



Responding to a Customer's Security Incidents—Part 4: Processing Incident Data

Vijay Masurkar—Sun Services

Sun BluePrints™ OnLine—October 2003



<http://www.sun.com/blueprints>

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
(650) 960-1300

Part No. 817-4002-10
Revision 09, 11/10/03
Edition: October 2003

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95045 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, SunDocs, Sun Fire, Sun Ultra, Sun Enterprise, SunSHIELD, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the US and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95045 Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuelle relatants à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Sun, Sun Microsystems, le logo Sun, Sun BluePrints, SunDocs, Sun Fire, Sun Ultra, Sun Enterprise, SunSHIELD, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Please
Recycle



Adobe PostScript

Responding to a Customer's Security Incidents—Part 4: Processing Incident Data

During or after an incident, collecting, processing, and developing evidence for law enforcement or for internal use is one definite goal for any security incident response team. Even if you choose not to report the incident, processing the incident data contributes to learning from the incident and preventing future incidents. In addition, one of the many encouraging aspects of the worldwide incident response community has been its willingness to share technical ideas and information from similar or past incidents among teams and individuals. This information, if stored and processed diligently, can be extremely helpful in possible future incidents.

The first three articles in this series discussed establishing a computer security incident response team (CSIRT) and policy, executing the policy, and following up after an incident, which included acquiring incident data. This fourth article focuses on authenticating, preserving, and analyzing the incident data. Here, only the salient points for best practices that should be executed in processing the incident data are discussed. These practices are typically preceded by a recovery phase and are only starting points for a more detailed analysis for building a policy with the associated processes and procedures. The scope of best practices described in this article is limited to computer and network systems data that is related to security incidents.

Security Incident Response (SIR) is the combination of resulting processes and actions an organization takes in responding to a security incident. It should be obvious that each and every security incident response program will contain unique elements that exist and make sense only for that organization.

Before you read this article, you should be familiar with the concepts described in the first three articles. This article is intended for computer security managers, security policy developers, system administrators, and other related staff, who are responsible for the creation or operation of a computer security incident response policy and service.

Processing Security Incident Data—A Monumental Task

A massive amount of information is gathered by the incident response teams. It could be collected from incident triage; interviews; conversations with the affected people; electronic discovery process; if the analysis is conducted by legal means; and various kinds of analysis of computer networks and systems at the constituent's site. Tools for entering, accessing, and tracking information and events can greatly facilitate, and at least partially automate, data manipulation and searches. Such tools can support the staff of any organization and its VCSIRTs in helping to establish, for example, the identification of the following:

- New events such as incidents and service requests (for example, from constituents or other CSIRTs)
- Current events that are tracked separately, but might have some relationship to the incident at hand, and ongoing corroborations between them, as deemed necessary
- Information directly related to current events (for example, captured images of the compromised systems or a set back in the litigation process)
- Information directly related to a previously closed event or to an incident that is similar to the current one (such as, a rootkit found on the compromised system)

How to gather, store, and retrieve all of this information is a topic of research by itself and will not be covered in this article. However, at this point, you should know that in addition to the tools that help in performing the types of tasks listed above, it is expected that a servicing organization should promote the use of de facto standards, such as incident tracking numbers, standard reporting forms, and preregistration of needed contacts, for incident processing by the VCSIRTs that the servicing organization deploys. In addition, it is important to work in harmony with the external CSIRTs (national or international) so that the standards used for incident data processing by a VCSIRT are in tune with those of the other response teams. For example, to track incidents, both the CERT/CC (<http://www.cert.org>) and the DFN-CERT (<http://www.cert.dfn.de/>) response teams allocate integer numbers within an agreed upon integer range.

Authenticating the Incident Data

As a general rule, all criminals or perpetrators leave evidence behind. What you have collected as evidence during and/or after an incident must be proven to be the same as what was left behind during a criminal or unauthorized activity. Both proof of integrity and time stamping are provided by calculating a value that represents an electronic footprint. This is a cryptographic technique, and the value is called a hash. All forensic utility suites include the software to calculate hash values.

MD5 (Message Digest), as described in IETF RFC 1321, and SHA (Secure Hash Algorithm) can be used to create a hash of the entire diskette, or hard drive, or individual files, as needed.

Note – The MD5 algorithm takes a message of arbitrary length as input and produces a 128-bit *fingerprint* or *message digest* output. It is conjectured that it is computationally infeasible to produce two messages having the same message digest or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications in which a large file must be *compressed* in a secure manner before it is encrypted with a private (secret) key with a public-key cryptosystem, such as RSA. In essence, MD5 is a way to verify data integrity and is much more reliable than checksum and many other commonly used methods. SHA is the basis for the U.S. Secure Hash Signature Standard SHA-1. This standard is also used for computing a condensed representation of a message or a data file. SHA-1 produces 160-bit output, called a message digest, of any message that is less than 264 bits. SHA-1 is called secure because it is considered computationally infeasible to find a message that corresponds to a given message digest or to find two different messages that produce the same message digest.

An example of the use of the MD5 utility to create a hash of the suspect system's disk from your system's clean (known good) mounted CD is shown below. Ensure that the object you are fingerprinting is not accepting active writes. The following command assumes that you are on the suspect system and that `sdisk` is the suspect system disk.

```
# /mnt/bin/md5 /dev/sdisk
fd834eafd2546cdbaf09e817645af34d /dev/sdisk
```

On the known good, collection host, where you will create image copies of the suspect disk, run MD5 (again) against the image file, as in the following example.

```
# md5 suspect_sdisk_img
```

Now, comparing the two hash values, you can make sure that the hash value of the suspect disk has not changed in the process of image creation.

A common practice in many tools, such as Tripwire, is to use multiple hash algorithms. For more information, go to:

<http://www.sun.com/software/security/tripwire/>

Make sure you record the values of hashes created so that later in the analysis (and in the prosecution of the incident if it is pursued), you can prove that the copies of the data you are using for your examination are identical to the original data collected as evidence of an incident.

The images of hard drives and any volatile information saved before shutting down a suspect machine are also candidates for time stamping. Some examples are:

- Collection of proofs of ongoing unauthorized or suspect activities, which might include items such as log files, sniffer output, and outputs from firewalls and intrusion detection systems
- Output from any reports or searches performed on a suspect (compromised) machine, including a list of all files and their associated access times
- Typed copies of the investigative team's daily notes

Preserving the Incident Data

Any evidence that can be used in a court of law must be preserved with extra care and security. The corporate security and legal departments for the incident-servicing enterprise should be consulted by the assigned geo-based security officers and the worldwide security manager to review the evidence.

When protecting data as evidence, the basic principle is *do no harm*. Organizations should seek advice on preserving the evidence from an experienced security expert, who might be the assigned geo-based security officer or somebody from outside of the VCSIRT. The last thing you want is to have to replace a confiscated system that was damaged either intentionally or by some unforeseen circumstances.

For example, in an incident in New England, handled by the Secret Services's NET regional task force (<http://www.ectaskforce.org/>), a disgruntled employee changed a router configuration to let illegal network traffic come through the firewall. The employee immediately left the company, suspecting apprehension. Before the company engaged local law enforcement and realized that the router changes were made with a malicious intent, the employee's computer was given to another employee who replaced him. The new employee not only changed the

configuration of the computer, but also fixed the problem in the router before notifying an official of the company or a law enforcement officer. As a result, there was no trace of the actions taken earlier by the attacker.

In another case, a well intentioned technician inadvertently damaged evidence and compromised the chain of custody after the court had permitted expedited discovery on computerized files (see *Gates Rubber v. Bando Chemical*, 167 F.R.D. 90; 1996 U.S. Dist. LEXIS 12423).

The chain of custody typically involves the following key questions:

- Who accessed and collected the data first?
- How was the data accessed and collected (explaining the manual and automated methods used)?
- Where was the data collected, including detailed location information?
- Who took the actual possession of it (for example, the person who accessed it might be different from the person who took possession of it)?
- How was the data stored and protected?
- Who took the data out of storage? When and why was the data taken out of storage?
- Where was it transported to next?

The following table contains an example of a recommended a chain of custody recording:

TABLE 1 Chain of Custody Recording

Item	Date	Time	From Location	To Location	Name	Reason
Sun Ultra-10, serial: 235789	06/30/01	11:21:00	Office 127, ABC Corp., Industrial Park, YourCity, MyCountry		Bledsoe	I took the memory snapshot of this machine before shutting it down using the guidelines. Then, I image copied this web server. Two disks are tagged as "case01-1" and "case01-2." I locked these disks in the cabinet "A-1" in office 127.
Sun Ultra-5, serial: 78901	07/03/01	14:55:00	Office 127, ABC Corp., Industrial Park, YourCity, MyCountry	Office 1000, ABC Corp., Industrial Park, YourCity, MyCountry	Brady	I unlocked Office 127. Tagged and moved the machine and disk 01 to Carlson's office 1000 for further analysis and safekeeping. Rice locked Office 1000.
Sun Fire 15K server, serial: 234567	07/07/01	23:10:00	Lab room 523, ABC Corp., Industrial Park, YourCity, MyCountry	Lab room 601, ABC Corp., Industrial Park, YourCity, MyCountry	Marino	Tagged, moved, and locked up the machine and associated media (disk 1 and disk 2) for next month's government agency review of email archives.
Toshiba laptop, serial: 124783	07/10/01	01:00:00	Home: 123 Ideal Rd., Hometown, HisState, MyCountry	ABC Corporation, Industrial Park, YourCity, MyCountry	McNabb	Moved to office location from the home of employee (101010) for forensic analysis by Carlson tomorrow.

Analyzing the Incident Data

A good analysis of any incident is critical in maintaining the effectiveness and reputation of the response team among its constituents. Although, in the earlier articles, data analysis was depicted as part of the follow-up phase (where it is more prominent), in reality, it occurs throughout the SIR. The following are important points for overall best practices:

- All of the data points must be tracked throughout the incident response process. Data points are in every phase of SIR. After an incident case is opened, it transitions through many states. All of the information relating to the incident, its change of states, and associated actions that caused those changes should be tracked, until no further action is required by a VCSIRT on behalf of its customer.
- During the life cycle of any incident, data analysis provides information that plays a key role in the decision-making process and in the steps to be followed according to the team's policies, processes, and procedures.

The first instance of analysis takes place when the incident is first reported. In the later phases of the SIR, different types of analysis can be thought of as the VCSIRT services, based on the outcome of the analysis. For example, a team could offer an *Artifact Analysis* service for analyzing remnants found after the intruder's exit from a constituent's network. Another VCSIRT could provide a *Denial of Service* or *Unauthorized Access* analysis.

- It is not uncommon that, even after the incident is closed, a team member might be required to respond to a query relating to the incident.

Determining the Response Plan

An initial broad analysis should help you to understand which response plan to follow. This analysis might include forensics, which is described in "Forensics" on page 16. The best practice is to demarcate the response plans clearly so that team resources can be deployed in the right direction. A compromise could result from one of the several types of attacks. For example, an attack could be a denial of service attack (DoS), a virus injection into the network (which might be the result of spam), or an intrusion and subsequent unauthorized access by a perpetrator who had camped out earlier on the customer's corporate network. The flow of the initial broad analysis is depicted in FIGURE 1. The following flow chart includes only the starting points for a DoS and unauthorized access attacks. Other types of attacks are not included.

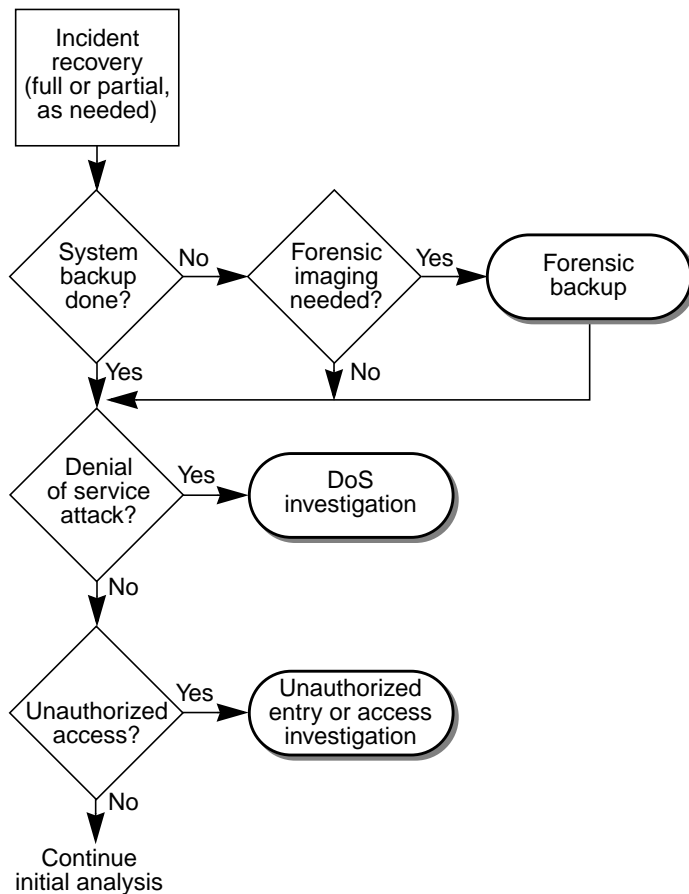


FIGURE 1 Initial Analysis Leading to Different Investigations

The initial broad analysis forms the basis for the next steps of the analysis:

- Determining the type of attack
- Drilling down into the detection, containment, recovery, and other response phases

The detection of suspicious activity (DoS, unauthorized access, or something else, as indicated in FIGURE 1 by the decision boxes) happens by analyzing firewall, IDS, or other log entries and alarms set up to notify the administrator of unusual activities. Common indications that lead to their respective response plans are discussed in the subsequent sections.

The following two sections contain examples of denial of service and unauthorized access analyses. In the flowcharts, the analysis actions are highlighted.

Denial of Service Attack

A Denial of Service (DoS) attack involves a user taking up so much of a shared resource that there is not enough left for other users to use the computing resources. The resources can be networks, processes, disk space, CPU, or modems. Bear in mind that brute force DoS attacks, such as Smurf, tend to rely on spoofed addresses to provide some cover for the attacker. But, be aware that there are also elegant and clever ways that expose only the lower-level footprints. Two examples are Teardrop and Ping of Death.

For more information on Teardrop, go to:

http://advanced.comms.agilent.com/routertester/member/journal/JTC_018.html

For more information on Ping of Death, go to:

<http://www.insecure.org/splloits/ping-o-death.html>

Note the following points for the particular context of the discussion here:

- There are a number of ways to destroy or damage information in a fashion that denies service to users using DoS techniques. For instance, an attacker can flood a network so that legitimate traffic cannot pass through or overload a service on a host so that it cannot respond to legitimate requests. In FIGURE 2, it is assumed that the response team has already determined from the cursory analysis that it was a network DoS attack (Smurf) caused by flooding the network, as noted in the router logs. The following is an example of the log entries:

```
00:00:05:207 spoofed.victim.net > x.y.a.255: icmp: echo requests
00:00:05:215 spoofed.victim.net > x.y.b.255: icmp: echo request
00:00:06:237 spoofed.victim.net > x.y.c.255: icmp: echo requests
00:00:07:838 spoofed.victim.net > x.y.d.255: icmp: echo request
```

`spoofed.victim.net` is the spoofed address and is not the hacker's real source address. `x.y.x.255` addresses are the constituent's target network with incoming ICMP echo requests. The Smurf attack has no effect other than to consume bandwidth.

Other possible indications could be user reports of system unavailability, unexplained connection losses, sudden increase in bandwidth utilization, firewall and IDS log entries, and packets with unusual source addresses.

Hence, a response plan must be put together to quickly respond and recover from the attack and to minimize the damage to the customer's business.

- The steps in the flowchart might apply to a few real-life situations, but they do not apply to all possible situations. Situations vary widely, so the decision points will vary depending on the type of business the constituent customer conducts, the resources available to the servicing VCSIRT, and the intensity and implications of the attack. The flowchart begins after a partial recovery has been completed, as in bring up the network service that had died as a result of the DoS attack.
- In FIGURE 2, the analysis is shown to occur at two junctures. Depending on the available resources and the customer's management decision, the analysis might not be started until the problem has been determined. In addition, the analysis and/or testing might be ongoing or might be accomplished before and after the complete recovery. The analysis steps are highlighted in the diagram.

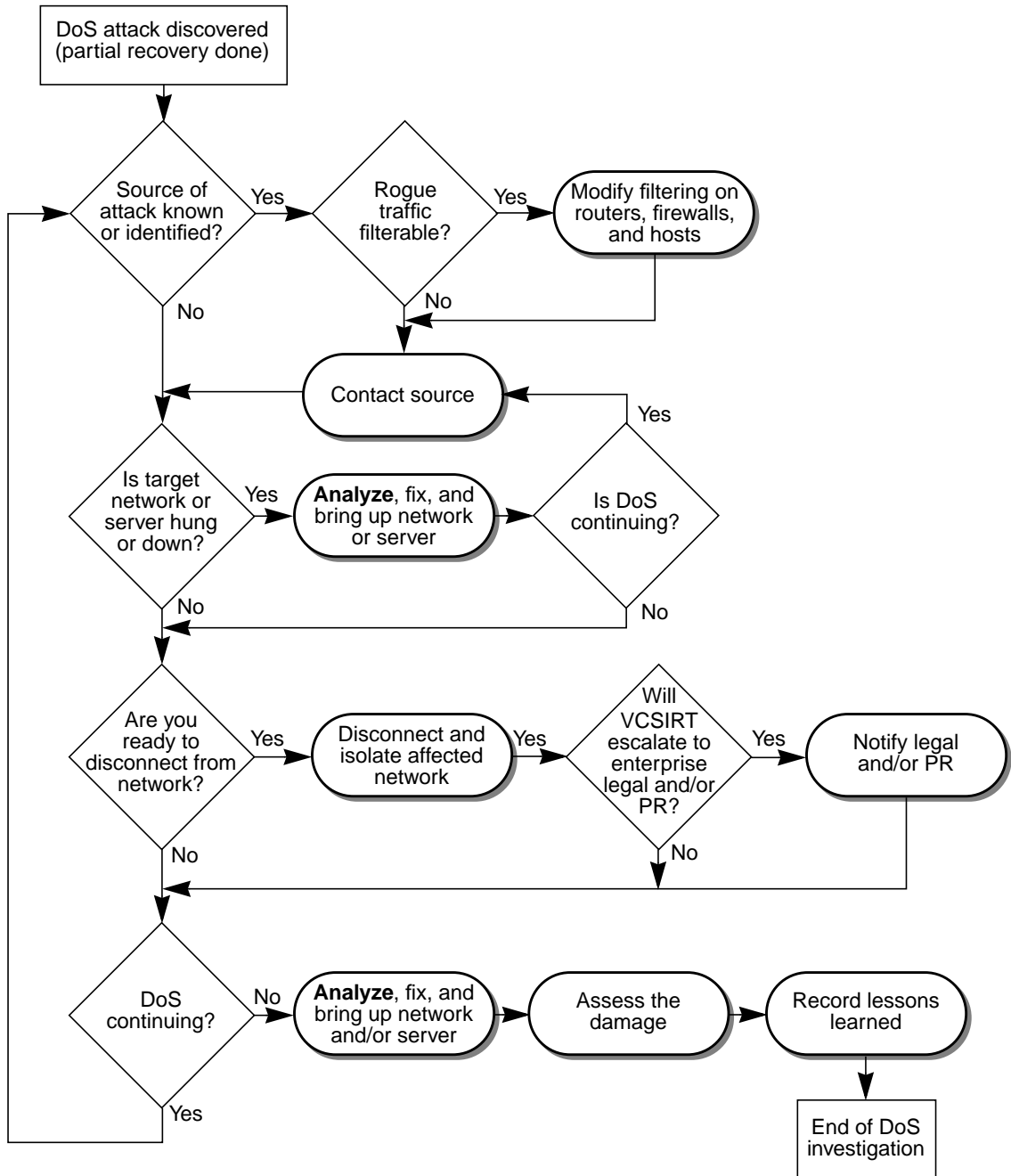


FIGURE 2 Steps for Example Denial of Service Investigation

Primarily, there are two steps to address a DoS attack: 1) egress filtering to stop spoofed IP and 2) stopping or preventing your network from being used as a broadcast amplification site. Analyze the customer's traffic, and see if the DoS attack is continuing. If it continues, you should contact the spoofed source (preferably through the legal and/or PR department) and inform them about the following prevention techniques:

- Use filters to stop spoofed IP packets from leaving your network. This implies that you are preventing your network from being the source of spoofed (that is, forged) communications that are often used in DoS attacks.
- Recommend the following actions to the constituent and all upstream sources:
 - Ensure that your routers and firewalls are configured to forward IP packets only if those packets have the correct source IP address for your network.
 - Deny invalid source addresses. Obviously, the correct source IP addresses would consist of the IP network addresses that have been assigned to your site.
 - Deny private and reserved source addresses (refer to IETF RFC 1918). This can be accomplished by filtering on routers, firewalls, and hosts. It is important to do this throughout your network, especially at the external connections to your Internet or upstream provider.
- Ensure that your network cannot be used as a broadcast amplification site to flood other networks with DoS attacks such as the Smurf attack.
- Recommend the following actions to constituent and all upstream sources:
 - Configure all of your systems (routers, workstations, and servers) so that they do not receive or forward directed broadcast traffic.
 - Disable IP-directed broadcast on all systems.
 - Test your network to determine if it is an amplification site.
 - Require that vendors disable IP-directed broadcast by default, as specified in IETF RFC 2644.

After these techniques have been implemented and the DoS attack stops, proceed with a detailed analysis of traffic logs, bring up any disconnected networks or systems, assess any damage that might have occurred due to the DoS attack, and document the lessons learned.

On the other hand, after containment and partial or full recovery, your constituent might decide to take action against the source (whose address keeps showing up in the firewall traffic) or the perpetrator (whether known or not), so you might have to inform the legal and PR departments of the constituent, as shown in the flowchart. The tracing of a DoS attack might continue indefinitely until the source of the problem is located and the problem is fixed.

Unauthorized Access Attack

FIGURE 3 shows an example of an investigative analysis for a case of intrusion. An unauthorized access to a customer's corporate database that contains sensitive employee data has occurred.

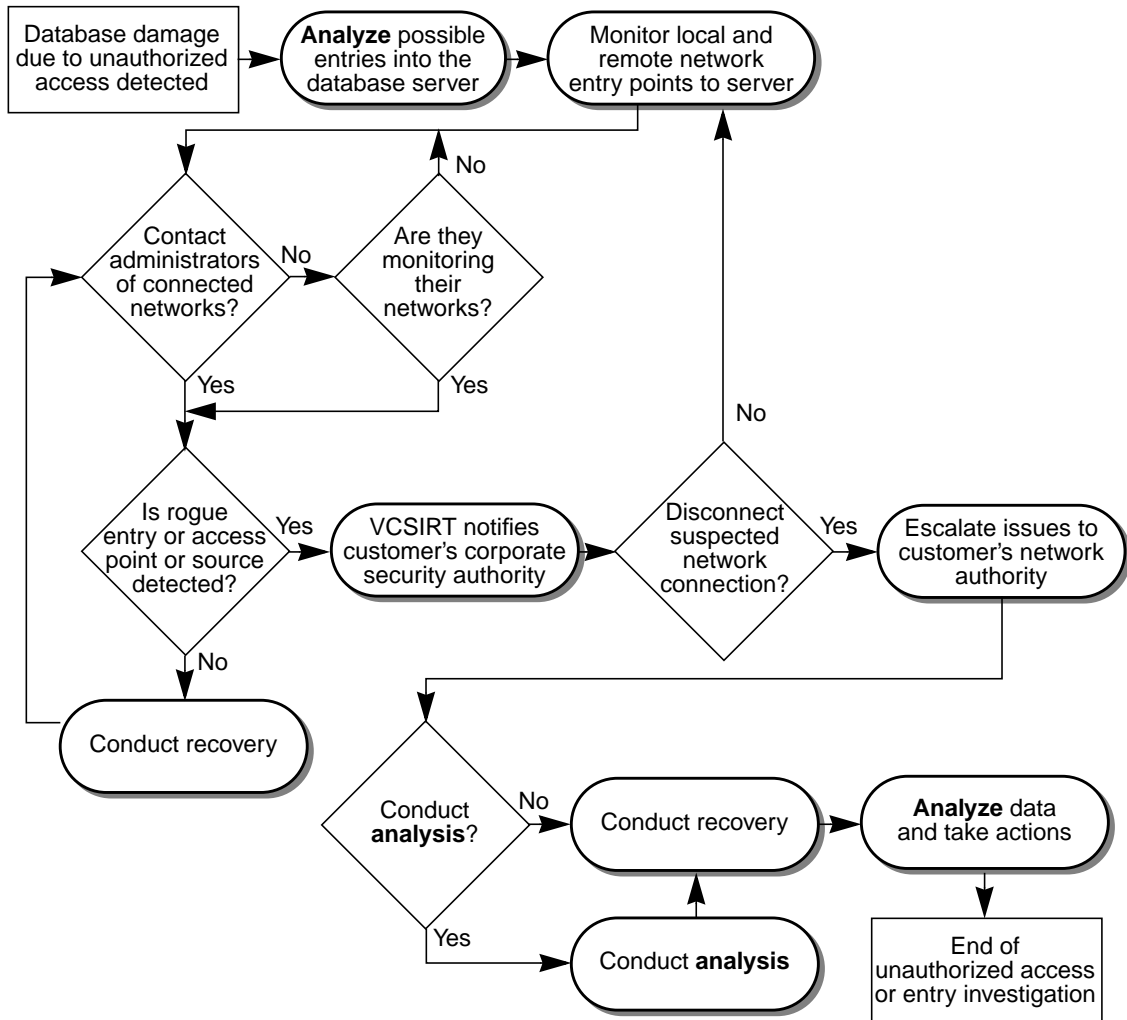


FIGURE 3 Example of Investigative Steps for Unauthorized Access or Entry

Unauthorized access is typically gained through the exploitation of operating system or application vulnerabilities, the acquisition of user names and passwords, or social engineering. Attackers might gain internal access to a constituent's site through one vulnerability and, later, use that vulnerability to attack more vulnerabilities, eventually gaining higher levels of access.

Unauthorized access incidents are often preceded by reconnaissance activity to map hosts and services and to identify vulnerabilities. Initial tracing might happen in different ways. The following are examples, along with the actions to follow:

- Attacker activity, that is traced through analysis, might include port scans, host scans, vulnerability scans, pings, and trace routes.

The following code example shows a host scan using UDP requests. In the following trace, the attacker is targeting multiple network addresses (for example, a.b.c.d, x.b.c.d, and y.b.c.d). Instead of relying on the ICMP echo request to find hosts, this scan is trying to find out if any host will reply on the echo port. `udp 4` refers to UDP payload with 4 bytes of data.

```
04:09:29:0345780 launchpad.scan.com.4555 >a.b.c.d.echo: udp 4
04:09:30:0217620 launchpad.scan.com.4555 >a.b.d.d.echo: udp 4
04:09:31:0456210 launchpad.scan.com.4555 >a.b.e.d.echo: udp 4
04:09:32:0128950 launchpad.scan.com.4555 >x.b.c.d.echo: udp 4
04:09:33:0205580 launchpad.scan.com.4555 >x.b.f.d.echo: udp 4
04:09:34:0121992 launchpad.scan.com.4555 >y.b.c.d.echo: udp 4
```

The echo port echoes back any characters sent to it! (In the first place, system administrators should not have echo port listening.) So, if the server host with the database in question has echo port listening enabled, the attacker gets the response.

All such activity is detected primarily through IDS software, secondarily through log analysis. The response team should analyze traffic and look for distinct changes in reconnaissance patterns such as a sudden interest in a particular port number or host, existence of unauthorized security-related tools or exploits, attempts to gain administrator privileges, or modification of critical system files, timestamps, and privileges.

- Users might report social engineering attempts to get user passwords. The servicing VCSIRT should immediately remind the users about the policy of handling social engineering attempts.

The logic in FIGURE 3 assumes that the constituent has configured network-based and host-based IDS software (such as file integrity checkers and log monitors) to identify and alert on attempts to gain unauthorized access.

In this case, after a rogue entry into the corporate database has been detected, a cursory analysis of all possible network connections to monitor starts before recovery because it is imperative that the unauthorized access be tracked in real time. The response team needs the cooperation from all of the administrators for the connected network to help in the analysis.

A detailed analysis could start after the suspected network has been identified and the issue has been escalated to advanced, network security analysts. However, this is a management decision. If a constituent decides that recovery is more important, then further detailed analysis, damage assessment, and vulnerability patching could be delayed until the recovery is completed. Under all circumstances, the VCSIRT or the security officer involved must advise the constituent that this is risky because after the recovery, you could lose the real-time evidence. In addition, it is possible that the environment is subjected to yet another unauthorized access sometime later. This is because it might still have undetected back doors, rootkits, breached accounts, or applications that could pose harm.

The last box in the flowchart, “Analyze data and take actions,” implies taking actions based on findings from the analysis. For example, you might secure all remote access methods, including modems and VPNs, that were previously using weak authentication, such as easily recognizable passwords. Remote access clients are often outside a customer organization’s control, so granting them access to resources such as the corporate database that stores employee information increases the risk. This could have resulted in this particular type of compromise.

Depth of Analysis

There is a whole range of actions and processes a VCSIRT can execute, depending on the time available to analyze events thoroughly and to disclose the outcomes to its constituent customers and to other teams. Consider the following to understand the best practice approach when it comes to determining the depth of analysis of an incident:

- You must determine the level of depth of the analysis up front.
With a deeper analysis, there will be increasing demand on resources for tasks such as examining log files, analyzing malicious code and software environments, providing workarounds or fixes, scrutinizing customer site security, and actively resolving problems.
- The decision of the level of analysis lies with the team, but the affected customer must be consulted before making a decision.
- You must take a bigger perspective that involves trends, statistics, and case studies.

For example, trends generally involve types of future attacks and security improvements. Statistical research involves the number of networks or hosts affected and the rates of incident reports and closures. Each VCSIRT needs to build this bigger picture for its own constituency.

- Refining the big picture certainly helps to provide key information required by law enforcement in cases where the prosecution process is undertaken. It also facilitates gathering lessons learned.

The following table contains descriptions of the critical factors for determining the depth of the security incident analysis.

TABLE 2 Critical Factors for the Depth of Security Incident Analysis

Incident Analysis Depth Factor	Description
Goals and technical abilities of the response team	The sole mission of the incident response team is to safeguard the security of their constituents. The team must possess the technical skills to perform a lengthy investigation. The team should know when to use the expertise of the external (independent) teams to eliminate bias or to bolster gaps in a specific set of skills.
Severity of the incident	There is a better likelihood that lower-priority incidents will be investigated more often when there are resources and funding made available to a VCSIRT. On the other hand, if an incident has caused a security disaster, preventive and reactive funding support for such extreme incident cases might be readily provided.
Probability of repetition	Will the intruder possibly strike again at another time or place with a similar modus operandi? Do you have enough information to make a judgment, or should you perform more research? Response teams can mitigate the impact of incidents by passing this kind of information to peer teams and law enforcement.
Possibility of identifying a new activity	It is not easy to identify new activity, but if you determine that the attacker is introducing new methods or a new variant of an existing method, then an in-depth analysis might be justified.
Support from the constituent customers	If a customer site reports an incident, but does not agree to provide further support in data gathering or answering questions posed by the VCSIRT, then further analysis is not possible.

Forensics

Forensics is the application of scientific information to solve a legal problem. But, many define forensic computing as the gathering and analyzing of data to reconstruct data to determine what has happened in the past on a computer system, in a manner as free from distortion or bias as possible.

Very often, the forensic process is used to support a criminal or administrative investigation. Irrespective of the purpose, you should balance forensics with the priorities of the business. The forensic process can be labor intensive, technically demanding, and costly, but it might be a serious option under circumstances such as when the system-generated logs are missing as a result of the security compromise. A VCSIRT should consider these two arguments before setting out to employ forensics:

- If the availability of the customer's business is the primary concern, then it is likely that an in-depth forensic analysis and an investigation is not feasible for the customer.
- On the other hand, forensics might be an important and unavoidable step, even if it is costly, particularly for smaller companies. Setting an example of taking a perpetrator to court might prove to be beneficial in the long run. Note that there are different levels of forensic analysis, and the level is determined by how much the affected customer wants to spend on forensics. Analyses range from moderate or casual analysis of event logging data to pedantic examination of every bit and byte in the affected system and network environments.

Attention must be paid to the ethical obligation involved in selecting a forensic expert. The criteria to meet are comprehensiveness, objectivity, and precision. A VCSIRT security officer should not undertake or delegate a forensic investigation if these criteria can not be met.

- Comprehensiveness implies a thorough collection of data that supports any hypotheses.
- Objectivity implies that the process of analysis and judgment should be objective and free of bias.
- Precision, along with explicitness, leaves no chance for misinterpretation, so it must be followed.

Available Options

Primarily, there are four options when you, as a VCSIRT member, suspect a compromise: 1) ask for help, 2) reinstall affected systems and continue, 3) take no action, and 4) investigate yourself.

Although option one is perhaps the best option, and it is recommended, it might take longer than the customer's business can tolerate—not to mention the negative publicity that could result due to the delays in the actions. Option two, cleaning up and reinstalling to continue, is tempting because there will be less publicity about the compromise, but the looming question is “can it happen again?” Option three, taking no action, could create a legal liability, as in the case in which your constituent's site was used as a launching pad. Option four might help maintain the privacy of the constituent, but the question of expertise comes up. Can you or your team really carry on the investigation without involving anyone else?

Forensic Precautions

The following are some important precautions (certainly, not all-inclusive) for the servicing VCSIRTs to follow so that a detailed analysis or forensic effort, if needed due to a compromise and subsequent follow-up, could be useful:

- First, when you are given a suspect system, make sure with the help of an expert that there has been a compromise on the system.

This can be trickier than you think. To avoid the tampering of any evidence, you have to follow all of the policies and procedures set by your organization or the customer, as the case might be (refer to “Responding to a Customer’s Security Incidents—Part 3: Following Up After an Incident” for details). The system must be off line, but make sure to preserve the memory and running process tables.

For example, for the Solaris™ Operating System (Solaris OS), suspend the system by using `Stop-A` or `L1-A` without doing a graceful shutdown (that is, `shutdown(1)` or `init(0)`). If you are not running the Solaris OS, check with the vendor of the operating system for equivalent functionality. Then, follow the detailed guidelines in Part 3 of this series for mounting the file system onto another system (refer to the “Acquiring the Evidence” section), making a full backup using imaging utilities (refer to the “UNIX File System Imaging” section in Part 3), and preserving the original disk. The following steps summarize the procedure and provide a UNIX® or Solaris OS example.

1. Take the suspected Solaris OS system off the network, preferably by pulling the physical Ethernet cable (or whatever existing network medium is being used).
2. Hook up the compromised system to your forensics (clean) system by using either a minihub or a crossover Ethernet cable (or whatever physical network medium is being used in the environment).
3. Start Netcat listening on port 6000 by using the following command:

```
clean# nc -l -p 6000 > /bigspace/suspect-partition -w 30
```

Anything that comes in on port 6000 will be redirected to the output file. The `-w 30` tells Netcat to wait 30 seconds for connection.

4. On the suspect system, execute the following `nc` and `dd` commands from the forensic CD-ROM disk.

```
suspect# /cdrom/forensic/sbin/dd bs=1024 if=/dev/rdisk/c0t0d0s0 |  
/cdrom/forensic/sbin/nc a.b.c.d 6000
```

This above command sends the `root` partition, assuming that the `root` partition is on the `/dev/rdisk/c0t0d0s0` partition, to host `a.b.c.d` port 6000 where the listener was set up in Step 3.

5. Repeat Step 4 for every (system) partition on the hard drive, including `swap`.
The repetition of this step will depend on the system disk layout. You should now have an image of every (suspect system) partition copied onto the trusted, clean forensic system.
6. Produce MD5 hashes to compare against subsequent copies.

```
clean# /cdrom/forensic/bin/md5 suspect-partiton > suspect-drive.md5
```

If copies are concatenated, create an MD5 hash of the concatenated image.

7. Make a second copy.

A simple `/cdrom/forensic/bin/cp image1 image2` command will suffice. All further analysis will be done on the second copy. Put away the first image copy safely after timestamping and tagging it properly. The best way to secure it is with evidence tape in an antistatic bag.

- When, or if, your constituent customer wants to rebuild the compromised system for business continuity (that is, after creating a clean copy of the original), the customer must be advised of trusted sources. Examples are CD-ROMs or trustworthy locations that provide guarantees of downloaded contents through methods such as message digests (for example, MD5 hashes). For information on using message digests for the Solaris OS, refer to:

http://www.sun.com/blueprints/tools/fingerprint_licence.html

- In addition, the constituent should be guided by the engaged VCSIRT in all of the uses of state-of-the-art software tools and a clean backup to remove or mitigate damage and to preserve a safe copy following chain-of-custody guidelines.
- System logs are potentially the most valuable source of forensic type of information pertaining to system activities. However, several prior conditions must be met for future usefulness of the logs.
 - a. Logging must have been enabled.
 - b. Logs must be protected and intact because the general lack of authentication allows hackers to create bogus records or to delete the ones that were created by the system during the hacking activities.
 - c. When examining a UNIX system, the `/etc/syslog.conf` file must be available to see how logging is configured for each system at the customer site in question.
 - d. Logs can be found on the backup media—onsite or offsite; thus, these locations and systems must be reachable.

- e. Along the same lines, pre-examine the `cron(1M)` jobs associated with the system log files used for rotating the files and moving them out to backup media or elsewhere.

Initial Steps of a Forensic Investigation

If an incident occurs or if you have a suspicion that an incident occurred, you should first know the technical limitations of the VCSIRT. Get expert help if a forensics expert is not in the VCSIRT or geo-based security officer team. Next, immediately preserve the state. But, even without any help, make sure to freeze the scene of the crime or incident, and physically secure the scene. At all times, the VCSIRT members must remember the volatility of information and guard against damaging it, collecting data in the following order from the most to the least volatile:

- Registers, peripheral memory, and caches
- Memory (kernel and physical)
- Network state (for example, on the Solaris OS, the output of the `netstat`, `route`, and `arp`)
- Running processes (using `ps` and `proc` commands, such as `pstack`, `pfiles`, and `lssof`)
- Hardware data residue, memory chips, and PDA-type systems
- Hard disks, because they contain the largest amount of potentially useful forensic information (for example, the Solaris OS Audit Log, formerly known as the BSM log)
- Diskettes and backup media
- CD-ROMs
- Printouts

Level of Sophistication of the Attack

During the entire analysis, some of which might happen during other SIR phases before the follow-up phase, it is important to keep an eye on the level of sophistication of the hacker. This helps to understand the motives for and possible implications of the attack. For example, the expertise of the attacker might be obvious from the password-cracking utility being used (that is, if it was a network intrusion) or from a steganographical utility left behind by the perpetrator. (Note that steganography means “to hide in plain site.” Steganographic utilities generally protect data in two ways. First, they make it invisible by hiding its very existence, and second, they encrypt it.) Although the motive and implication might not be obvious, experienced forensic experts are likely to find the information useful in making key decisions about the subsequent steps in the analysis.

When a CSIRT's security expert starts examining a system, an educated guess should be made as to the level of the compromise because it helps to choose an appropriate overall investigative approach.

Look at TABLE 3 for the levels of compromises on a UNIX host that determine corresponding levels of sophistication of investigative procedures. Hacked binaries on UNIX are common, but do not assume that only system binaries are affected. You might consider a hacked kernel as a rare occurrence, but, remember when it does happen, it is hard to detect.

TABLE 3 UNIX Host Compromise and Corresponding Implications

Level of Compromise Due to a Host Intrusion	Implication	Possible Investigative Approaches
Access to user account	Not difficult to continue and get root access.	Simple to detect in logs (either missing or visibly incomplete) and access times
Application	Might create fake data, or data might be missing.	Hash values of all application binaries compared against hash values of known good binaries can reveal proof.
System application	Allows hiding of tracks.	Check hash values of all system binaries against known good binaries (for example, from <code>/sbin</code>).
Shared libraries	If not statically linked, the dynamically loadable binaries are exposed to hacking. Note that, in general, shared libraries can be hijacked or superseded by <code>LD_PRELOAD</code> or <code>crle</code> (configure runtime linking environment).	Examine the hash values of the shared libraries against hash values of good shared libraries (for example, <code>lib/*.so</code>).
Kernel	Entire kernel and the operating system become suspect.	The investigation takes great expertise, but booting from a diskette or placing the data on another system isolates the kernel or operating system from the effect of the compromise.
Hardware	Entire host system and its environment are suspect.	Most likely, this will require a laboratory analysis by an expert.

MAC Times

The MAC times (that is, the `inode`-related modify, access, and create times) can help correlate system events with file system disturbance times and are useful in any investigation. They are particularly useful when examined in relation to audit logs, system logs, and file systems.

- As long as you analyze a read-only copy of a suspect file system, you can collect the MAC times whenever it is required or convenient. But, it is advisable to capture the inode times as early in the investigation as possible so that there is no intentional or inadvertent modification of the times associated with the evidential data.
- On a running system, inode times are highly volatile, so you must collect them before somebody or some process accidentally changes them. For example, on UNIX, you can do this with the following command, but you might get awkward output from the suspected hacked system:

```
$ find /mnt/hacked.root.image -type f -print0 |  
xargs -0 stat > hacked.root.statout
```

Note – Publicly available tools (see “Forensic Tools” on page 28) include the convenient `mactime` utility that outputs a single line for each file system object and sorts all of the entries by inode change time.

- Use an editor (for example, `vi(1)`) to examine the output. Look for files that have been introduced into the system recently.
- For deleted files, some of the file attributes are erased, but the ownership and MAC times are preserved in many UNIX systems. MAC times of inodes that were attached to files might be recovered by using the `ils` tool in The Coroner’s Toolkit.

Note a few shortcomings and cautionary points:

- The most obvious shortcoming is that MAC times represent only the last time a file was disturbed. There is no historic data on earlier activity. MAC times are not a silver-bullet solution.
- Do not use MAC times as a replacement or alternative for digital signatures or other forms of file validation. For example, the output of the `ls(1)` command might show that the binary has not been used in a while; however, an attacker can put a trojan horse on it, then simply modify the access times.
- On UNIX systems, the `touch(1)` command can be used to change the access time (`atime`) and the modification time (`mtime`). On UNIX and NTFS file systems, the `utime()` system call can also be used to change those times. The following is a Perl code example:

```
$set_time = time(); # get current time  
utime($set_time, $set_time, $file); # set file's atime, mtime to current time
```

- MAC times are less useful on busy, multiuser systems. Degradation is unavoidable because the hacker activity rolls into the past.

Logs

Logs are usually very helpful when used in conjunction with a report on MAC times of a suspect system. In UNIX systems, generally, a major weakness of `syslog(3)` is its lack of authentication services. Intruders exploit this to their benefit by cleaning or corrupting log files. Log files can provide information such as:

- Who logged in (when and from where)
- What kind of login occurred (for example, Telnet, rlogin, and X)
- What destinations were sent email
- What errors occurred and system events happened (for example, server devices going down and coming up)

A few common suspicious logging circumstances and the corresponding tips to aid in forensics are described in TABLE 4. (This list is not all-inclusive.)

TABLE 4 System Logs and Forensic Tips

Characteristics of Logs	Forensic Tip
Log entries missing	It is hard to determine which entries are missing, but other procedures, along with log file analysis such as MAC times analysis, might help.
Log has unusual activities at odd hours	What is unusual in the log is perhaps deterministic if system administrators are involved, rather than just CSIRT and their forensic expert. National holidays and weekends are typical days for attacks.
Log shows login entries from unusual sources	Customer site network policies should be examined to determine what is permissible. Possible logins from outside the country should be suspected.
Log has failed login records	Normal users mistype and re-attempt to log in; however, repeated logins at odd hours should indicate an unauthorized individual using an authorized user name and password.
Log shows time periods without any activity	Deletion by intruder is likely, but usually, there is a lot of subtlety. Other methods, such as MAC times analysis, could help validate the deletion. In addition, look for the starting time for the log. Is it correct? You will need information on any <code>cron(1M)</code> jobs and other administration procedures to corroborate.
Log has records with attempts to access the <code>/etc/passwd</code> file	Remote attackers often try to obtain the hashed passwords for cracking them. Use this train of thought, and trace unauthorized access. UNIX commands such as <code>ps(1)</code> , <code>w(1)</code> , and <code>who(1)</code> all report the terminals to which each user (for each process) is attached. Note that the intruder could call your computer by telephone. You need prior arrangement with the telephone company to have caller ID-type features.

TABLE 4 System Logs and Forensic Tips (Continued)

Characteristics of Logs	Forensic Tip
Log has unusual or failed superuser logins	See implications for other users. Were they being prevented from using the system? Was the legitimate system administrator able to log in? These are indications that hackers had superuser access.
Missing log on the system	Deliberate removal is a possible occurrence if the <code>syslog.conf</code> file rules indicate the creation of a log.
Log shows records of network or name service errors	Examine the times, and see if the network or name service errors happened before a known compromise. If yes, then DNS or some other network or name service might have been attacked.

Audit logs are helpful—the primary intent being to record user actions to detect malicious intent. For example, the Solaris OS provides the capability to log the activity on a system at a granular level. This is part of the SunSHIELD™ Basic Security Module (BSM). For more information, refer to the *SunSHIELD Basic Security Module Guide* from the <http://docs.sun.com> site.

A malicious user might not take certain actions, knowing that those actions will be recorded by an auditing mechanism. On the other hand, if the hacker is not smart enough, you might get some useful records. The best practice and challenge is in deciding the following:

1. Which events generate meaningful information in the environment of the CSIRT's constituent.
2. What can be derived from an audit log after a potential security compromise on the system.

There are login and logout events and administrative events such as for a reboot, a mount, and so on. Additionally, the Customer Audit Events class (which is not part of the default) captures file ownership changes, changes to the `root` directories (`chroot`), changes to process priorities, and changes to a process UID. These are very useful in the intrusion and unauthorized access detection.

File System and File Content

You might realize from the analysis of the MAC times and the logs that certain files within a file system or partition appear to be suspect. For example, they might be suspect due to a set of missing log records in which boundary times either exactly or approximately correspond to certain file access or modification times from the MAC times output. After you note what partitions the drive has, examine the directory listing (including subdirectories) to get a sense of what needs to be analyzed, and print out the listing using shell commands, such as `ls(1)` from a known good CD-ROM disk.



Caution – Be aware that the CD-ROM discs cannot be assumed to be clean if patches have been downloaded to them from some arbitrary Internet site and not directly from the vendor. There is a chance of corruption.

In a UNIX or Linux environment, if a data collection device is mounted as `/mnt/myexport/` (read-only), you can redirect the standard output to the external device with one of the following commands:

```
$> ls -al > /mnt/myexport/targetdirlist  
  
or  
  
$> find . -ls >/mnt/myexport/targetdirlist
```

Typically, a hex editor or some forensic program is used to view the master boot record and the boot sector on the victimized system.

Note – The Coroner’s Toolkit (TCT), Sleuth Kit, and Autopsy are examples of user-friendly tools that have useful capabilities for file system and file content analysis. Autopsy is browser based and used as the front end to the Sleuth Kit. See “Forensic Tools” on page 28 for more information.

Deleted Files and Residual Data

With patience, deleted files can be retrieved manually using a hex editor or using forensic utilities, such as those in TCT or Autopsy. Note that manual retrieval of fragmented files is a complex process. It needs prior training; however, it can be worth the effort if the law allows you to retrieve detailed information in the country where the compromise has taken place.

In the U.S., courts have held that the deleted files on a hard drive are *discoverable*, and an expert must be allowed to retrieve all recoverable files (see *Easley, McCaleb & Associates, Inc. v. Perry*, No. E-2663, GA Superior Court, July 13, 1994). However, the court might say such access is not unlimited. In two recent decisions, access to a litigant’s computer was denied because the party seeking discovery could not show a likelihood that relevant information could be retrieved (see *Strausser v. Yalamachi*, 669 So. 2D 1142, 1144-45, FA Appellate Court, 1996, and *Fennell v. First Step Design, Ltd.*, 83 F.3d 526, 1st Circuit Court, 1996).

Unallocated Space

After the deleted files have been recovered, the next step is to check for residual data in unallocated and slack space, usually accomplished with specialized software tools.

- When you start the examination of a hard drive, you need to find out how many partitions it has and how big they are. Then, make sure that the total space in all of the partitions adds up to the size of the drive (with some allowance for any adjustments done by the storage manufacturer).
- There is usually quite a bit of unallocated space, most of which contains some kind of data. Almost always, it is not deliberately hidden data. It is the data the operating system left orphaned, and it can be overwritten some time in the future. Slack space, on the other hand, is the space that is left over between the end of data and the last block. Every file that is not an even multiple of the block size has some slack space associated with it. The most convenient way to analyze the data in the unallocated space is to use forensic tools, such as TCT or Autopsy. On a UNIX system, you can view an entire partition as a single object, so you can use a hex editor to search through a partition or a hard drive. However, this kind of search is not convenient.
- The investigating or responsible CSIRT team must ensure that if there is sensitive data in these spaces belonging to the customer, the data does not leak outside the team when the hard drives under examination are reused or disposed.

Hidden Data

In addition to unallocated space as a possible place for investigation on a compromised system, the actual existence of the data could be hidden by the attacking intruder, as in the following:

- There could be plain-looking data that has alternate meaning, or data could be concealed within files that appear to be normal. The data could have a detrimental impact. It could contain some special characters (for example, watch for and track down control characters in the `/etc/inetd.conf` file on UNIX systems to replace carriage returns).
- There could be files with invisible names that cannot be seen by operating system utilities. They could include files with misleading names and inappropriate suffixes and files that are intentionally misplaced and kept in obscure directories. These need to be tracked by smart keyword searches with specific characteristics.
- There could be zero-link files (as in UNIX operating system environments) that have no directory entries and must be tracked before shutting the system down so that they are not lost.

- Consider hidden data in the EEPROM and data hidden using steganography. For the former, you must rely on the utilities provided by the vendor of the OS. For the latter, there are tools such as S-tools to help reveal the hidden data. For more information, go to:

<http://members.tripod.com/steganography/stego/software.com>

- There could be data stored on removable media, handheld devices, and other remote devices, or moved away from the target system onto other LANs, WLANs, or onto the Internet. In general, for a detailed analysis, you must examine all of the possible media and search for evidence of remote connectivity.

Unknown Code

While examining the file system, you might discover unknown code. When analyzing this unknown code, which could be malware (dropped by attackers on the compromised systems), there are several things to check to get useful information:

- If it is a script, you might be able to read it. The installation time can tell you something about the intruder's activity during the compromise.
- The name of the code is handy in using `grep(1)` on UNIX or Linux to find other locations of this code on the network and to learn about it on the Internet.
- The `strings(1)` command can search binaries for a specific string you know a hacker might be using. Pay attention to strings that are not casual, and remember that intruders sometimes intentionally spell common words differently or incorrectly. Look for source code on the system if you found a binary because hackers leave traces.
- The `nm(1)` utility, from Berkeley UNIX and in many versions of UNIX, is useful in identifying known symbols in binaries of tools or libraries. After you find some unique strings, search for them by using Google, and you might be surprised to find similar issues elsewhere, often on security mailing lists.
- The `truss(1)`, `sotruss(1)`, and `whocalls(1)` utilities can provide information about the opening and closing of devices and files, and other system-call level inner workings of the unknown programs. The `strace(1)` command can print system-level (streams on the Solaris OS) trace messages for an executable. These must be run on a machine that is disconnected from the network and in a contained lab environment. After such experiments, the machine must be reinstalled with an operating system before further use.
- Whenever necessary, VCSIRTs must take help at least from programmers if a computer forensic expert is not on hand. But, note that many of the previously mentioned commands can be effectively used by advanced UNIX users who are not necessarily familiar with forensics.

Other Items of Interest

There are many other items of interest in forensic analysis. The following lists a few of the important items, along with helpful UNIX commands (refer to the specific man page for details):

- Video display (for example, `xwd` and `xwud`)
- Running processes (for example, `uptime`, `who`, `ps`, `top`, `lsof`, and `fuser`)
- Network connections (for example, `netstat`, which should be used to ease the data collection as described earlier, and `arp`)
- Hostile executables (for example, `strings`, `grep`, and `nm`)
- Login logs, that is, logs that contain records of every console login, Telnet session, X-session, `rsh` use, and FTP session (for example, `last`, `lastlog`, and `who`)
- Invisible files and directories (for example, `find` and `nccheck` for locating SUID and device special files on System V-based UNIX versions)
- Core dumps (for example, `find`, `file`, `adb`—the UNIX debugger, and `gdb`—the GNU debugger)
- Firmware breaches

An attacker with physical access to a workstation without EEPROM security enabled (for example, with the superuser password before entering the privileged shell) can easily compromise the system by booting it in single-user mode.



Caution – The output of the commands listed above cannot be trusted under all circumstances. Consider that the programs, libraries, or the kernel could have been compromised on the system where the commands are executed. On a compromised system, the commands you use to list items often get replaced by a rootkit. Therefore, you, as a response team member or representative working at the constituent's compromised site, should always run software (or commands) that you need and often use from a secure CD-ROM disk, prepared specifically for recovery or investigation.

Forensic Tools

Computer forensic analysis is a technically intense task. A cursory examination must first be conducted by the CSIRT-deployed forensic expert. Then, if the preliminary findings warrant a more detailed investigation, you can select from many excellent tools that are freely or commercially available. For example, if an intruder logs in, installs a network sniffer or a backdoor program, and deletes the system event logs, there will be a considerable amount of crime data left behind for a forensic expert to uncover. This is where tools can help to sift through the data to help you make educated and informed deductions as to what the perpetrator might have done on the system.

Note that the following also pose problems for data recovery in a UNIX system:

- You do not have superuser privileges on the machine.

This could be because you are not considered savvy enough on UNIX and your organization does not trust you with superuser privileges. It could also be because you are not in a role to use such privileges. In either case, you will need to bring an appropriate person with authority.

- You are not experienced with UNIX.

It is better to admit it up front and get immediate help.

- You have very little time or patience.

This could prevent any kind of thorough and methodical analysis.

- Your management does not support these activities.

This is often a business decision, particularly when you are dealing with a compromised production system. Bringing up the system online is more important than data recovery.

- You are trying to recover large files, greater than 15 to 20 kilobytes.

- You are analyzing data with a very large storage subsystem (greater than 100 gigabytes).

This could also pose problems for copying with the intention of preserving the copied data. The customer site might not have a duplicate storage array for analysis.

Utility Media

Your incident response team should have an ISO 9660 CD-ROM containing binaries you will need for conducting investigations. At a minimum, you should have `dd(1M)`, `cp(1)`, `cat(1M)`, `ls(1)`, `ps(1)`, `lsof`, `strings(1)`, `find(1)`, `grep(1)`, `less(1)`, `vi(1)`, `perl(1)`, `TCT`, `ifconfig(1M)`, `kill(1)`, `Netcat`, and `tcpdump`.

Finding all of the pieces needed to create statically linked binaries can be hard and cumbersome. There are some web sites with ready-made statically linked binaries for use in incident response.

Tools From Sun

Sun Microsystems has MD5 signature-based utilities for free download. For more information about these tools go to:

http://www.sun.com/blueprints/tools/fingerprint_license.html

In addition, Sun Microsystems offers the Solaris Fingerprint Database Companion (`sfpC`) and the Solaris Fingerprint Database Sidekick (`sfpS`). The Solaris Fingerprint Database Companion automates the process of collecting and checking MD5 cryptographic checksums against the Solaris Fingerprint Database (which has the checksums of the known good system binaries). The Solaris Fingerprint Database Sidekick simplifies the process of checking for the presence of rootkits on the compromised system.

`sidekick.sh` is a simple tool to help you in the collection of MD5 checksums. A typical check might include running the following command:

```
# sidekick.sh -R /mnt/suspectdisk -a -S
```

This command collects an MD5 checksum of all the system files started at that mount point. The `sidekick.sh -r` command collects the MD5 checksums of a subset of binaries that are commonly found in a rootkit.

After you have MD5 checksums of all of the files, you can compare this list to a list of MD5 checksums from a known uncompromised system. Whatever does not match is a patched binary that was not in the original vendor release, is not a Sun binary, or is a trojan horse.

Tools Available on the Internet

The following list contains a few locations for publicly available tools that are helpful in incident response and follow-up. Be sure to use a well-known mirror of these sites:

- Sun BluePrints™ Scripts and Tools web site
(<http://www.sun.com/blueprints/tools/>)
- CERT/CC (<ftp://info.cert.org/pub/tools/>)
- DFN-CERT (<ftp://ftp.cert.dfn.de/pub/tools/>)
- Purdue University's Computer Operations, Audit, and Security Tools
(<ftp://coast.cs.purdue.edu/pub/tools/>)

Netcat (`nc`) is an extremely useful utility for transferring information over TCP or UDP when creating a copy of the information from the compromised system. Be certain that your incident response team has the authority to install and use Netcat on the customer's network before doing so. For more information about Netcat, refer to the following sites:

- http://www.atstake.com/research/tools/network_utilities/
(primary site)
- <http://www.sans.org/rr/audit/netcat.php>
- <http://www.insecure.org/tools.html>

The Coroner's Toolkit

The Coroner's Toolkit (TCT) is a set of software tools that help in gathering and reconstructing data, as well as in the analysis of the data. For example, if your system has been broken into and your file system has been damaged by a vindictive system cracker, you can use TCT to recover the files. On a UNIX system, however, there is no way to recover lost data that has not been backed up.

To use TCT, you will need specific knowledge about your customer's compromised system and spare disk space to recover deleted files. The following list contains some tips for using TCT:

- You will need at least 220 percent of the file size of disk space on the affected disk because the unallocated blocks that you will be recovering need 100 percent and the Lazarus tool will need at least 120 percent.
- You will need at least a few hours of time because the programs can be slow.
- You should use `unrm` to recover the data from the affected file system to another file system.
- You should run Lazarus and search the results until you find the data.

To preserve the ephemeral data, you can use Grave-Robber to capture data from a system, usually from the most volatile data to the most stable data (for example, program memory, network connections, and MAC times). The `mactime` program can selectively view a slice of historical time (for instance, from 8:35 a.m. to 9:45 a.m.) from the database that Grave-Robber creates. Take, for example, a remote login session in UNIX. The `inetd` program listens to the Telnet port and forks off the Telnet daemon. The Telnet daemon then executes the `login` program. The `login` program authenticates the user and becomes the shell after updating the login accounting files. `mactime` displays all of the executed programs (`inetd`, `in.telnetd`, `login`, and `csh`), the configuration and authentication files used (`passwd`, `group`, `ttytab`, and `motd`), and the system accounting files (`utmp`, `wtmp`, and `lastlog`).

For more details about TCT, refer to the <http://www.fish.com/tct/> site.

Sleuth Kit and Autopsy

The Sleuth Kit (previously known as TASK) is a collection of UNIX-based command-line file system and media management forensic analysis tools. The file system tools enable you to examine NTFS, FAT, FFS, EXT2FS, and EXT3FS file systems of a suspect computer in a non-intrusive fashion. The tools have a layer-based design and can extract data from the internal file system structures. Because the tools do not rely on the operating system to process the file systems, deleted and hidden content is shown.

The media management tools enable you to examine the layout of disks and other media. The Sleuth Kit supports DOS partitions, BSD partitions (disk labels), Mac partitions, and Sun slices (volume table of contents). With these tools, you can identify where partitions are located and extract them so that they can be analyzed with file system analysis tools.

When performing a complete analysis of a system, command-line tools can be tedious. The Autopsy Forensic Browser is a graphical interface to the tools in The Sleuth Kit, which enables you to more easily conduct an investigation. Autopsy provides case management, image integrity, keyword searching, and other automated operations.

The Sleuth Kit and Autopsy are both open source and free to download. Their combined features include:

- Viewing allocated and deleted files and directories, access to low-level file system structures, time lines of file activity
- Sorting and checking file categories and extensions
- Conducting keyword searches, including `grep(1)` regular expressions
- Identifying graphic images and creating thumbnails
- Looking up hash databases, including the NIST NSRL and hash keeper, investigator notes, and report generation

For more information about Autopsy and Sleuth Kit, go to the following sites:

- <http://www.sleuthkit.org/autopsy/>
- <http://www.sleuthkit.org/sleuthkit/>

Memory Imaging and Forensic Tools for Palm OS Devices

Tools, such as `pdd` from @stake, that aid in data acquisition and analysis of portable devices should be readily available in the incident response toolkit of a CSIRT. PDAs are ubiquitous in the consumer marketplace, and they will become targets for criminal investigations and forensic analysis. The `pdd` tool retrieves and displays device information and databases. It provides acquisition capabilities for use by

forensic investigators, incident response teams, and criminal and civil prosecutors. The pdd tool is a Win32 executable. The source code is available for free at the <http://www.atstake.com/research/tools/index.html#pdd> site.

Some very handy forensics features are recovery of records marked for delete but not removed, retrieval of Palm OS system passwords, and information retrieval from telephony applications (speed dial database, call history database, web browser cache, scripts, and passwords for network connections). Flash ROM is used to back up applications and databases, so these can be imaged and analyzed during forensic acquisition.

ForensiX on Linux

Linux can support many more file systems than most operating systems, and ForensiX takes full advantage of this flexibility by offering the automated ability to mount images or media in dozens of different file systems. File system mounts are read-only. Many types of supported media can be quickly imaged, checked with MD5, and logged into a case database. The tool has several unusual capabilities such as a function to check a UNIX system for known vulnerabilities. It also has a means to build a baseline picture of a file system, store the hash values and filenames, and then compare the baseline to other file system images. It also has the ability to create and analyze TCP dumps. For more information, refer to the <http://www.all.net/> site.

Commercial Forensic Tools for Windows

The following table contains some of the types of forensic-related tools available in the industry. For example, one vendor for DOS and Windows 95, 98, 2000, and XP forensics tools is New Technologies, Inc. Their tools are described at the <http://www.forensics-intl.com/tools.html> site.

For convenience, the tools are categorized according to their usage for various forensic analysis tasks.

TABLE 5 Computer Forensic-Related Tool Types

Type of Tool	Descriptions
Media copy	<ul style="list-style-type: none"> • Diskette duplication tools can be used to create duplicates of frequently used diskettes in incident responses. • There are programs that can be used to create an evidence-grade bit stream backup of a computer hard disk drive, zip disk, or flash memory card.
Content validation	<ul style="list-style-type: none"> • Cyclic redundancy check (CRC) file hashing programs validate the contents of one or more files. The CRC is a very powerful technique to obtain data reliability. • A CRC hashing tool can be used to validate mirror image backup accuracy.
Data scrubbing	<ul style="list-style-type: none"> • U.S. Department of Defense (DoD) grade disk drive scrubbing utilities can be used to securely eliminate all of the data on a disk. • A forensic filter can be used to eliminate binary data and to control characters from ambient data sources. • An ambient data security scrubbing utility can also be used, as long as it meets the DoD standards.
Accounting	<ul style="list-style-type: none"> • Hard disk and diskette cataloging tools can be used to evaluate computer usage time lines. • Forensic software tools can be used to quickly identify Internet account uses and abuses. • There are tools that can be used to document the CMOS system time and date on a computer that has been seized as evidence. (Note that in addition to CMOS time, you will need the use of NTP and collection of time deltas from other affected components such as terminal servers, routers, and switches to prove a network event.)
Language usage identification	<ul style="list-style-type: none"> • Intelligent fuzzy logic filters can be used with Windows swap and page files and other ambient data sources to identify English-language communications. • Intelligence-gathering tools can be used to quickly identify the names of individuals on a target computer. These tools are <i>name aware</i> concerning English, European, and Arabic name formats. These tools are only available for military and intelligence agency uses.

TABLE 5 Computer Forensic-Related Tool Types (*Continued*)

Type of Tool	Descriptions
Text or data pattern search	<ul style="list-style-type: none">• Intelligent fuzzy logic filters can be used to identify data patterns associated with credit card numbers, social security numbers, phone numbers, and bank account numbers.• Forensic hex search utilities can be used to find binary data patterns associated with file headers and foreign language data patterns. Intelligent fuzzy logic filters are useful in quickly identifying HTML patterns in ambient data sources.• DoD-grade text search utilities can be used to locate targeted strings of text.
File manipulation	<ul style="list-style-type: none">• Forensic utilities can be used to capture static swap and page files for analysis with other forensic tools and processes.• Ambient data collection tools can be used to capture unallocated (that is, erased) file data.• Ambient data collection tools can also be used to capture file slack for analysis and to quickly and automatically reconstruct BMP, GIF, and JPG files in cases involving the inappropriate or illegal downloading of images or viewing of pornography on the Internet.
Disk drive utilities	<ul style="list-style-type: none">• Disk drive utilities can be used to analyze and document disk drive operating systems and allocated partitions.
Computer access	<ul style="list-style-type: none">• Computer access programs can be used to lock and secure evidence on computers.

These tools can help corporate and government computer forensic experts and other investigation specialists dealing with realized and potential computer risks and compromises associated with accidents, malicious acts, criminal acts, and corporate policy abuses.

Forensic Toolkit for NT

This is a set of command-line utilities that are helpful in reconstructing access in a Windows NT file system. It is available from Foundstone and described further at the <http://www.foundstone.com/rdlabs/tools.php> site.

The various utilities have the following functionalities:

- Provide a list of fields by their last access times, without changing the directory access times
- Scan a disk for files that either have the hidden attribute set or use the Windows NT directory and system attributes to hide files
- Scan an entire disk for hidden data streams
- Report all of the attributes of a single file

Encryption Issues in Analysis

Electronic cryptography is a collection of practical and easily affordable techniques for encoding electronic data so that it can be understood only by the intended users or recipients. Therefore, the encryption of data has been one of the ways to protect privacy in any kind of data storage, retrieval system, or communication system. However, such type of data privacy is not on everyone's mind. Data encryption, with the intentional misuse by computer equipment and network attackers, poses problems in subsequent analysis of security incident data. In general, encryption can give criminals and terrorists a powerful tool for concealing their activities from being analyzed for the following reasons:

- Encryption makes obtaining the evidence needed for a conviction or the intelligence for criminal investigations difficult for law enforcement agencies.
- Encryption can frustrate communication intercepts used for analysis, which have played a significant role in averting terrorist attacks and in gathering information about specific transnational threats, including terrorism, drug trafficking, and organized crime.
- Encryption can delay investigations and add to their cost.

Password protection, and especially file encryption, used by hostile intruders can cause the following problems for law enforcement agencies, military agencies, corporations, and government agencies:

- Forensic text search utilities cannot identify strings of text in encrypted files.
- Critical business data can be held hostage through the use of file encryption.
- Evidence in criminal cases can be secured by criminals through file encryption.
- Passwords used to secure data in files can be forgotten or lost when an employee becomes unavailable.

Note – The L0phtCrack password-cracking tool by @stake uses cryptanalytic techniques to effectively reduce the potential key space or total population of usable keys. Then, it offers you a choice of a dictionary attack (that is, an attack based on permutations of dictionary words) or a brute force attack. For more information, refer to the <http://www.atstake.com> site

The use of encryption to hide criminal activity and to prevent analysis is not new. The April 1970 issue of the *FBI Law Enforcement Bulletin* contains several cases in which law enforcement agencies had to break codes to obtain evidence or prevent violations of the law. None of the cases, however, involved electronic information or computers. Relatively simple substitution ciphers were used to conceal speech.

Nevertheless, the majority of computer crime investigations have not been stopped by encryption, particularly those related to computer searches and seizures. A few options are available as proven best practices, as follows:

- Authorities can obtain the key by consent from the subject or suspect.
- Authorities can find it on a disk drive or crack the encryption in some way.
- Authorities can use other evidence, such as printed copies of encrypted documents, unencrypted conversations and files, witnesses, and information acquired through other, more intrusive, surveillance technologies such as bugs.

In addition to encryption, you need to be aware of other common forms of data manipulation that can complicate data analysis. For example, compression is a way of reducing the size of a data object. Compression is normally a reversible process. There are various kinds of compression techniques. So, the one deployed on the data at hand must be identified to uncompress the data so that text string searches and other data investigative tools can work.

Analysis in Wireless Networks

It is estimated by Gartner that at least 20 percent of organizations already have rogue WLANs attached to their corporate networks from authorized network users. There are tools for analyzing data on a wireless network, but more progress is needed to keep up with the members of the hacker community. TABLE 6 contains a few examples of tools available on the Internet and what they do on the WLAN.

Note – IEEE Std 802.11-1997 specifies a single Medium Access Control (MAC) sub-layer and 3 Physical Layer Specifications. The standard provides two Physical Layer specifications for radios that are operating in the 2400 to 2483.5 MHz band, depending on local regulations, and one for infrared. Refer to the <http://grouper.ieee.org/groups/802/11/main.html> site for details of the standard.

TABLE 6 Wireless Network Analysis Tools

Tool	Location	Function
Kismet	http://www.kismetwireless.net/	Kismet is an 802.11 wireless sniffer that is capable of reporting raw packets.
WEPCrack	http://wepcrack.sourceforge.net/	WEPCrack is an open source tool for breaking 802.11 WEP secret keys.

TABLE 6 Wireless Network Analysis Tools (*Continued*)

Tool	Location	Function
wellenreiter	http://www.remote-exploit.org/	wellenreiter is a wireless network discovery and auditing tool.
AirJack	http://802.11ninja.net/airjack/	AirJack is the current development version has full station and ad hoc modes of operations, while supporting raw (802.11 headers and all) traffic injection and reception.
libwlan	http://libwlan.tuxfamily.org/	libwlan is an 802.11 frame injection library.

Effective encryption and authentication security measures for wireless LANs (WLANs) are not keeping pace with tools on the net, and hackers possess easy-to-use tools that can launch increasingly sophisticated attacks on WLANs.

For example, on May 27, 2003, there was an intrusion of the Cryptome (<http://cryptome.org/>) administrator's home network from an address in the Washington D.C. area or, at least, spoofed from that area. The address was shown to belong to the ATT Wireless services, so it could have come from anywhere, including by a variation of the newly introduced wireless spam attack.

There are two points to take from this when you undertake an analysis of WLAN security incident data:

- Wireless attacks are on the increase due to the ease of originating hard-to-trace attacks through wireless telephone gateways and widening access to unrestricted Wi-Fi services.
- There is a likelihood that WLANs will be used more by criminal attackers (and snoopers from intelligence agencies) wishing to cloak their origins.

For detecting and analyzing data gathered from enterprise-level WLAN rogue accesses and intrusions, there is some help. Some commercial tools, such as those from AirDefense (<http://www.airdefense.net>), are available that identify rogue access points that broadcast a connection to the enterprise network, create accidental associations with neighboring WLANs, or form ad hoc peer-to-peer networks between devices. The 802.11 WLAN IDS abilities are also available to perform, for example, real-time network audits and analysis.

Article Series

The "Responding to a Customer's Security Incidents" articles are an ongoing series. The next article will cover best practices for managing vulnerabilities. This topic was presented briefly in the third article.

References and Links

The following is a list of references and helpful links:

- Autopsy at: <http://www.sleuthkit.org/autopsy/>
- Denning, D. and W. Baugh. "Hiding Crimes in Cyberspace," at: <http://cryptome.org/hidingdb.htm>
- Electronic Crimes Task Force and Helpful Links, at: <http://www.ectaskforce.org/> and http://www.ectaskforce.org/Helpful_Links.htm
- Farmer, D. and W. Venema. "Computer Forensic Analysis Class Handouts," at: <http://www.fish.com/forensics>
- Farmer, D. W. Venema. "What are MAC times?" Dr. Dobbs Journal, October 2000.
- Federal Bureau of Investigation. *FBI Law Enforcement Bulletin*, April 1970.
- Grand, J. "pdd: memory Imaging and Forensic Analysis of Pal OS devices," *Proceedings of the 14th Annual Computer Security Incident Handling Conference*, June 2002.
- Internet Engineering Task Force, Request for Comments (RFC 1321, RFC 1918, and RFC 2644), at: <http://ftp.rfc-editor.org/in-notes/>
- Masurkar, Vijay. "Responding to a Customer's Security Incidents—Part 1: Establishing Teams and a Policy." Sun BluePrints OnLine, March 2003, at: <http://www.sun.com/solutions/blueprints/>
- Masurkar, Vijay. "Responding to a Customer's Security Incidents—Part 2: Executing a Policy." Sun BluePrints OnLine, April 2003, at: <http://www.sun.com/solutions/blueprints/>
- Masurkar, Vijay. "Responding to a Customer's Security Incidents—Part 3: Following Up After an Incident." Sun BluePrints OnLine, September 2003, at: <http://www.sun.com/solutions/blueprints/>
- Noordergraaf, Alex. *Enterprise Security: Solaris Operating Environment*. Prentice Hall, 2002.
- Osser, W. and A. Noordergraaf. "Auditing in the Solaris 8 Operating Environment," Sun BluePrints OnLine, February 2001, at: <http://www.sun.com/solutions/blueprints/>
- Powell, B., "Forensics Lite," *login*, November 2001.
- Rude, T. "DD and Computer Forensics: Examples of Using DD within UNIX to Create Physical Backups," at: <http://www.crazytrain.com/dd.html>
- SANS Denial of Service website at: <http://www.sans.org/dosstep/index.php>
- Solaris OS Fingerprint Database at: http://www.sun.com/blueprints/tools/fingerprint_licence.html

- Sun BluePrints Scripts and Tools web site at:
<http://www.sun.com/blueprints/tools/>
- Sun Microsystems, Inc. Solaris Security Toolkit, at:
<http://www.sun.com/software/security/jass>
- The Sleuth Kit, at: <http://www.sleuthkit.org/sleuthkit/>
- The Coroner's Toolkit, at: <http://www.fish.com/tct/>
- U.S. Government Secure Hash Signature Standard at:
<http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>

Third-Party URLs

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Acknowledgments

The author would like to recognize the following individuals for their contributions: senior personnel from Sun Services, Engineering, and Sun Corporate, and IT Security for reviewing this article and providing helpful comments (in particular, Joel Weise, Martin England, Glenn Brunette, Matthias Kussinger, Larry Dunn, Mark Lawler, Steve Gillis, and Brad Powell). Sun BluePrints program contributors in editing and revising are very much appreciated (in particular, Dan Barnett and Billie Markim).

About the Author

Vijay Masurkar is a Principal Engineer and a leading services architect and technologist for Network and Security at Sun Microsystems. Currently in Sun Support Services, his research interests are best practices for enterprise and Internet level reliable and secure network architectures and customer security services. Vijay has been in the computer network and security industry for twenty-eight years. He has led large-scale research and development projects, consulting and support for VAX/VMS, Wang VS, Solaris, and TCP/IP-based network and security products and services. He publishes frequently, represents Sun in several industry forums, and has spoken at numerous international conferences. He is often invited to teach at Sun and in the industry. Vijay holds a B.S. in Electrical Engineering, an M.S. in Computer Systems Engineering, and an M.B.A., majoring in operations research.

Ordering Sun Documents

The SunDocsSM program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine web site at: `http://www.sun.com/blueprints/online.html`

