process list (ps), the list of open files (lsof), and so on. But at the same time, be aware that everything you do risks destroying evidence – for example, by overwriting parts of memory or the swapfiles. Make sure you document everything you do, but not on the compromised system! A tape recorder may be helpful at this point. One critical piece of volatile evidence is the clock drift (difference between system time and "real" time), without which it may be impossible to later correlate timestamped log entries from various sources. Do *not* change the clock!

Also, take copies and MD5 (or better, SHA-!) checksums of relevant data, checking that the checksums of the original and copy match. Sign and date these checksums, possibly using a digital timestamping service, and place evidence, checksums, and signatures under lock and key. With respect to what to copy, a bitwise copy of the entire hard disk is best, followed by a bitwise copy of the file systems, followed by copies of files.

File extracts are unlikely to stand up in court.

Log extracts may be helpful when *presenting* evidence, but complete records must be submitted to the court; some courts will accept log files in digital form (CD-ROM), while others will insist on inches of printout! Note in particular that IDS logs are incomplete, and while an IDS is a great burglar alarm, better and more complete sources of evidence exist in the form of system logs, router and terminal server logs, etc.

Once you have documented the scene (i.e., noted any users at the computer, which switch port the host is plugged into, etc.), acquired whatever volatile information you can, made copies of the disks, and taken whatever actions are necessary to exercise due diligence with respect to your liability (e.g., protected your information and services, as well as any third parties that may be involved), it's time to analyze the data.

Since you'll be analyzing copies of the data, store the original data (disk images from compromised hosts, router logs, terminal server logs, etc.) in a safe place to ensure that you preserve the continuity of the evidence (i.e., you protect it from tampering). Data analysis requires a reasonably deep understanding of how evidence is created, what might be missing, and what can go wrong. For example, a particular entry in a UNIX wtmp file might indeed correspond to a user login, but it might also have been faked by an intruder, and even if not, there's no guarantee that the account owner was the one who logged in.

When correlating logs from various sources using timestamps, be aware of (and correct for) clock drift and time-zone disparities. Also, be aware that some activities are logged when they begin (such as tcp_wrappers logging the start of a Telnet session) and some when they end (wtmp contains the time when a login successfully completed). And, of course, take into account that the logs themselves may have undergone tampering by the attacker: they may have been overwritten, or the software that produces log entries may have been modified. Syslog-type logs sent over the net use UDP and are subject to data loss and spoofing. Guard against these problems by using data from as many different sources as you can. If you've been logging to a secure log server, all the better.

As for analyzing the contents of a disk, be aware that many tools exist to help you reconstruct deleted files: Farmer and Venema's Coroner's Toolkit, for example. You'll be looking for the "standard" stuff, such as specific text fragments pertaining to your incident, IP addresses, email messages, and so on. Some people have found it useful to build a Web-style index on the files on disk and search that!

Don't neglect the backup tapes (you protected them, right?); they can show you when and how certain files changed. File checksums (such as Tripwire) are an invaluable aid – especially if you have checksums of the known "clean" system or of a virgin system — but if not, it's still useful to compare data from backups.

Investigating Malware Incidents

Christine M. Orshesky, i-secure Corporation, USA

A large majority of sites seem to be making some use of antivirus software, yet in recent surveys, well over half had suffered malware infections. Some of this is explained by incorrect use or infrequent updates of the AV software, but one must remember that there will always be a time lag between the launch of a new virus or worm and the availability of countermeasures from the AV software vendors.

Malware was defined as software, firmware, or hardware that is intentionally introduced into a computer system for unauthorized purposes, usually without the knowledge or consent of the user. This session covered software instances only. Note that the malware may or may not inflict actual damage.

Malware was classified as:

- Viruses: attached to an existing file, such as a diskette boot sector (boot sector virus), a program file (file infector virus), or a document (macro virus).
- Worms: self-contained programs which spread from system to system, usually over the network, often using the email system to propagate themselves.
- Trojans: programs which masquerade as a legitimate program to trick the user into invoking them.
- Hoaxes: malware warnings which count on human intervention to remail them to large numbers of people. While they do no direct damage, the resulting volume of email traffic can cause problems. (Summarizer's note: we have recently seen "virus warnings" which trick the receiver into manually deleting legitimate files!)
- Logic bombs: unauthorized code introduced by the programmer of an application, which performs some action based on a trigger event. For example, upon finding that the author is no longer on the company's payroll, they might destroy all of the business records.
- Nasty or joke programs: (no definition).

Many tools, including the well-known anti-virus programs, are available to help combat malware. These range from scanners (which look for known attacks) to heuristics-based behavior checkers (which can detect unknown attacks but suffer from false positives) to file

Malware Name	Year Created	Time to Become Prevalent	Type	Cost of Damage
form	1990	3 years	boot sector virus	$50M over 5 yrs
concept	1995	4 months	Word macro virus	$50M
Melissa	1999	4 days	email-enabled Word macro	$93M to $385M
LoveLetter	2000	5 hours	email enabled script	$700M

Changes over time in malware

integrity checkers (which can report unwanted changes but not how they happened). In addition, firewalls and router filters can block access to known problematic sites or traffic types (to stop the delivery of viruses), intrusion detection systems monitor the network for signs of compromised computers, and content filters are used for files downloaded from the Web.

HOT TOPICS

S/MIME Integration in SYMPA Mailing List Management Software

Olivier Salaün, CRU – Université de Rennes I, France

SYMPA is a mailing list management program which is available under the GPL license; it is used by about 2000 sites. Release 3.0 features:

- Authentication for submission based on S/MIME signatures of messages.
- Encryption for outgoing messages: decrypt once upon receipt with the list key, encrypt outgoing messages for each recipient with each recipient's key.

SYMPA itself uses an RDBMS to store mailing-list information, to ensure performance and scalability. It has a Web interface for both subscribers and list owners; it has a shared document repository; and it has been localized to several languages.

OpenSSL is used to implement certificates, and user X509 certificates are used not only in the S/MIME messages but also for authentication in Web interactions. Note that CRLs (certificate revocation lists) are not yet implemented.

Authentication customization is available per list, per command; it defines who (subscriber, list owner) may do what (subscribe, review, send), and the authentication methods accepted (SMTP, MD5, S/MIME). Note that PGP is *not* implemented.

For more information: *http://www.sympa.org/*.

SRMail (Secure Remailer)

Cory Cohen, CERT-CC, USA

Goal: to address the needs of the FIRST community mailing lists, and to avoid sharing a quarterly changed symmetric encryption key.

Problems:

- Many incompatible mail encryption methods (try to support as many as possible)
- Changing technologies (e.g., many versions of PGP, which keeps evolving)
- Implementation incompatibilities (e.g., different PGP packet formats).
- Key management problems
- MIME is not universally adopted, but is desirable. Moving to it while maintaining backward compatibility is hard
- Scalability problems: must be able to send 1000 differently encrypted messages
- Security concerns: implementation must be solid and reliable.

SRMail can:

- Send signed and encrypted form letters.
- Decrypt encrypted data, verify signatures.
- Manage contact information and encryption keys (associate encryption keys with organizations).
- Manage access to encryption keys, especially shared keys.
- Manage an encrypted mailing list: the sender signs the message and encrypts it with the mail-server key, the server decrypts the message and validates the signature, then it re-encrypts and signs the message for each recipient.

IPv6 Migration and Security

Jean-Jacques Bernard-Gundol, Hervé Schauer, Consultants, France

IPv6 is the next-generation Internet protocol; it has new features and new security issues. The IETF has proposed several possible migration methods from the current IPv4 to IPv6, and some of these are vulnerable to problems. For example, if we tunnel IPv4 inside IPv6, then someone can insert a bad IPv4 address into the inner packet, and the IPv6 stack will blithely unpack and use that address; similarly, tunneling may bypass "usual" checks. NAT-style solutions (with protocol translation) are quite vulnerable to denial of service. IPv6 supports multi-homing in a way that may make ingress/egress filtering much more difficult.

Authorization and Privacy of Internet Applications

Yves Deswarte, LAAS-CNRS, France

Good security can breed a need for better privacy; DDoS, e-commerce fraud, and transnational crime have spurred the development of better security practices, including more reports and moni-

toring. It has become easier to collect private information, which, while it may enjoy legal protection, is rarely protected in practice. There's no economic pressure for privacy; security research is funded by security agencies, not by civil libertarians!

Client-server implementations decide to grant or deny access based on identity, so information about people's identities is collected. Many transactions now involve more than two parties, such as: a customer, a merchant, a bank, a credit card company, etc. Most of these parties do not *need* very much information about the others, but they often collect it anyway; for example, the merchant should not need to know *who* the client is, but only if the payment is OK.

Of course, in the case of a judiciary request (e.g., to prevent money laundering), it should be possible to disclose real identities. The proposed solution involves a set of central authorization servers, each of which holds only part of the information identifying the parties in a transaction. According to this mechanism, only a judge can get enough data to actually decode the information and reveal those identities.

KEYNOTE

Ensure Security and Confidence in Cyberspace, a Priority for France

Henri Serres, Secrétariat Général de la Défense Nationale / Service Central de la Sécurité des Systèmes d'Information (Service du Premier Ministre), France

The French approach to information security was described. France, like many other countries, is undergoing rapid development of electronic technology and is establishing itself as a major player in cyberspace. The government action program for a cyber society has as its goals: (1) to connect everyone, avoiding a "digital divide"; (2) to support French commercial involvement in this new economy; and (3) to improve security and increase confidence in cyberspace.

Protecting the infrastructure and ensuring safe and honest transactions are a strong priority for the French government. French legislation has kept up with new crimes such as malware and unauthorized access. The French government has intervened at the level of personal data protection, implementing the European directives in that area, has recognized the legal value of digital signatures (a decree sets up rules for certificate and signing authorities, again compatible with the European approach), and has fully liberalized its encryption laws, without key length restrictions.

In addition, government enforcement has been strengthened: a central office for high-tech crimes now exists to assist other forces; CERT-A has been created to assist government bodies with computer-related incidents and attacks; a Central Directorate for Information Security now reports to the prime minister and has a role as a national regulatory authority, monitoring, for example, cryptography products for government use (and also the French scheme for certification). The directorate also participates in operational matters, assisting government departments in setting up their network infrastructures.

CSIRT OPERATIONS

Incident Organization and Security Incident Handling

Jimmy Arvidsson, Telia AB HQ – Telia CERT, Sweden

A taxonomy of events was established: event, incident, security incident, crisis, catastrophe. When there is indication of activity, the activity should be categorized into this taxonomy. Appropriate actions can then be identified. Events can be handled if necessary, recovered from, legal action taken if appropriate, and then the whole event can be followed up on to improve procedures.

The author suggests a first level of assessment, where the type and severity of the incident are determined, and an "incident owner" is contacted; the incident owner would be a representative of the entity responsible for the systems affected: for example, a departmental manager, or the owner of the host or information affected. Also at this initial stage, "first aid" might be applied as necessary; for example, in the case of a Web server defacement, the system might be taken "offline" using the DNS.

Then, a second level of assessment is performed. "Events" are merely logged. "Incidents" are handled by permanent operations staff. "Security incidents" might merit the setting up of a virtual CSIRT: a temporary, project-oriented response team whose existence ends with the resolution of the security incident. A "crisis" or a "catastrophe" would be referred to a crisis management team.

The virtual CSIRT draws on existing resources and competence, and can be especially useful when the size or budget of the organization makes it difficult to justify a permanent CSIRT. A security manager might take the role of incident leader, and CSIRT members might be recruited from three groups of people: the incident owner (system owner, departmental manager, information owner), specialists (sysadmins, network admins, central CERT), and administrative people (help desk, lawyers, public relations).

Introducing Constructive Vulnerability Disclosures

Marko Laakso, University of Oulu – OUSPG, Finland

The author is looking for a compromise between full disclosure and non-disclosure models for software vulnerabilities; he proposes "constructive vulnerability

disclosure." The PROTOS project, of which he is a member, studies methods to test protocol implementations for security vulnerabilities.

Consumers continue to be plagued by computer vulnerabilities, many of them avoidable. Poor software quality (leading to large numbers of vulnerabilities based on known trivial programming errors), the inefficiency of the traditional vulnerability reporting/fix/release process (reappearance of vulnerabilities in future releases, small variations which bite multiple vendors, inability of customers to evaluate products), and waste of time (the time used debating full/non-disclosure would be better spent addressing the real issues!) continue to impede meaningful progress in this area.

The goals of the PROTOS project are:

- Low-cost black-box evaluation of products
- Early elimination of some of the most trivial vulnerabilities
- Vendor awareness beyond one particular vulnerability
- Regression testing of future versions
- Customer-driven product evaluation

The author's group created software to bang away at products and report problems; the results are packaged and released initially to vendors, though with the identities of the competitors removed. After a pre-announced grace period, the test suite is released to the public.

Among the first fruits of the project were a test suite for WAP, the Wireless Application Protocols suite, which generated 4236 test cases and tested seven WAP gateway products. All implementations failed at least some of the tests; some implementations failed in half of the 39 test groups. The results were reported to the vendors, along with, privately to each vendor, exploits (DoS in three cases, arbitrary code execution in the other four cases). Vendors had a grace period of at least 51 days before public disclosure of the test suite, and the entire process took 86 days. Vendor responses ranged from absolute inaction (in two cases) to prompt patches and advisories; a few vendors were even motivated to review their code more thoroughly.

For more information on the PROTOS project and its collection of test suites, please visit *http://www.ee.oulu.fi/research/ouspg/protos/*.

(Summarizer's note: a few weeks after the conference presentation, CERT Advisory CA-2001-18, "Multiple Vulnerabilities in Several Implementations of the Lightweight Directory Access Protocol (LDAP)," was released, which listed multiple vulnerabilities in nine different LDAP products, again based on test suites created by the PROTOS project.)

Experience with Abuse Management in Privacy-Enhancing Systems

David Bratzer, Zero-Knowledge Systems Inc., Canada

Zero Knowledge's "Freedom Network" provides anonymous Web browsing and chatting, and pseudonymous email and Netnews services. They tried to design their service to be resistant to abuse, recognizing that it is not possible to prevent abuse completely. They claim a 0.2% abuse rate, or one problematic account in 500.

DENIAL OF SERVICE

DoS Attacks on Transit Networks

David Harmelin, DANTE, UK

DDoS attacks via network flooding were studied. Usually, a master controller sends commands to a number of "handlers," which may in turn contact many compromised hosts to make them run denial-of-service software. The traffic tends to become aggregated in the transit network.

Each router in the transit network logs netflows to a workstation nearby. DANTE wrote a tool which has a central workstation "poll" each of these log hosts every 15 minutes and take a sample of 1/500 of the flows that occurred in a 10-second interval. For each router, an alarm is raised if there are more than 10 flows with the same destination IP address per sample.

About 98% of alarms were confirmed as attacks in progress; the tool can detect attacks at rates greater or equal to 100 packets/second. DANTE found that 90% of the attacks were "C class," i.e., were from a set of spoofed addresses all within the same class C network, to get through the ISPs egress filters. Most attacks (58%) lasted less than 15 minutes.

CSIRT COOPERATION

Collaboration of European Computer Security Incident Response Teams

Gorazd Bozic, Arnes SI-CERT, Slovenia

In the 1990s, an attempt was made to create a "EuroCERT" to coordinate interactions between European CERTs. This was the SIRCE project (1997–1999), which ended with the conclusion that a top-down approach to this task was not suitable.

In May 2000, TERENA established a task force to work with a wider number of individual CSIRTs. Why this in the face of previous failure? This time, a very informal process is being used, with quarterly two-day meetings. Here are some of the initiatives of TF-CSIRT:

- Trusted introducer program (signing PGP keys to identify CSIRTs to other CSIRTs)
- IODEF (Incident Object Description and Exchange Format) – workshops for new staff of CSIRTs
- Cooperation with EU officials (eEurope program) – clearinghouse for CSIRT tools

PROACTIVE CSIRT TOOLS

Experiences with National Wide-Scan Detect Systems

Hyunwoo Lee, Korea Information Security Agency, Korea

In 1999 the author's group experienced a set of automated, stealthy, distributed DoS attacks and, surprised by the scale of the attacks, felt that proactive countermeasures should be deployed immediately.

They found that when they received an alert about a DDoS attack, it was already too late to intervene. Traditional "passive" responses are ineffective and fail to stem the flow of attacks – manual email response is much too slow, and attackers have nothing to fear from the CSIRT community.

Port scanning is the initial step in attack preparation, so automatically detecting and reacting to port scans could help – though it is necessary to share that information with other members of the security community.

Realtime scan detection agent software was deployed to run at various end sites and report to a central data collector. Also, a system of alerts was developed within the community, where known incidents are reported as formal alerts, and suspicious occurrences as informal alerts.

It became possible to collect statistics of scan incidents and, using them, to detect and identify previously unknown attacks – for example, a sudden increase in scans on port 12345 was found to correspond to the Detlog worm. But the greatest gain of this project was the formation of a cooperative security community.

The CyberAbuse Project

Philippe Bourcier, XP Conseil, France

The CyberAbuse project developed from some IRC Undernet projects to prevent IRC abuse, especially DoS. The first such project, "Abuse-DoS," found "smurf amplifier" routers and sent information to administrators about the correct configuration of routers in this respect. Of the 190K misconfigured networks found, 25% were fixed after the project sent mail to the admins.

Another project, "Abuse-Proxy," addresses the problem of open IRC proxies, which provide anonymous IRC connections. The proxy scanner detects open proxies, and email is again sent to the admin of the misconfigured host.

In the "anti-hack" project, certain IRC channels were monitored for DoS tools and Trojans. Admins of victim sites, as well as CERT, were warned when compromised hosts were found. Problems were fixed by admins 80% of the time.

The CrimeWatch project makes available to security professionals and law enforcement agencies information about criminal groups and activities as well as new techniques.

Automated Incident Report Processing and Cross-Correlation of Probe and Scan Information

Mark McPherson, University of Queensland, Australia

CSIRTs receive numerous reports of many kinds of attacks, such as probes/scans, access, compromises, DoS, viruses, and spam.

Probes can indicate preparation for attack, or they may be a cover for some other attack in progress. The sheer number of scans provides a smokescreen which makes it quite hard to see what's really going on. Cross-correlation of logs across multiple sites can help pinpoint the "real" attacks; CSIRTs are the logical choice for collecting those logs, since they have already established trust relationships with their communities.

AusCERT created "probelogger" to collect, process, and acknowledge probe reports sent in by various sites. There is even a function to optionally report the scan to the originating site. If the origin of the probe is an AusCERT member site, the software raises a flag so that the incident receives special handling.

INTRUSION DETECTION

A Protection Mechanism for an Intrusion Detection System

Takefumi Onabuta, Information-Technology Promotion Agency, Japan

If the host on which the IDS is running suffers a system-level compromise, it is impossible to protect the IDS files and processes from the attacker. Thus, a kernel-level approach was taken, where mandatory access controls are implemented.

An access-control model (LOMAC) was considered which defines low- and high-security levels, and assigns these levels to subjects (processes) and objects (files); a low-level subject is prohibited from accessing a high-level object. Problem: system logs are written by low-level (userland) processes but read by high-level (IDS) processes – thus the information in the logs is not protected.

Another access-control model (LAM) was considered which defines different limitations on access to objects: read-only, write, append, create, delete, link, modify, execute.

The authors created a hybrid of LAM and LOMAC, called E-LOMAC (extended LOMAC), which not only permits access to high-level objects by low-level subjects, but limits even high-level access to specific operations. The new access-control system was tested on a host running a host-based IDS; most attacks were stopped. The system was benchmarked for performance, and it was shown that the impact was minimal (98.32% and 85.20% of no-E-LOMAC performance). E-LOMAC seems to successfully protect a host-based IDS from

being disabled by an attack, but it is fairly difficult to configure.

SECURE PRACTICES

Securing Web-Based Applications with Hole-in-the-Chroot

Anne Bennett, Concordia University, Canada

A scheme was presented whereby it is possible to run a Web server and CGI scripts in a UNIX "chroot" environment and yet communicate with applications outside the chroot using files and named pipes. A daemon running on the "system" side listens for requests from the "chroot" side by monitoring a named pipe. When a request is received, it is checked against a list of defined job names, and the incoming data is checked before being passed to (possibly fragile) applications on the system side.

OS Fingerprinting

Franck Veysset, INTRANODE, France

Techniques for discovering the OS and version of a system on the Net were presented. They ranged from grabbing banners (Telnet greeting, HTTP header), to analyzing executables if such could be obtained (such as /bin/ls from an anonymous ftp server), to observing the behavior of the system's TCP/IP stack, especially in response to malformed packets but also with respect to TCP sequence numbers.

Panel Discussion: Ask the Experts

Moderated by Roger Safian, Northwestern University, USA

Q: Monoculture on the desktop (Microsoft) has caused a rash of problems; will monoculture on the Net (Cisco) do the same?
A: Juniper is starting to take some market share from Cisco. Within an organization, one must weigh the risks of such monoculture with the cost benefits in terms of ease of administration.

Q: What's the biggest bang for the buck in terms of securing computers and networks?
A:

- Assigning responsibility for keeping things up-to-date
- Sending people to conferences to establish networks of people who've "been there, done that" and who could be asked for advice or validation in difficult circumstances
- Assessing risks of highest impact; assigning someone to take the time to follow the security announcement mailing lists
- Use of digital signatures recommended to assist in detecting intrusions and recovering quickly from them

Q: What about kernel-mode rootkits?
A: They are out there and are quite effective, and their installation by crackers (including via worms) is increasingly smooth. Beware: NetFlow and traffic analysis are likely to detect them using unusual ports.

Q: If you had sensors sampling information about a network, what would be the most useful piece of information to have?
A: Analysis of flows before and during an event is the best; note that many IDSes do not provide that level of detail. Network flows are the best tool; they can be pumped through MRTG to see trends.

Q: How are the areas of incident response and viruses converging?
A: Worms combine the two; we see more and more overlap between viruses/intrusions and the use of the network. More cross-pollination is needed between the IDS and anti-virus industries.

POST-MORTEM ANALYSIS

Disk Analysis Hurdles

Philippe Bourgeois, CERT-IST – Alcatel, France

Disk analysis is sometimes required during an investigation of a compromised system or a legally seized system; this task is becoming more popular, and tools (such as The Coroner's Toolkit) are becoming available. Still, many things can go wrong or cause problems.

You may have trouble getting the disk image without cooperation from a sysadmin; you may have to bypass the BIOS protection to boot from alternative media (it would be dangerous to boot from a "hostile" system), or just move the disk to another host if that is possible.

Sometimes the file system is unreadable; you'll have to use data recovery tools to try to reconstitute files from blocks of data on the disk.

Dealing with large disks can be a problem, and disks are getting larger all the time. To reduce the forensic effort:

- Focus the investigation on a specific set of files.
- Discard from consideration all "known good files," based on MD5 signatures of the OS and applications, and analyze only unknown files. This can easily remove most files from consideration.
- Try indexing the data on disk to speed up searches.

When faced with encrypted data, check for weak encryptions which are easy to break. If necessary, try a brute-force approach to guess the key. However, be aware that these efforts may well fail. Don't forget that the plain text may be somewhere on disk as a deleted file or part of a swapped-out process.

INDESTRUCTIBLE INFORMATION

Wietse Venema, IBM, USA

Although commonly received wisdom suggests that it is very hard to recover a deleted file (since its blocks are reallocated and often overwritten), it turns out that data on disk can be read, assuming appropriate equipment, even after having been overwritten several times.

Sorting files (including reconstituted files) by time (access, modify, create) can often show what happened on a system; for example, access to compilers, libraries, and header files shows a compilation. Of course, bear in mind that file times can be forged! Linux rootkit v4 has a "footprint" of about 800 file changes, of which about 460 are deleted files (probably rootkit source).

In practice, the longevity of deleted files can be quite significant; a 10-month-old machine was examined, and numbers of deleted files (by age in one-month increments) ranged from 172 at four months to 51205 files at 10 months. In one case, a compromised Linux honeypot was examined, but traces of its previous lives running Solaris (including a firewall config file!) and Windows 95 were found in unused space.