## Information

Jeffrey King
GIAC Security Essentials Certification v1.4b
“10 Vulnerabilities a Scanner Might Not Find”

## Abstract

In a world where services are becoming economically more emphasized than products, those organizations seeking to remain in the products sphere survive due to differentiation alone. However, despite the billions of dollars spent on security products and services, the innovation and inventions behind these organizations remain dangerously exposed to theft, destruction, and modification. This paper presents 10 vulnerabilities a scanner might not identify.

## Audience

Even the most diligent or most talented system administrators forget the fundamentals. Consequently, they are members of my intended audience. Members also include executive level personnel, who sometimes need starting point for their security planning.

## Introduction

An unfettered oligopoly of powerful hardware and software companies continues to transition the orientation of the global technology industry from products to services. With items like closed-source software and proprietary standards, the majority of organizations wishing to enter and compete in the sphere are integrators or consultants. And this transitioning has permeated the into the academic community. As Rob Pike states in the thesis of his rather eloquent polemic *Systems Software Research Is Irrelevant*, “at a time when computing is almost the definition of innovation, research in both software and hardware at universities and much of industry is becoming insular, ossified, and irrelevant”[1]

For those organizations attempting to compete in the services sphere, take a number, and get in line. Differentiation is premised on political prowess and rearrangement of buzzwords[2] within corporate mission statements. Barriers to entry are low: primarily financial capital, human capital, and halfway decent marketing. And the market is vast, as our society has become complacent with existing technology, demanding seamless integration rather than innovation (in fact, some declare seamless integration as ‘innovating’).

For those organizations attempting to compete in the products sphere, best of luck! Barriers to entry are high, as dependence on capital is low, reverse engineering is arduous, and marketing is irrelevant (obviously these are gleaming generalizations). However, the market is initially small. Therefore, differentiation is premised on innovation through invention, management of public perception, the aegis of a knowledgeable leader, and perhaps miracles of God.

However, despite billions of dollars[3] being poured into information security, organizations are *still* failing to adequately protect the innovation/invention part of the latter-mentioned equation. This paper seeks to: isolate the common misconceptions and mistakes associated with basic security; propose solutions to these misconceptions and mistakes.

**Problem 1: Failure of technical personnel to adhere to technical policies.**

Security policies begin as broad declarations at the executive level and transcend the organization, multiplying exponentially until they arrive at the enduser community as finite technical rules. Hundreds of papers have been written about the construction and implementation of these policies, focusing primarily on how to best promulgate them to organizational stakeholders. However, one issue seems to consistently arise: the ignorance of technical personnel.

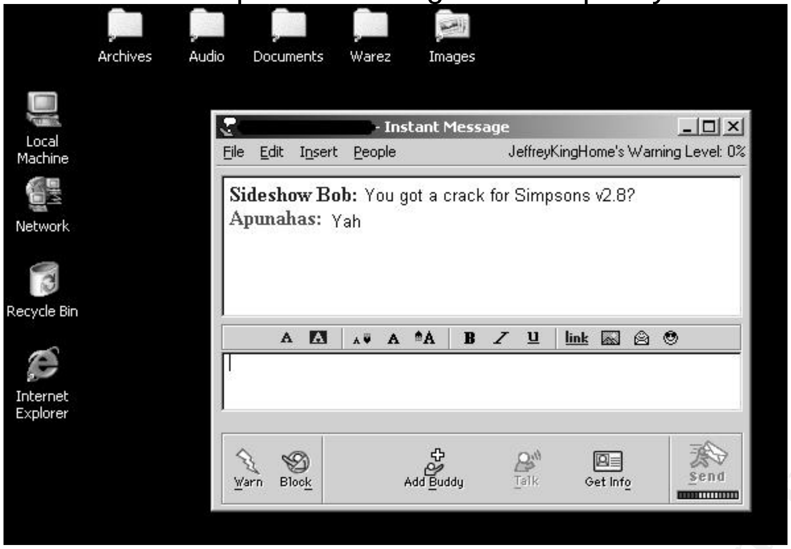
Here is an example concerning Solaris password policies:

```
# grep test /etc/passwd
apu-test1:x:103:100:Apu Test 1:/home/apu-test1:/sbin/sh
apu-test2:x:104:100:Apu Test 2:/home/apu-test2:/sbin/sh
apu-test3:x:105:100:Apu Test 3:/home/apu-test3:/sbin/sh

# grep apu /etc/passwd
apunahas:x:101:100:Apu Nahasapemapetalan:/home/apunahas:/sbin/sh
apu-su:x:102:1:Apu Nahasapemapetalan:/home/apu-su:/sbin/sh
apu-test1:x:103:100:Apu Test 1:/home/apu-test1:/sbin/sh
apu-test2:x:104:100:Apu Test 2:/home/apu-test2:/sbin/sh
apu-test3:x:105:100:Apu Test 3:/home/apu-test3:/sbin/sh
```

In this case, we can see that has Apu Nahasapemapetalan has five accounts on the system: one regular account, one account for su'ing, and three test accounts. For the most part, this is generally typical and benign system activity. However, there are a few concerns: Apu Nahasapemapetalan stopped administering the system over a year ago; organizational policy forbids the use of test accounts; and, upon investigation of the aging of the accounts, none of the passwords expire.

Here is an example concerning software piracy:



**Sideshow Bob:** You got a crack for Simpsons v2.8?
**Apunahas:** Yah

In this case, we can see administrator Apu Nahasapemapetalan: readily exchanges cracks with his colleague Sideshow Bob; maintains a “Warez” folder on the file server; and uses America Online’s (AOL) Instant Messenger® as a messaging client. Once again, there are a few concerns: organizational policy (as well as Federal law) forbids software piracy; organizational policy forbids using an administrative machine to store personal software; and organizational policy forbids use of instant messaging clients from administrative machines.

The latter examples demonstrate an all too typical problem: technical personnel exempting themselves from technical policies. Simple root passwords. Cisco ‘enable’ passwords kept the same across scores of routers. Passwords written on index cards. These are some of the dozens witnessed over years of penetration testing and auditing.

**Solution 1: Accountability reigns king.**

Typically organizations attack the above problem from a policy perspective. They modify their existing policies to include consequences for noncompliance. They write *new policies* to address shortcomings of old policies. The send out memorandums indicating existing policies extend to all personnel, including tape jockeys, network ninjas, and Network Operations Center (NOC) monkeys. Naturally, dissemination of the new policies  is ignored (or never occurs). Memorandoms are deleted from inboxes or tossed in the recycling bin with the latest copy of PC World. Consequences go ignored or are improperly enforced.

I am a big fan of accountability. And not just activating the Basic Security Module (BSM) in Solaris or backing up Windows 2000 event logs to CD-ROM. Though these are import activities, they are only part of a comprehensive auditing program which should answer the following questions:

- What activity do we capture on a given system? If the system is of a certain classification level, auditing standards might actually be dictated by an existing Federal standard. If the system is an extremely high risk system (but not formally classified), process level auditing might be considered. If the system is an high risk system, perhaps process level auditing is a bit too much. In essence, the type of activity captured on a given system should be commensurate with the nature of the system. Many administrators capture every morsel of audit information possible

- How much audit information should we capture to derive meaning? Again, if the system is of a certain government organization, auditing might  be dictated by an existing Federal standard. Some organizations, for example, require auditing data be kept on tape for five years.
- Where do we capture our audit information? Standard practice is to allocate a specific directory for storage, which usually contains a file(s) readable only by powerful users. While this is good practice, it is my opinion that additional logging facilities should be enabled: (1) a console purely dedicated to capturing all system information (this is an effective means to maintain accountability should the system/audit logs become victim to ‘rm -rf’’ or zap.c); and (2) a remote roach motel—information is fed to a centralized blackbox but cannot be modified thereafter.
- What do we do with the audit information once captured? Ensure an organized (documented, adhered to) audit review program is created and maintained. This includes: (1) designation of an official security auditor who gathers and reviews audit logs for specific failures; (2) generation of an escalation chain and XXXXXXXXXXXXXXXX; and (3) propagation of the audit program to the organization as a whole, which not only addresses liability issues (for example, when an employee tries to sue the organization for invasion of privacy by monitoring system activity) but also serves as a deterrent.

**Problem 2: Failure to backup laptops and workstations.**

Many organizations have coveted relationships with manufacturers of enterprise backup software. And many of these organizations have systemized incremental and full backups scheduled for their production machines, finance systems, file servers, and so forth. However, few organizations install backup clients onto laptops or workstations. They expect users to manually transfer critical data onto network file servers or periodically burn CD-ROMs. Indeed this is a far less expensive solution than buying 10,000 user licenses of enterprise backup software. However, having a month-old CD-ROM of your hard drive is pretty worthless when the 100 slide proposal PowerPoint you created yesterday gets cooked spilling Yuengling on the keyboard.

**Solution 2: Install backup clients.**

At the bare minimum, have sizeable, designated, and *secure* (see Problem 5 below) disk space available where users can move critical information at designated intervals. If such an option is exercised, ensure employees are *aware* the space exists for their use (it is truly remarkable how many companies afford their employees drive space without employee cognizance). The ideal solution, as stated above, is to install backup clients which systematically migrate important data to a backup server. If such an option is exercised, ensure employees are *aware* of how to use the software (it is truly remarkable how many employees have their browser cookies files backed up but not their mail spool, documents directories, and so forth).

**Problem 3: Expecting users to update virus signatures.**

Unless your organization practices the art of Naziesque micromanagement, expecting users to update virus signatures is like expecting a fourth grader to do his/her homework under his/her own volition. A few diligent ones will actually do it, but the rest have to be coerced on a daily basis.

Failure to update virus signatures pretty much negates the purpose of antivirus software. Symantec Antivirus Research Center (SARC) has a five hour turnaround time upon submission of a simple virus[4]. For extraordinarily complex and mutating viruses, the turnaround time is approximately 24 hours. Therefore, with automated signature updates, one's machine should be virtually bulletproof from the latest viral attacks.

**Solution 3: Configure antivirus software for automatic signature download.**

Ensure antivirus software is configured to pull the latest virus signatures from either (1) the vendor signature database; or (2) an enterprise signature database every 24 hours.

**Problem 4: Failure to have an efficient patch management system.**

The first concern with patch management is obvious. Despite hundreds of mailing lists, subscription-based patch services, circulated emails, hundreds of security books, telephone calls, and software configuration boards, thousands of machines all over the world are running outdated software. The consequences of this have been demonstrated several times over, particularly by the gang of worms that terrorized the Internet during the year 2000. As Ed Skoudis states in his article *Infosec's Worst Nightmares*, "Code Red taught us…the preventable spread of the worm underscored the importance of keeping up with system patches: rapidly identifying their release, testing them on quality assurance systems and moving them into production at a controlled but rapid pace".

Part of the problem is the 'the machines will be patched when something goes wrong' attitude. Part of the problem is lack of systematic control over patches—a process to obtain, test, and deploy patches does not exist. And part of the problem is financial allocation. After purchasing servers and software, the information technology (IT) administration has little cash remaining for patch management. And patch management is not cheap.

The second concern is less obvious. It concerns organizations that have existing patch management programs. Many of these programs are extraordinarily inefficient. They often require extensive research, testing, retesting, further research, deployment, and regression testing of a single patch. It is this type of bottleneck that prevents timely release of service packs and patches. It is this type of bottleneck that increases exposure to attacks.

**Solution 4: Create a comprehensive patch management program.**

While it is extraordinarily difficult to see the return on investment arising from allocation of funds to patch management, cost-benefit analysis (from a contingency planning perspective) should be far more visible. This means designating a person (or perhaps a portion of a person's time, depending on size of the organization) into obtaining, testing, and installing patches. This means purchasing equipment where "vanilla bundles" or "gold loads" can be regression tested for post-installation anomalies. This also means analyzing all systems and creating specific procedures which address responsibilities, activities, and documentation arising from the patch management process. And this also means establishing strong relationships with vendors for operational support (in fact, most vendors already have patch management segments in place).

**Problem 5: Creating shared drives/repositories without access controls.**

As a remedy to the situation described in the "Problem 2" section of this paper, organizations purchase file servers with colossal physical capacity. Drivespace is then established so that employees can transfer and store critical files, files for collaboration, or MP3s for intraorganizational distribution. The allocated drivespace is generally a Windows 2000 or Novell share and is divided into subdirectories or folders, perhaps by functional team or user.

These network drives/repositories are a good thing. They are a cheap alternative to installing enterprise backup clients on organization-wide workstations and laptops. They are a great place to dump data in temporary situations (technical support personnel typically bundle and transfer important files to the shared drive while user machines are upgraded). And they are wonderful for collaboration, as they eliminate the incessant emails that arise when small groups work on one document. However, these shared drives raise two concerns.

The first concern is that they are very difficult to monitor. Unlike a mainframe with mandatory access controls, there is very little (other than common sense) to prevent a user from transferring a document with sensitive material. This occurs *frequently.* Users transfer information under the auspices that data on the wire passes securely to these shares and is adequately protected upon arrival. Items like network diagrams with Internet Protocol (IP) addresses. Items like vulnerability scanning results. And dozens of other examples that cannot be mentioned for obvious reasons.

The second concern is that access controls are seldom placed and enforced on any of the folders or directories created on the network drive. Therefore, users are generally free to roam in and out of other folders, transferring files as they please. Some might argue their system/network administrators are diligent and ensure that appropriate permissions are placed. Maybe. In my experience, these drives become network data dumps.

**Solution 5: Back to the basics: discretionary access controls.**

Because this problem is compromised of a number of smaller issues, there are a number of smaller solutions:

- Mandate your technical support personnel *not* migrate user files to shared space. If they insist on such a practice, mandate they use discretionary access controls to prevent unauthorized access to that data. Alternately, designate a specific portion of fileservers where only technical support personnel can migrate files.
- Ensure folders and subfolders generated on fileservers have read/write permissions automatically granted to the owner/creator only.
- Scan fileservers routinely for stale subfolders. Backup these items and move them. Likewise, scan fileservers for subfolders with unrelated content (for example, MP3s and movies).
- Teach employees how to establish and distribute group permissions.

**Problem 6: Failing to identify unauthorized servers or sub networks.**

According to a Yankee Group survey, "penetration of broadband for the average online household grew 57 percent from 2001 to 2002"[5]. Consequently, one might conclude that employees would migrate network services (Web, FTP, streaming audio, file sharing) to their home networks. However, erecting a personal server at home would choke the average broadband connection. And who could pass up dropping an MP3 File Transfer Protocol (FTP) server on an OC-12[1]? Unauthorized personal servers, particularly on networks with poorly configured firewalls, are a considerable and not uncommon security threat. A cursory scan of Mitre's Common Vulnerabilities and Exposures database[6] reveals dozens of buffer overflow and denial of service vulnerabilities associated with popular personal FTP, mail, and Web servers. And Microsoft's Internet Information Server has not had the best of records.

But the difficulties associated with unauthorized servers are not confined to the personal realm. Groups within organizations build servers for purposes such as collaboration, testing, and configuration management. They install server

[1] An OC-12 has a maximum data rate of 622.08 Mbps.

software on vanilla operating systems and ignore patch management due to the small user population and infrequency of the system's use. They move the server to a small room, obtain an IP address through dynamic host connection protocol (DHCP), and reboot the machine as necessary. It is through these types of machines that security breaches occur.

Unauthorized wireless networks represent the most common and, unfortunately, worst threat to security. With the decreasing costs of wireless access points (WAPs) and the arrival of new laptops with built in 802.11 support, users are installing their own wireless networks so that they can communicate while shoveling down Krispy Kreme doughnuts in the cafeteria. They are difficult to detect, sit behind the firewall, have transmission ranges exceeding one mile, and generally arrive out-of-box with minimal security features enabled. However, few organizations have taken the initiative to physically eliminate them. Furthermore, few organizations have taken the initiative to add unauthorized wireless networks to their organizational computing policy.

**Solution 6: Scan your own assets.**

Though this solution might seem a bit obvious, it can still be enlightening: during non-critical hours, perform vulnerability scanning (using perhaps nessus) and mapping (using perhaps nmap) of your entire network(s)[2]. This will pinpoint a number of problems, particularly:

- unrecognized network components
- network components in dire need of patching (see Problem 4);
- unrecognized address blocks
- network components that might be missing;
- any number of misconfigurations, from network address translation (NAT) difficulties to firewall mishaps.

Also, extend this vulnerability scanning to the airwaves. Install a wireless network sniffer (using perhaps kismet) and scan for unknown or unauthorized networks. Upon identification of the networks, see if components can be mapped or scanned. See if components can be identified.

**Problem 7: Firewalls, firewalls, firewalls.**

A "firewall", in the most primitive language available, is defined as "a thing that allows my stuff to go out of the network but prevents the bad guy's stuff from coming in". In many settings this might be true, but firewalls still continue to raise a number of concerns.

The first concern is configuration. Many employees within organizations believe that if their environment is firewalled, their information will be completely secure.

[2] For those of you with Class A networks, choose a segment as a sample (perhaps a critical development environment, systems which handle sensitive data, or your Quake® servers).

They fail to consider the fact that hackers, having grown bored with coding buffer overflows for Solaris, have moved onto bigger challenges. Challenges such as network intrusion detection systems (NIDS). As an example, in 1998, most vendors in the industry assumed hackers would not have the prowess to defeat their products. However, as detailed in Tom Ptacek and Tim Newsham's *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*, a paper which broke every network intrusion detection system NIDS on the market, such assumptions were quickly eradicated.

They fail to consider the fact that many default firewall rule sets have statements like "block from any to any tcp allow from any to any tcp" on the last line, defeating the purpose of any rules previously established. They also fail to consider that their firewall might be mythological. Firewalls are not cheap, and many organizations choose to employ them in areas of true sensitivity (financial systems, laboratories). So that "well, I'm behind a firewall at work" statement might not be necessarily true.

The second concern, like patch management, is financial allocation. Organizations are willing to give infinitesimal funding to firewalls and related products. However, funds allocated to internal hardware such as intrusion detection systems, load balancers, and other important production elements are quickly slashed. As Simson Garfinkel states in *Technology Review*, "…and by focusing on defending the perimeter, rather than on defending information assets within an organization, firewalls foster lax internal security practices that magnify the damage that insiders can inflict".

**Solution 7: Do your firewall homework.**

Ensure that firewalls are configured correctly: this means *testing* in addition to inspecting rulesets. Ensure firewalls are protecting important assets, including likely key points of entry. Use demilitarized zones (DMZs) as much as possible—if a series of unimportant components do not need internal protection, migrate them outside the network.

**Problem 8: Training inadequacy.**

Training is a touchy subject. Organizations are afraid of training their employees too much because the employees might leave for greener pastures: an average UNIX administrator who becomes a SANS GIAC Certified Intrusion Analyst (GCIA) can expect a 12% certification bonus[7]. So suddenly the idea of leaving the organization with a 12% bonus (plus expecting a raise in salary upon arrival at a new organization) becomes somewhat appealing. At the same time, organizations are afraid of training their employees because they believe it is more important to spend funds on budget items that have a tangible byproduct. These mindsets must change, as demonstrated by the following concerns.

The first concern is obvious. Employees are not receiving enough training to perform their daily tasks adequately. Solaris system administrators should know how to define auditing events and how to capture them. Windows NT system administrators should know the importance and ramifications of enabling "syskey" on a given system. Network administrators should know the importance of disabling telnet access to routers and restricting access to Secure Shell (ssh) or serial connections at the console. I could go on for hours with a list of the erroneous things I have witnessed or heard in conversations or on teleconferences.

The second concern is similar but less obvious. Employees are not receiving enough training to perform security tasks adequately. Where would the Solaris system administrator begin to look if a hacker kept penetrating a system despite commenting all services in /etc/inetd.conf? What would the Windows NT system administrator do if a strange service was discovered in the process list? What would the network administrator do in the event of a distributed denial of service attack?

**Solution 8: Spend the money on training.**

There are dozens of training courses which do not directly result in a form of certification. A number of them are offered by the SANS Institute, perhaps the most respected source of training for security and system administration professionals. Not only will personnel be more competent, but they will also be more happy.

**Problem 9: Accountability is minimal.**

Although I personally disagree with the notion that "the worst threat to security remains on the inside", I am still a strong advocate of accountability. At the current time, the accountability structure I have seen within most organizations raises some concerns and leaves much to be desired.

The first concern is separation of duties. Many organizations have a single employee administer, backup, audit, and upgrade sensitive systems without oversight. Part of the reason is financial—with the current state of our economy and the fast paced dynamics of many industries, spreading duties of one system over multiple people is difficult. But the main reason is ignorance—organizations do not understand *why* system logs should be evaluated by an independent party. Organizations do not understand *why* code should be tested and migrated into production environments by individuals other than those who created it.

The second concern is auditing. Few organizations have established a systemized program to gather, evaluate, and store audit logs[3]. And for those

[3] For our purposes, audit logs include system logs, event logs, or any other logs which correlate a user with an activity and a time.

organizations that have, few know how to evaluate the data for inconsistencies and suspicious activity.

Hacking is running rampant, this is a certainty (and even the most high profile of organizations are not exempt[8]). With the passing of each generation comes the promulgation of more powerful technology, more sophisticated tools, and more complex attacks. Being able to replay the attacks is not only important for the "who did it?" element of forensics but also for the "how did they do it?" element.

### Solution 9: Accountability reigns king.

A combination of training (as mentioned in Solution 8) and increased accountability (as mentioned in Solution 1) should be a sufficient solution to this problem.

### Problem 10: Contingency planning for intellectual property emergencies.

Some contingency plans have well established procedures which address how to shut down certain systems in the event of unauthorized activity, who to call thereafter, what resources are necessary for restoration, and how to prevent the activity from reoccurring. However, despite the words "contingency plan" implying something about the future, few are very forward looking. Here are a some concerns.

One concern is related to public persona. In the event of serious unauthorized activity, few contingency plans address how to manage public relations. In the event of a hacker disrupting a serious portion of supply chain operations, how will the organization handle relationships with the vendors and partners? How will the organization respond to public inquiry should it be a prominent organization? How will it address concerns about system stability if under a distributed denial of services (DDoS) attack[9]?

Another concern is related to law. In the event of serious unauthorized activity and possible theft/espionage, how quickly will the organization react? Does the organization have in place processes to prevent auditing information from being destroyed in the event of a serious problem? Does the organization have a competent, technical lawyer that understands security issues (one that could tell you the difference between a buffer overflow and a race condition)? Are there plans in place should a new or existing competitor swiftly brings to market a piece of technology that violates patents and other trade protections?

### Solution 10: Consider all dimensions of contingency planning.

Take the time to itemize assets, especially those of the intangible nature. September 11th of the year 2001 demonstrated to the world that *anything* is possible. Therefore, contingency planning must be considered from the most obscure perspectives. Scenarios must be devised that incorporate more elements than the usual nature disaster/human threats. Horizontal consultation within the company must be considered: talks with the marketing departments, legal departments, facilities personnel, etc.

## Conclusion

Despite billions of dollars being poured into information security, organizations are still failing to adequately protect their most important resources. Careful evaluation of the ten latter-mentioned elements will serve as an excellent foundation upon which to build a satisfactorily secure operating environment.

## References

[1] Pike, Rob. *Systems Software Research Is Irrelevant.* February 21, 2000
http://www.cs.bell-labs.com/who/rob/utah2000.pdf
[2] Everson, Zach. *Business Dictionary.*
http://www.zacheverson.com/Business_dictionary.htm
[3] Darby, Mullany. *IT Security Market to Double.* October 29, 2002
http://www.crmdaily.com/perl/story/19809.html
[4] Symantec Antivirus Research Center (SARC)
http://www.symantec.com/avcenter
[5] The Yankee Group, *Technologically Advanced Family Survey of the Digital Home Office*. 2002
http://www.yankeegroup.com/public/products/research_note.jsp?ID=9117
[6] Mitre Common Vulnerabilities and Exposures
http://cve.mitre.org
[7] Foote, David. Information Security. *Security Still Pays.* August 2002
http://www.infosecuritymag.com/2002/aug/securitymarket.shtml
[8] Attrition.org. High Profile Defacement Summary.
http://www.attrition.org/security/commentary/hp-01.html
[9] National Infrastructure Protection Center. Major Investigations – Mafiaboy
http://www.nipc.gov/investigations/mafiaboy.htm