Adverse Termination Procedures -or- "How To Fire A System Administrator"

Matthew F. Ringel
Thomas A. Limoncelli – Lucent Technologies/Bell Labs

ABSTRACT

When an employee is terminated, his or her access to the organization's network and computer systems must be removed. However, the most difficult employee to terminate is often the person that built the system. We propose a three tier model for coordinating access removal that is useful in normal and adverse termination scenarios. We then work through a number of case studies to see how the model performs in this difficult situation. We feel this model performs extremely well. We also discuss, informally, how to minimize the risk of backdoors and how employees can reduce the possibility of being blamed for security incidents if they are terminated.

Introduction

One of the more unpleasant subjects in the working world is how to deal with a person who is being forcibly terminated from a company, possibly without warning. This could range from calling the person at home and telling them not to come into work again, to something as confrontational as two security guards meeting the employee at his cubicle, Human Resources informing him that his personal effects will be shipped to him, and the security guards escorting him out of the building. Either way, the outcome is a potentially unhappy ex-employee and, if the person in question is technically adept, a potential liability for the company's computer infrastructure.

This paper will discuss system administration practices that are used to reduce the risk of attack by former employees. Several companies have been interviewed to get an idea of the best current practices in the field. This paper will also consider what can be done to reduce the risk of backdoors that might have been installed by a former employee. In addition, this paper will also examine the terminated employee's perspective. It will offer advice on how to reduce or eliminate the possibility of being blamed for security incidents after you leave a company on not-so-amicable terms.

Motivation

Adverse termination tends to be looked at as a very rare exception, when in fact it happens more often than expected. No one likes to be forcibly terminated and, all kidding aside, there are very few system administrators who take pleasure in dealing with the termination and clean-up of recently departed employees. Therefore, termination procedures get about as much attention as one might expect for something deemed to be an unpleasant reality: ad hoc solutions when necessary, and not much in the way of formalized action. The authors note that larger organizations – corporate and academic – are much more likely to have termination procedures in place. With this in mind, this paper primarily focuses on small- to mid-size sites that are looking to go from ad-hoc solutions to a more formal approach. Larger organizations might find new ideas for looking at the complex issue of adverse termination.

The reasons why terminated employees need to be locked out of an institution's systems in an efficient and thorough way should be fairly obvious. What may not be so obvious are the measures and procedures that can be put in place to reduce the security risks created by a disgruntled ex-employee with access to your network. The purpose of this paper is to give some suggestions as to how one might establish a termination procedure, as well as some ways to put it into effect.

The Three Tier Model

The authors propose that the following three tier model should be used when constructing a procedure for removing access for any employee. The benefit of this model is that it can be used as the starting point when writing a formal policy for normal or adverse terminations, or used as a check-list for ad hoc situations. An example of an ad hoc situation would be a small company that does not see the need for a formal policy, but finds itself terminating a system administrator.

There are three aspects one must be concerned with when removing access: Physical access ("Can she get into the building?"), Remote access ("Can she remotely access our network?"), and Service access ("Have access rights to applications been withdrawn?"). This model is somewhat fail-safe because, as will be shown later, even if one of the three tiers isn't secured properly, the ex-employee will not be able to do damage.

Physical access means, quite simply, "Make sure they can't get in the building(s)." Usually this is the function of Human Resources or Corporate

Security. For example, Human Resources should already have a procedure for retrieving the employee's ID badge, which is checked by guards at the building entrance. If card-key access is used, the card-key system must be programmed to deny access to that card-key whether or not the card-key has been returned. Keys to any rooms must be retrieved or locks changed. Combination locks, safe combinations, etc. must be changed. If there are there multiple buildings, additional questions must be asked. Is there a barn, maintenance shed, shack, outhouse, dog house, or annex that must be considered? Because physical instruments like doors and keys have been around for longer than computers, most companies usually have good procedures related to them already. Follow them. Very small companies might not have any kind of identification badge or similar system in place. In that case, have the receptionist or other employees been notified of what to do if they see this person in or near the building?

Remote access refers to the many ways one might get into the networks. These include, but are not limited to, modem pools, ISDN lines, in-bound connections through a firewall, unfirewalled internet, and VPN service. Access to all of these systems must be disabled. This can be difficult if each of them is run by a different team or has a different access control system.

Service access refers to the applications and services that are inside the network. Each of these services usually has a password. Examples include IMAP4/POP servers, Database server, UNIX servers (each with their own /etc/passwd file to be checked, or a global NIS or Kerberos database to be checked), SMB servers (NT File Server), etc.

Logistically speaking, a team of people can be assigned to each tier. This three tier model is somewhat fail-safe, because if any single tier is not fully implemented but the other tiers are, then he or she will not be able to negatively affect the network. For example, if a UNIX account hasn't been disabled, but the ex-employee can't get into the network, he or she can't do any harm. If the ex-employee can get into the network via modem, but all services are disabled, the ex-employee will not be able to do damage (or will only be able to do the same damage internal employees are currently doing). This is where a single authentication database can save a lot of time. If all services are authenticated by NIS (or NT Domain, or SecureID) then there is usually only one database that must be updated to remove access. However, systems that use such a database often have "local" databases as well. UNIX hosts using NIS for password data still have an /etc/passwd file that is accessed before referring to NIS. NT has "local accounts." In this case, it can be useful to have an audit script that will seek out such local accounts.

Case Studies

Case Study #1: Large University – An Adverse Termination

Academic entities tend to be very astute about termination procedures, as they often have batches of terminated accounts at the end of every academic year. The university setting is a good example of how "practice makes perfect." However, our case study involves someone with privileged access to many machines.

At Large University, a long-time operator with root access to all UNIX systems was to be terminated. After some discussion it was decided to use the following procedure. A small team was assembled to list all the access she had and how to disable each, including card-key and host access. When the designated time arrived she was told her boss needed to speak with her in his office. Her office and her boss's office are in opposite parts of the building and the trip would take at least 10 minutes. By the time she reached her boss's office all of her accounts had been disabled.

Evaluation: Since Large University is a public university, they were not able to eliminate physical access to the building, but card-key changes prevented the person from gaining direct physical access to sensitive machines. Remote access could not be disabled since Large University does not use a firewall. Because the operator had access that was so extremely pervasive, the only way to assure complete coverage was to have all the senior system administrators involved in removing all Service access. Grade: A-

Case Study #2: Small Software Development Division (SSDD) – A Good, But Flawed, Termination

A system administrator left SSDD and offered one month's notice to help the company transition. The system administrator was actually involved in interviewing a replacement. All responsibilities had been transitioned the day before the termination date. As previously arranged, on his last day the system administrator walked into his boss's office and became root in a window on his workstation. While the boss watched, he disabled his own regular login. He then issued the command to change the password of various routers and let his boss enter the new password. He then repeated this for a few other "system accounts." Finally he issued the command to activate their "change root password on all systems" procedure and let his boss enter the new password. The boss was shocked that the soon-to-be ex-employee was doing such a complete job of transitioning duties and deleting himself from the system. Finally he turned in his card-key and physical keys and left the building. About two weeks later the ex-employee remembered that he had forgot to change the password on a particular non-privileged system account on a machine that had a modem directly attached to it. The ex-employee reports that he confirmed with former co-workers that the ex-employer didn't disable the account until he notified them of the error.

Evaluation: With one exception, the termination was complete. However, the exception crossed Remote and Service tiers because the account (Service access) was on a host that was directly connected to the outside world (Remote access). Also, the authors do not feel it was safe for the new passwords to be set in front of the exiting employee, who could have been watching the keyboard. The employee also could have recorded the new passwords as they were being set. The garden-variety *script*(1) command does not record non-echoed input (such as passwords). However, a custom version of the *script*(1) command (let's call it my_script) that could record non-echoed input would make recording the passwords easy to implement.

This short script would allow the exiting employee to capture the changed passwords:

```
cd /tmp && my_script
Mail < typescript secret@example.net
rm typescript my_script;
```

The company took a significant risk by not disabling access as soon as notice was given. However, it was a reasonable risk to take considering that it was not an adverse termination. Grade: C-

Case Study #3: Large Hardware Company (LHC) Terminates The System Administrator's Boss – An Optimal Termination

LHC had to "suspend, pending investigation" the manager of a team of system administrators. As a result of the investigation, the employee was terminated. This person had administrative access to most systems in his domain and had access to the network via most of the remote access methods that the system administration group provided.

Without the accused manager knowing, a meeting was called by the accused manager's director, the lead system administrator, and corporate security. The corporate security officer explained the situation and the action plan. The lead system administrator was to change all the "root" (and privileged account) passwords that evening. In the morning, the system administration group (without their manager) would meet and be informed of the situation. They would suspend all access to the systems. If something could not be suspended, access would be deleted. The system administration group used an action plan similar to the authors' three tier model.

The lead system administrator spent the evening changing the root password on every system. In the morning, the system administrators were brought into a closed meeting and the situation was explained to them. The physical access aspects were taken care of by corporate security. The system administration team brainstormed on all the Remote and Service access issues. Breaking the problem down into two tiers focused their discussion.

One potential problem was the manager's home ISDN service. It would be 12 hours before the ISDN admin (a system administration team member currently on vacation) would return. The manager's home ISDN equipment had been surrendered, but it was possible that he could have programmed other ISDN equipment to enter the system. Even so, the risk of that happening was low, and the system administration team was confident that Remote and Service access had been sufficiently removed and logs could be monitored for intrusions. (In other words, doing a complete job on two of the tiers should compensate for being incomplete on the third tier.) Two hours later all access had been disabled.

The instructions to the team also included two important elements:

1. Keep a log of what is disabled and forward logs to the lead system administrator, who will maintain a master list.
2. Disable, do not delete. Out of respect for the manager, they must do all of this under the assumption that the manager will be cleared of charges. If and when the manager is exonerated, service will need to be restored immediately. Every service that they forget to re-enable will be a sad reminder of the entire ordeal. The system administration team did not want to have those painful reminders pop up months later.

During the investigation, the accused manager resigned. The logs were useful as a check list of what access had been suspended and now needed to be deleted.

Evaluation: This case study used the three tier model perfectly. Note the increased efficiency gained by splitting into teams, and how the inability to disable the home ISDN access was compensated by effectively removing all other access. Grade: A+

Case Study #4: Small Financial Company (SFC) Fires Their Lead System Administrator – A Worst-case Scenario

SFC was going to fire their lead system administrator due to non-performance. The lead system administrator had already been warned several times to improve his performance and knew that his termination was imminent. On the day before the termination, a junior system administrator (the only other system administrator in the company) was informed of the termination and given the task of revoking Remote and Service access for the lead system administrator by noon the next day.

The lead system administrator came into work the next morning and found that though he was able to log on to various administrative machines, his mail spool had been archived and deleted from live storage. Since the lead system administrator knew then that he would be terminated that day, and given that he had a considerable (and long-standing) grudge against the company, he swung into action. Due to an ever-changing network layout, the lead system administrator had long ago planted a UNIX host of his own on one of the company networks without anyone noticing. The machine had been passively sniffing packets going across the internal network, capturing passwords over time. Once the lead system administrator knew that he was going to leave, he set the machine to a much more active mode, starting various services that would allow him (or anyone he gave access to) to get in from the outside.

As expected, two hours after arriving for work, two security guards and the lead system administrator's manager arrived at his desk, led him to a meeting room. Once there, he was told that he was fired, that he would be escorted off the premises immediately, and that his personal effects would be shipped to him. SFC was conscientious in making sure that he had no physical access to the property and posted security guards for the next week to assure that he wasn't going to try to physically break in.

Two months later, the remaining system administrator and the replacement for the lead system administrator noticed that they had an unusually large amount of traffic coming into their internal network from the outside world. As they started investigating that issue, the Chief Technical Officer called the system administrators, informing them that SFC's web pages had been defaced, and that the company was fielding inquiries as to whether or not SFC had been hacked. The new lead system administrator verified that the company's web pages had been defaced, noted that there was a large amount of data flowing out of the site, and went into the machine room and pulled the plug on the router connecting SFC to the Internet.

The next several weeks were spent doing damage control (both network and corporate image), figuring out when the intrusion happened and how much damage had really been done, and then re-installing operating system and application software from media. Eventually, the system administrators found the rogue machine, completely devoid of data (no doubt it had erased itself when it was cut off from the world for an extended period of time). Unfortunately for the company, the first signs of intrusion occurred about a month before they noticed the upswing in traffic. The swell in traffic was correlated to a message posted in a Usenet newsgroup, saying that the following system was "owned," along with directions on how to access it. The system administrators hypothesized that the former lead system administrator has entered and compromised the network remotely about a month before the Usenet posting; these were the first signs of intrusion they found. The Usenet posting, as well as the self-destruction of the rogue machine, were meant to cover the original infiltration.

Evaluation: Although physical access was secured remarkably well, the Remote and Service access tiers were secured haphazardly or not at all. No firewall will protect you against an internal attacker, especially one who is mostly passive (i.e., the rogue machine). Grade: F (although an honorable mention is given to the replacement lead system administrator for understanding that when feasible, pulling the plug is one of the most effective ways to secure a network).

Checklists

We have assembled several checklists of things to disable in the event of an adverse termination. While no checklist is complete, these can serve as a basis for your own procedures. You might want to use your company's "new hire" procedure as a starting point. In general, whatever is done for a new hire has to be undone for a termination.

- Physical access (Tier 1):
  - Change combination locks
  - Change all applicable safe combinations
  - Doors with keys should have locks changed (even if keys are returned)
  - Have ex-employee surrender:
    1. keys
    2. card-keys
    3. hand-held authenticator cards
    4. PDAs (Pilot, Newton, etc.)
  - Remove access for all buildings (e.g., remote locations, shacks, utility buildings, etc.)
- Remote access (Tier 2):
  - Modem pools
  - ISDN pool
  - VPN servers
  - In-bound network access (i.e., ssh, telnet, rlogin)
  - Cable Modem access
  - xDSL
  - X.25 access
- Service access (Tier 3):
  - Database servers
  - NIS domains
  - NT domains
  - Netnews: "news," "usenet," and "uucp" passwords
  - RADIUS servers

Improving Accuracy

A good inventory of possible access points improves your accuracy. If these case-study sites had had well-managed inventories, they wouldn't have had to do as much last-minute brainstorming.

A good way to improve accuracy is to reduce the number of access databases. Large numbers of access databases (for example, an /etc/passwd file on every machine, embedded passwords in routers, etc.) increases the amount of work needed to remove access and increases the chance that coverage will be incomplete. If all access was controlled by a single database (impossible in today's environment), all access could be disabled from a single point. In case study #3, access to many services (VPNs, in-bound telnet through the firewall, root access, etc.) were controlled by hand-held authenticators (HHA) keyed to a particular database. Disabling the manager's record in the HHA database resulted in immediately disabling many different services at the same time.

Preventing Backdoors and Logic Bombs

We have all heard a variation on the story of a disgruntled employee who sets up a program at work that he must deactivate every week or it will cause massive damage within the company's system infrastructure. Sure enough, the person is terminated, and the program "goes off," wrecking two months of work by everyone in the company (or something similar). While the story itself might be apocryphal, it is entirely possible to build such a program. In addition, there's the story of the terminated employee who kept a modem hooked up to a spare analog line in the data center, and was able to access various internal systems from the outside, completely undetected by any intrusion detection infrastructure. This story is also apocryphal by itself, but is an entirely likely scenario.

The first story included an example of a "logic bomb": an autonomous program that is set to trigger a detrimental event under some set of conditions that is known only to the writer. The second is an example of a "backdoor": a covert way of entering a system or network while bypassing security that was installed after the system went live. Either one could be planted by someone who is leaving the company in the hopes of getting revenge at some significantly later date. If the perpetrator has administrative access, there are any number of ways to hide a logic bomb or a backdoor, but there are ways to increase your chances of finding or eliminating them before they do their dirty work.

One possible method of sweeping for logic bombs or backdoors is to compare each machine to which the person had administrative access (a "touched" machine) with another machine that has a known-good "/" and "/usr" filesystem. Ideally, you will have taken a snapshot of the checksums when the machine was put into production (using tripwire [1] or a similar tool) and placed it onto read-only media in a safe place. The question of how to keep that media safe is beyond the scope of this paper, although one of the authors saw a bank safe-deposit box used to great effect for that purpose at a previous job. Maintaining checksums of "/" and "/usr" for each production machine can be cumbersome, but is not hard to automate and will save you trouble in the long run.

Next, check the *crontab*(1) and *at*(1j) files for every user on each touched machine. Every job must be accounted for. A machine might appear to be running perfectly, but you don't know what the former employee might have inserted into those files, especially if they're large enough (e.g., root's *crontab*(1)) that no one really holds them up to close scrutiny.

After that, you should probably make a thorough accounting of all the daemons running on each touched machine. Much like *cron*(1) and *at*(1), there is usually so many programs running on any given production machine that yet another daemon will get lost in the noise.

Checking the checksums, cronjobs, and atjobs on each touched machine is a very good first step to show that a machine is clean from the most blatant and simple attacks. It should be noted that it is not difficult for a skilled person to hide themselves extremely well. For the cautious, one should put a statically-linked copy of *ps*(1) on the machine, and check for daemons using the statically-linked version of the command. The same goes for a statically-linked copy of *crontab*(1) and *more*(1) when examining other files on the system, just so you know you're running commands with uncorrupted libraries.

One last thing to check for is machines that are passively sniffing packets on your network. As mentioned in Case #4, no firewall will stop an internal attack, and a machine that passively collects data and stores it for later transmission or retrieval is one of the more straightforward and effective backdoors. As of this writing, there is a publicly available tool called AntiSniff [2], which uses a combination of OS-specific and protocol-based transmissions over Ethernet to determine whether a machine is passively sniffing packets from the network.

All the advice given above is for the purpose of maximizing uptime while doing your audit. There will be times when the terminated employee in question is highly skilled and will be able to hide themselves from even a reasonably close inspection of a system. Sometimes taking a system down and re-installing from known-good media is the only choice, such as in Case #4.

A full backdoor audit checklist is outside the scope of this paper. The authors recommend the "Server Security Checklist Overview" [3] and "UNIX Configuration Guidelines" [4] as papers that investigate a large variety of typical system installation pitfalls which may be used as backdoors, and how to protect against them.

The Other Side of the Coin

Now that we have taken a look at the employer's side of the equation, we will talk about the employee's side: what to do when you're the one going out the door, and not necessarily under friendly circumstances.

The authors would like to note that they are not lawyers, and that the law is often ambiguous in this area, mostly due to lack of court precedent. The only advice that we can give with confidence is that if you are terminated, immediately remove your hands from the keyboard and never touch a computer of your employer’s again. That, plus a good lawyer, is the best (albeit possibly flawed) advice that we can offer.

If you are being terminated unilaterally for whatever reason, and you’ve had administrator access on any machine that is considered important, it is safe to assume that your soon-to-be-former company considers you a security risk, as you are not only knowledgeable, but possibly disgruntled. Therefore, you want to make sure that the company has absolutely no reason to believe that you are responsible for any computer security mishap that happens in the future.

There are two scenarios to consider:

- When you see the termination coming.
- When you don’t.

It is beyond the scope of this paper to discuss ways to tell that your termination is imminent. It’s not hard to figure out when your employer is planning for your departure, even if your employer hasn’t told you yet. If you see termination in your future, here is a list of things that you can do to make sure that you leave in good faith, if not on the best of terms.

- Make a list of the machines to which you’ve ever had administrative access. As a system administrator, your employer will assume the worst. He might suspect that you have planted logic bombs, especially on machines that made their way into the data center without much administrative scrutiny. Volunteering all the information you can is your first priority. It is a double-edged sword, though. If you miss a single machine, you open yourself up to being accused of hiding something.
- Be civil. It sounds obvious, but when you’re getting terminated against your will, there will be hostility and you want to make sure things go as smoothly as possible.
- Set personal files aside. The authors believe that personal and work files should stay completely separate, but accidents happen. Assume that your employer will go through all of your files, including your voicemail and mail spool. Set aside a well-marked directory with your personal files, and mention it to your superior when the time comes. If you are up-front about the information, you’re much more likely to get it all back intact. Do not be surprised if they do not permit access to the data, though. Make sure you understand any and all relevant policies. If your employer forbids the storage of personal data on their computers and you have personal data clearly marked and set aside, your employer has further grounds for termination.
- If you haven’t already, make professional connections with your fellow employees. The people you work with will be the best defense of your character if your superiors suspect foul play by you. They are also a good source of references for your future job search.

When you don’t see the termination coming and you’re pulled into a meeting similar to the person at the Large University mentioned in Case #1, the best you can do is volunteer as much information as possible. Give all of your passwords, be very businesslike, make sure that you get what’s coming to you (i.e., vacation pay, severance, other benefits), and walk out the door without a fight. It might hurt your pride, but in the end it will save your professional reputation.

DO’s and DON’Ts

There are a few rules of thumb that can be drawn from the technical and non-technical recommendations mentioned in this paper.

For the Employer

DO the job quickly. Once the process starts, don’t let it drag on. If you do, you will have a demoralized employee on your payroll.

DO make sure that you can effectively ‘‘flip the switch.’’ Make sure that you and your staff are ready to remove all access at a specific time. As an example, having email access go away a few hours before account access will not only look unprofessional, but tip off the employee, especially if they are hostile.

DO make sure that all the necessary staff is informed of the termination. No more, and (more importantly) no less.

DO prepare to re-install the operating system on any touched machine. You can lock the person out from the machines, but their programs might still survive. As a good first check, you can check the crontab and atjobs to make sure that everything running has a well-known purpose.

DON’T get cute or clever in finding a way to communicate to the person that they are being terminated. This is not a game, and your employee is a human being with emotions. Your Human Resources department will have specific policies about how the communication is to be done.

DON’T treat the employee like a criminal. Mutual civility is important. The person was hired as a professional and you must respect that status during termination proceedings. Besides, you might be looking for a job someday, and this employee might be part of the hiring process.

For the Employee

DO make sure that all your project work and documentation is in order. Actively making the transition easier goes a long way to reassuring people that you’re leaving the company in good faith.

DO volunteer all the information you can about what was in your control, the status of your projects, and all of your passwords. Once again, this is all about leaving in good faith.

DO resist the urge to give a "parting shot" by sending out email to all of your coworkers that you're leaving and "going to a better place." It might make your former employer question your professionalism, which might cause them to see you as a security risk.

DON'T mix work and personal files. You will eventually leave the company at which you're currently working and separating the two gets harder with time. In addition, your employer probably has a policy about keeping personal data on corporate computers. Usually, the policy forbids it. Even if you have a more liberal policy, it may be a good idea to adopt a personal policy that is more strict.

Conclusion

Securing systems after employees have been terminated is a problem that is not given much attention because of the unpleasantness of the subject. This results in ad-hoc solutions that are of varying effectiveness. The authors present a more formalized model. The process of securing systems from an employee who has just been terminated can be broken down into three tiers: Physical access, Remote access, and Service access. The model is fault-tolerant and is still effective if one of the three tiers is not secured properly.

The primary benefit of the three tier model is that it provides a structured approach to securing a network against a knowledgeable attacker, with the additional benefit of providing staff with a way to look at securing the network from other kinds of attackers as well.

This model is one of many different approaches to the issue of securing systems against terminated employees. The authors hope that this paper will encourage others to give thought to this often overlooked and difficult subject.

Author Information

Matthew F. Ringel received a BS-CS from Columbia University in the City of New York in 1995, after which he worked as a system administrator for Phantom Access Technologies and PSInet/Pipeline New York. He currently lives in Cambridge, Massachusetts and is a network administrator whose current projects include process design for a network operations center and large-scale implementation of several network management systems to monitor a major web-hosting provider. Reach him electronically at ringel@mithril.org.

Tom Limoncelli is a MTS at Bell Labs, the R&D unit of Lucent Technologies, where he is chiefly concerned with the architecture and operation of the data network for much of Research. Tom started doing system administration on VAX/VMS systems in 1987 and switched to UNIX in 1991, and in 1996 decided to focus on networks, not operating systems. He holds a B.A. in C.S. from Drew University, Madison, New Jersey. Reach him via U.S. Mail at Lucent Technologies, Room 2T-408, 600 Mountain Ave, PO Box 636, Murray Hill, NJ 07974-0636. Reach him electronically at <tal@lucent.com>. His web page is http://www.bell-labs.com/user/tal .

References

[1] Tripwire Security Systems *The Tripwire Intrusion Detection System*, http://www.tripwiresecurity. com/prodintro.html .

[2] L0pht Heavy Industries *The Goal and Purpose of AntiSniff*, http://www.l0pht.com/antisniff/purpose.html .

[3] Vandenberg, P. D. J. and Wyess, S. *Securing Solaris Servers – A Checklist Approach,* Feb. 1999, http://cne.gsfc.nasa.gov/security/sun_security/index.html .

[4] CERT Coordination Center *UNIX Configuration guidelines,* Oct. 1997, ftp://info.cert.org/pub/tech_tips/UNIX_configuration_guidelines .