;login:
THE MAGAZINE OF USENIX & SAGE
November 2001 • Volume 26 • Number 7

Special Focus Issue: Security
Guest Editor: Rik Farrow

inside:

BEST PRACTICES
A Crash Course in Managing Security
by David Brumley

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

a crash course in managing security

by David Brumley

David Brumley is well known for his site *http://www.theorygroup.com* and for his role as Assistant Computer Security Officer for Stanford University.

dbrumley@stanford.edu

Introduction

This article is about managing infrastructure from a computer security perspective. In this article there will be three recurring themes. The first theme is the need to do things the proper way the first time. A correctly written program will always act predictably, even on bad or malicious input, and hence be secure. Similalrly, when designing an infrastructure each service is reduced to its essential components. Each component is combined to act predictably.

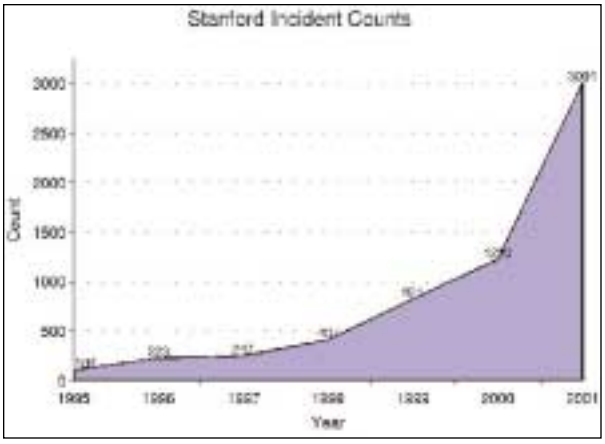
The second theme is the need for proper planning. A clear and consistent plan keeps all the different parts of the computer infrastructure functioning together. Without a plan, security is a catch-up game, constantly behind the latest exploits and news. With a clear plan, security provides *defense in depth*. As new components are added to the infrastructure, defense in depth will be enhanced, not compromised. By considering security implications at the appropriate levels of infrastructure planning, a system can become not only more secure but also easier to use and more robust, and can provide better availability.

The third theme is that scalable infrastructures can reduce overall cost while enhancing security. A scalable infrastructure is one that not only can be extended, but also can decrease the cost of extension. Scalable infrastructures are modular, allowing new technology to update and extend the old.

The Need

Do your users have any expectation of privacy? Do you have assets that need protecting? Have you considered the cost of a system compromise? These are dumb questions. Yet, we still fool ourselves into thinking we aren't the targets.

A picture is worth a thousand words.



This graph depicts the incident counts from Stanford University. Included in the incident counts are attempted intrusions from one of our network links to the Internet.

Notice the figure depicts an exponential increase in incidents. Each year, except one, the incident count doubled. These figures demonstrate that computer security incidents are on the rise. Without automated, scalable mechanisms to resist attacks, systems will be compromised quickly. For example, the recent "Code Red" Internet worm was able to compromise some servers at Stanford within only hours of installation.

Now think back to the questions I just asked about computer security. These incident counts are alarming because users expect some level of privacy and protection on the Internet, which you must provide.

What Is Computer Security

The term computer security often conjures up a mental image of a teenage hacker breaking into the US Department of Defense computers. In a dimly lit room investigators watch a screen "trace back" the "hacker," with a SWAT team on hand for good measure. While this sounds exciting, it is not what computer security is primarily about.

Computer security is the art of ensuring confidentiality, integrity, and availability of compute resources.

CONFIDENTIALITY

Confidentiality is the best understood of the three areas. Every computer has something that needs to be kept private. On one end, there are databases containing the crown jewels: intellectual property. On another end, there are user passwords and other personal information.

Even systems which don't contain sensitive data may have access to one of the largest pools of data at your organization: the network. Network sniffers can gather enough information to leave all your systems vulnerable.

For example, in February 2001 several computers at Stanford were compromised. These computers were used by faculty, primarily for email. Nothing *on* the computer was confidential. However, using the computers, the hacker who broke in gleaned confidential information from the network the computers were on.

```
xxxxxx.Stanford.EDU => yyyyyy.Stanford.EDU [110]
USER user1
PASS password1
STAT
UIDL
QUIT

———- [FIN]

xxxxxx.Stanford.EDU => yyyyyy.Stanford.EDU [23]
 !'''#P
38400,38400#xxxxxxx.stanford.edu:0'DISPLAYxxxxxxx.stanford.edu:0XTE
R
Muser2
password2
elm
jjjjjjjjjjjjjjjjjjjj
———- [Timed Out]

xxxxxx.Stanford.EDU => yyyyyy.Stanford.EDU [21]
(#USER user3
PASS password3
SYST
PORT 171,65,0,0,5,104
LIST
CWD /home/pub/gary
CWD /home/pub/
CWD /home/

———- [Timed Out]
```

While I have removed the actual usernames and passwords, be assured they were just as clear as the words "user" and "password" are above.

The total log file the hacker had of passwords was approximately 8,000 lines long. The point is that just because a system doesn't have direct access to confidential data, it is quite likely it has indirect access to confidential data via the network.

ENSURING CONFIDENTIALITY

Confidentiality is best ensured by using encryption. Any data that is sensitive should be encrypted.

Network sniffers can gather enough information to leave all your systems vulnerable.

Some of the most important data to encrypt is user authentication tokens. Those who still use clear-text telnet and FTP should immediately develop a plan to switch to a model that protects the authentication data from eavesdroppers.

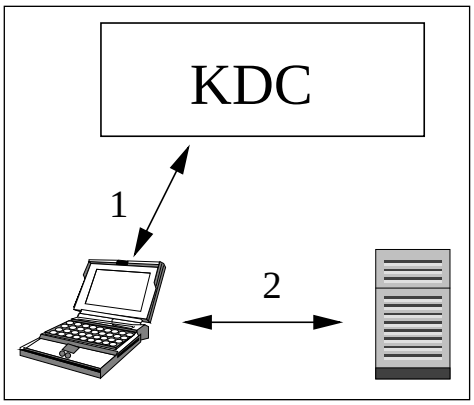
Think back to the cracker's sniffer log file I showed you earlier. Encrypted communication turns those clear-text logins into a jumbled mess of ASCII characters. In that same file was the following entry:

```
xxxxxx.Stanford.EDU => yyyyyy.Stanford.EDU [23]
%%user1%IR.STANFORD.EDU@(P^$:-)':ca<'%.+vc6s}DF~T[f8FLc|vI;#wG\CN6MYI
P%6M-&&&&
& #'$&&Y`&&VT100&
wI\cfCCSDK) >aWHW^H
>rGhsN{q0jxU
`&$$ vQa;j:T8%H>VzL d>7s_
——- [Timed Out]
```

The difference is clear. Those who used clear-text authentication such as telnet and FTP had to change their passwords. In most organizations, this would result in a help desk call. Those who used an encrypted protocol did not have to change their password, saving a help desk call.

This is just one example of how computer security measures save money over the long term.

Universities generally have a large user base that changes rapidly. Stanford, like many, uses Kerberos as our base authentication model. Kerberos provides a central infrastructure for managing user information.



Kerberos uses a Key Distribution Center (KDC), which contains authentication information on all users. (The following is a simplified description of Kerberos. For a more accurate report, please see the reference.) Using Kerberos, a user contacts the KDC for an authentication token (called a ticket). The KDC sends back an authentication token encrypted with the user's password. If the user can type in the password correctly, they can decrypt their authentication token. This is step 1 in the figure.

The user can use the decrypted token (ticket) to authenticate to other services (step 2).

The centralized model gives:

- A single place for adding new users
- A single authentication scheme for adding new services
- An abstraction between service and authenticating to that service

If we need to disable a user, we can do so in the KDC. Further, once disabled in the KDC the user cannot use *any* service. We don't have to go from the email service to the file server to the Web server and revoke login privileges . . . it's all done in one place.

At Stanford, our KDC contains over 58,000 active principles. Every year, each new student is given a principle. Each graduating student has their principle suspended. All of this is done automatically, without manual processes, just as computers should.

Integrity

The integrity of compute resources generally relies upon proper enforcement of protection domains, such as file permissions. When a system is compromised, the cracker can do anything. Besides installing an Ethernet sniffer to grab unencrypted network chatter, as above, the cracker can replace system utilities to hide his presence.

If you don't stop the initial compromise, there is a good chance with a skilled attacker that you will never notice the intruder. You simply won't see the bad guy's changes to the system. For example, a hacker can replace ps so you do not see their processes. They can replace ls so you do not see the files. They can install a kernel module so you cannot see their open files.

ENSURING INTEGRITY

There are two axioms (attributed to Cheswick and Bellovin) when *hardening* a system against attack:

- all programs are buggy;
- if a program isn't run, it doesn't matter if it is buggy.

In essence, these principles mean each computer should ideally only run one service. That service should run with the economy of a mechanism utilizing the principle of least privilege on a stripped down operating system.

Unneeded services and programs should be removed because:

- Programs periodically need to be patched, and fewer programs means less patching, an often time-consuming task.
- More resources are available to the needed service.
- Unneeded services add complexity to the system.
- There will be fewer services to support.

Unfortunately OS vendors ship computers with most services enabled. They do this on purpose: they want the computer to be easy to use. They also assume that the user will customize the system appropriately.

Pushing out secure servers to the end user means disabling all programs that are not needed. At Stanford, we created SULinux to do just that.

SULinux is secure. There have been no recorded compromises of an SULinux system. More importantly, SULinux gives us the opportunity to not just secure the system, but also to integrate the host into our environment. Users don't just see the security, they see the increased usability.

There are approximately five installations of SULinux per day on average, or about 2,000 per year. Assume a low-ball estimate that each compromise costs the university approximately $300 to fix. Costs include downtime and employee time to reinstall, regardless of whether research or other data was modified. Given the current rate of scanning, it's appropriate to assume any unpatched system would be compromised at some time or another.

Thus, the savings can be estimated at 2,000 hosts x $300 = $600,000 per year. This is only a ballpark figure, but it demonstrates the scale of the problem and the possible savings from implementing security solutions.

Most operating systems allow for automated installation. By distributing hardened versions of the OS, whether it be through Windows RIS or Ghost images or IRIX roboinst, you can significantly increase the security and integrity of compute resources, while at the same time allowing users to easily integrate into the infrastructure provided by central IT.

AVAILABILITY

The goal for computer services is 99.999% availability. Computer vendors tout the number, yet realizing the 5 9's in the real world is difficult.

There have been no recorded compromises of an SULinux system.

The company was caught with their proverbial pants down.

Achieving 99.999% availability in practice requires technology that will work consistently. Remember that a secure program is a correct one, and it will produce an answer or error whenever possible. Availability is built upon programs functioning in exactly this manner. Hence, a highly available service must be built upon secure components.

Today, there are many threats to availability, including system intrusions, DoS attacks, and service hijacking.

Computer security can increase availability by keeping hackers from compromising systems, creating robust services that resist DoS attacks, and properly securing your domain to prevent hijacking. Though computer security alone can't accomplish each of these things, it can facilitate such an environment when implemented in a consistent and serious manner.

For example, one of the leading companies in computer security had their Web site defaced in February 2000, right before the widespread DDoS attacks. The hacker changed the Web site by compromising the DNS server, pointing their main splash page to point to a Web server in Brazil. The company was caught with their proverbial pants down. Their site was defaced for over 13 hours, along with significant downtime of their primary DNS server while they fixed the problem.

The reason: a hacker named Dennis Moran, age 18, who lived across the country on the east coast. The lesson here is security can not only allow the organization to save face, but it can also help minimize downtime. In this specific case, a coherent security plan may have:

- Prevented the system compromise.
- Detected the system compromise.
- Planned for a backup server to minimize downtime in case of compromise.
- Provided staff so that compromises could be reported and resolved quickly and easily.

Each would have reduced downtime. Each must be planned and implemented before the compromise.

How to Start

There are consultants charging well over $500/hour who will help you implement a security architecture at your organization. If you need a consultant to overcome political boundaries, by all means hire one.

If you don't have the cash, I'll tell you for free what to do. First, create a position or office for a person who will be in charge of computer security at your organization. At Stanford that office is called the Computer Security Office. Having a central authority is the only way I know of for security to be effectively implemented in any organization. Authority over computer security should be given to the person responsible for it. This may sound trite, but it is the most often overlooked aspect of creating a sound infrastructure. (A corollary: if you are ever offered a security position without authority, run away.)

Next, you should find the correct people to be responsible for computer security in the organization. The correct person will understand computer security risks and be able to evaluate them with a long-term perspective. Those without a long-term perspective generally do not last.

Last, you should let the people do their job. If you have given security the proper position and authorization, and you have selected the right people, the rest will take care of itself. This is because security is a process, not a goal.

This isn't just what you should do, this is how Stanford implemented the Computer Security Office. Our policies give the office responsibility, and we have hired the best talent around.

Policies

Effective policies are essential to the success of an organization. Foremost, policies should give the community the expectations surrounding the use of compute resources. While policies often differentiate between acceptable and unacceptable behavior, their purpose should be to let the community know how the compute resources are to be used to realize the goals of the organization.

For example, at Stanford the goal of the organization is learning. Our compute resources are for academic pursuits. Our policy explicitly mentions that only incidental non-academic use is tolerated.

Second, policies provide a consistent mechanism for addressing the eventual security incident. More, carefully crafted policies and procedures provide a chain of command so that incidents can be dealt with quickly, efficiently, thoroughly, and consistently.

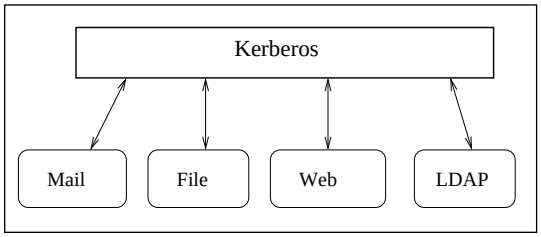
At Stanford, the computer-use policy appoints the computer security officer at the top of the chain, with administrators and then users following.

The Plan

At a high level, the critical areas to plan well are

- What base-level authentication system to use
- How to ensure system integrity
- Educating the community on using that infrastructure
- Educating the community on security-related matters

The answer Stanford has come up with is we will use Kerberos for base-level authentication. Authorization is handled through the local services, such as AFS file permissions.



Everything hinges on our authentication choice. Kerberos is used to authenticate for file access (AFS), Web access (WebAuth), directory services (LDAP), and even enrolling in classes (AXESS)

Ensuring system integrity is not complete. Our partial solution includes SULinux for Linux, Norton Anti-Virus for Windows, Best Use Documents for other OSes, and periodic vulnerability assessments.

The Computer Security Office, along with others, educates the community. We give talks, presentations, and provide documentation and tools to the community.

Quick Response

On May 29, 1999, over 30,000 people received a hate mail derogatory to certain racial groups. The mail was forged to appear to be from a Stanford engineering student, probably in order to exact some sort of revenge.

USEFUL URLS.
SULinux – *http://sulinux.stanford.edu*

Stanford Security Office – *http://security.stanford.edu*

Stanford Kerberos Infrastructure – *http://lelandsystems.stanford.edu/services/kerberos/*

David's Site – *http://www.theorygroup.com*

Over 30,000 people believed that this student sent the mail. Within an hour angry people were outside the innocent student's residence. The situation was critical.

Without proper resources the incident could have turned into a hefty political problem. However, Stanford was able to act quickly because the Computer Security Office was already in place to handle this sort of problem. Within 14 hours of the hate mail, the Security Office was able to identify suspects and distribute a response from the university president. The quick response turned a potentially disastrous situation into a positive racial-awareness campaign.

While not everyone will have a similar hate-mail incident, every organization will at some time need leadership during a computer security crisis. It may be an Internet worm or a DoS attack. Having a group for computer security ensures a quick response.

As a Public Service

In February 1999, a computer intruder who went by the nickname "ShadowKnight" compromised several Stanford computers. Some of the computers compromised were responsible for one of NASA's largest and longest-running research projects, with a multi-million-dollar budget.

On November 6, 2000, Jason Diekman, aka "ShadowKnight," appeared before a United States district judge and pleaded guilty to recklessly causing damage to a protected system, unauthorized access of a non-public computer, and unauthorized use of an access device.

Our office coordinated the investigation with several Stanford network administrators. The network administrators were capable of assisting because of a commitment at all levels to providing a safe and secure Internet. The results of that investigation were turned over to the FBI, who then arrested and prosecuted Mr. Diekman.

Protecting People

A CAT scan uses radiation to map out the human body. A CAT scanner is a computer, normally running SGI's IRIX operating system.

IRIX machines by default have a "guest" username with no password. These same IRIX boxes with the "guest" account are used in CAT scan machines. Imagine a hacker simply logging into your local medical facilities computer and viewing or changing the results of this diagnostic procedure.

Critical infrastructure, such as the CAT scan machine, should be identified. After identification, regular audits protect the organization from liability. More, audits alert people to potential problems before they affect the people who use the equipment daily.

Summary

Creating a security infrastructure is just like planning, implementing, and deploying any other service. While you may receive more recognition for your deployment of a Web mail service, the long-term safety of the community is considerable compensation. Though many may be angry when they must change the way they compute, you are giving the community good habits that will follow them into careers outside the university. Finally, implementing a coherent computer security strategy may protect you, your organization, and your coworkers from harm done by computer intruders.