Retain control of Security (even in the wake of an IT Outsource)

Leslie Martinez

Retain control of Security (even in the wake of an IT Outsource)

Leslie Martinez, MSc, CISSP

SANS GSEC Practical Assignment Version 1.4b, Option 2

February 2003

Abstract

Outsourcing Information Technology (IT) was once thought to be an exception; now it is considered the norm. Many enterprises would rather move away from the expensive and complex tasks of IT systems management to focus on aspects of the business they are expected to be good at – manage the core business. Enterprises expect that the contracted IT specialist company would

- have more experience of dealing with the ever-changing IT industry;
- have the ability to specify more apt solutions for the Enterprise, especially as IT products and platforms becomes more and more complex; and
- do it more efficiently and cheaper than if the Enterprise did it themselves.

Many business-critical applications operate on IT systems that are outsourced, and the security of these systems is often paramount to the successful running of the Enterprise. How can the Enterprise evaluate the security posture of outsourced IT? In this paper I attempt to deal with the real issue of 'How can the Enterprise retain control of the security of its business-critical information systems whilst it is in the hands of a third party?' The paper discusses actual problems encountered and two real solutions that were deployed. It gives examples of the tools used, policies that were implemented and so on. More importantly, the paper serves as a methodology for dealing with any outsource where security is of concern. The paper is intentionally non-technical and is expected to transcend technical solutions. It is directed to

the security-responsible or security-aware decision making officers in any Enterprise that is considering outsource of IT systems.

Contents

1 Introduction

With so much dependency on Information Technology (IT), many large Enterprises are finding it increasingly difficult to manage their core business, as well as the IT systems that support it. Some believe that by uncoupling the two and outsourcing the IT parts to an Information Technology Specialist Provider (ITSP) will give them the much needed breathing space to concentrate on their main business mission; typically, running their core business toward a profit. However, this is not as simple as it sounds. It may be simple for the physical IT equipment; however it is not that straightforward for the information that it harbours. Furthermore, the security of the information may be paramount to the core business itself, and hence both parties often share the responsibility of the security of the information via contracts and Service Level Agreements (SLAs)

This paper discusses a methodology that allows a few, well-appointed security resources to have a degree of autonomy over a large population of outsourced IT servers and the System Administrators (SysAdmins) that maintain them.

2 Setting the scene

I was contracted by a small Audit and Risk (A&R) department of a much larger and fairly well established telecommunications company. For the purpose of this paper I will refer to the company as TELco. I was the resident Security expert hired to complete a program of Unix Security Improvement throughout TELco. At the time, little did I know that:

a. I had to rejuvenate and complete a failing project (which was still in the early part of a pilot phase), then extend it to an Enterprise-wide rollout of some 500+ Unix servers.

b. This project was catalysed by an IT outsourcing deal with International Business Machines (IBM) [CA9809] that, if not managed carefully, would cause TELco to lose control of the security of its critical business systems.

In all honesty, TELco started the process on the right footing. A Joint Verification team (JV), formed of members from both companies, went through the list of TELco's assets and decided what was in scope for being outsourced and what was not. Part of JV was to forge agreements and SLAs on security policies that were to be enforced on systems managed by IBM. Measuring SLAs against policy is a very difficult task, so too was keeping eyes on the SysAdmins' root access to the TELco systems.

2.1 Measuring Policy Compliance

A&R had recently developed some very sound system-level security policies but did not have the time or resources to audit the IT infrastructure against these policies adequately and efficiently. This became even more critical due to the pending outsource, as policy compliance will be one of the ways to measure against the Outsourcing Service Level Agreements [TWSLA]. To gain more insight of the security dynamics of the company, A&R had already decided on Symantec Enterprise Security Manager™ (ESM) implementation [ESM55] for policy compliance. ESM was one of the few well tested and established policy compliance products in the marketplace. Nowadays, there are many more products to assist with Policy Compliance including Bind View's Enterprise Security Solution [BVESS], BMC's Patrol Enterprise Manager [BMCPEM] and IBM's Tivoli [TIVOLI].

2.2 The Root Account Audit issue

It is fairly typical that during these large outsourcing deals that transfer of staff is necessary. In this particular outsource it was negotiated that over 1000 IT specialists from TELco were to be transferred to IBM. This would include the various groups of Unix SysAdmins (and yes, these SysAdmins held the keys to the kingdom – i.e. the superuser (root) passwords to each of the outsourced systems). This was of major concern because without being able to monitor or audit root accesses, TELco could easily loose total control of the security of these servers, and this point (when I raised it with the A&R management) had a devastating impact on the entire outsource process.

A viable commercial solution to provide the solution that I was after did not exist (and I am still unable to find one to date). I did find a recent article that seems close [HC0102]. It is obvious that there is a need for this in todays market, given the fact tha similar solutions have been developed by other SANS candidates [RS0108][HM0201].

3 ESM™ Implementation

This section recalls the series of events that took place during the implementation of ESM at TELco. I was the security specialist for Unix with the single task of completing this implementation.

3.1 Identifying servers

My first task involved the identification of the servers on the network. Using the ISS Internet Scanner [ISS62] with the OS identification module enabled, I was able to scan almost all of TELco's internal networks (several times at different times of the day/week) so as to ensure as many machines as possible were identified. These scans were reconnaissance scans only, hardly intrusive – just enough to find out what was on the networks with some idea as to where they ma be located.

The task of identifying SysAdmins and the responsible owners of systems and applications was next. This meant e-mailing, telephoning and interviewing hundreds of SysAdmins and System Owners to find out what services the particular servers provided for the business. A database (or rather a spreadsheet) was created containing all the information gathered including IP addresses, SysAdmin, System Owner, Applications on Server, contact details, etc. This list was verified against the JV's lists (taken from the TELco asset inventory) and it was found that there were over 500 Unix servers to be outsourced; each being considered as either critical or important to the business of TELco.

3.2 Completing the ESM™ pilot

The immediate requirement was to use a solution where a few resources (namely me) could check and control the Operating System (OS) policy compliance of many (in this case over 500+) Unix servers. I mentioned that an ESM pilot scheme was initiated at TELco. My next task was to complete this pilot and gather useful metrics for an Enterprise wide rollout. The choice of ESM was simple. If configured and implemented properly, ESM can empower a small, centralised group of security analyst to check policy compliance over a vast number of servers [OM0106]. It was perfect for this particular outsourcing situation.

A typical ESM pilot should consist of the following phases; each phase with a set of defined tasks[1]. This is summarised as follows:

[1]My predecessor on this project substituted the Design and Implementation Planning phase for 'shoot from the hip' option. My Design and Planning documentation was actually done after the Pilot Project, as proper planning is needed for a successful Enterprise-Wide rollout.

a. The Design and Implementation Planning - due to the flexible nature of the ESM client/server model, options were explored in order to provide a viable configuration for implementation of the solution. This phase included the following:
   - Design and propose options (with advantages and disadvantages of each option) that fit the requirements of the Enterprise.
   - Select the best option, and mitigating controls against the disadvantages
   - Develop a project plan and roadmap with realistic timescales and resource requirements for the implementation – to include pilot, testing, acceptance criteria, etc.

b. The Pilot Program - this is essential to ensure that the solution is fit for purpose. It allows SysAdmins and management to become familiar with ESM and its reporting functionality, as well as the ability to fine-tune the product closer to the business requirements. Tasks for the pilot included the following:
   - Install the ESM Manager Software
   - Install Agents and register to ESM managers
   - Install ESM console and utilities, and connect to managers
   - Test communications amongst agents, Manager and consoles
   - Analyse written policies to create a test baseline policy on the ESM manager
   - Test and run Baseline policies, produce reports
   - Analyse reports and determine vulnerabilities in pilot environment
   - Use the automated correction utilities, and check effect on agents.
   - Document all of the above in readiness for Enterprise rollout

A group of 20 Unix servers were selected for my pilot scheme. I re-configured two ESM Security managers – one on the Windows NT platform and the other on the Sun Solaris platform. The choice of OS for the ESM managers were to attempt to leverage the capabilities of the ESM manager functionality on different hosts, and also TELco was expected to monitor Microsoft servers in the future. The servers selected for ESM agents were administered by a single group of about 20 SysAdmins, who were easily contactable and offered 24/7 SysAdmin support. They were also pro-ESM and quite keen to remove the security responsibilities from their overall tasks, as they contemplated their migration to IBM.

I then proceeded to develop the ESM security Policies. The first was a Baseline Policy - which included the absolute minimum of the security checks. The Second was the Full Compliance check – in line with the written TELco policy. The third was a higher Security Compliance check – for servers that were especially sensitive.

In addition to the 3 main policies, I developed 4 other Specialist policies that ran on different schedules to the main policies. These I called

a. The Agent Scan, used only once for new agents to ensure that they are set up properly and responding to the ESM manager

b. User Accounts, runs weekly to check for inactive accounts, failed logins and expired passwords. It also looks for changed in account privileges, invalid shells, home directories, etc.

c. Password Brute Force, runs every 6 weeks (as it is CPU intensive) and uses a simple dictionary attack in an attempt to identify user passwords.

d. System Files, runs every fortnight with a series of tests against the file systems including a CRC checksum, changes in device files, identify world-writable files and files with the 'sticky bit' set.

The pilot scheme gave some invaluable insight into the operational security of TELco. The A&R team did not perform any 'hands on' tasks on servers, even though they were responsible for its security. All changes to a particular server is expected to be performed by the SysAdmin(s) responsible for that server. Based on metrics from the pilot scheme, Appendix A includes a calculation of the amount of effort that was required to bring a typical Unix server to policy compliance. This estimation proved to be true in the Company wide rollout.

3.3 Pre-Deployment tasks

Once the pilot was deployed it was allowed to run for a 2-month period for tuning and refining. It also allowed the IT staff to settle in with their new employer (Yes – the outsource had started).

In the meantime, I prepared a 'road show' of presentations and meetings with the new IBM Unix SysAdmins and the TELco system owners for the remaining servers in the Enterprise. It was clear from an early stage that A&R did not have the necessary resource to carry this project. Support and 'buy-in' from the Unix SysAdmins was therefore paramount to the overall success of this particular project.

It was just as important to keep track of the Unix server community. More ISS scans were performed on the networks and my host database (spreadsheet) was updated. To keep track of the server community a new procedure was issued and ratified by the JV team. The procedure stated that changes in the server community must be reported to A&R. Any additions, removals or changes (permanent or otherwise) of Unix servers were then filtered to me so that it can be reflected in ESM.

Lessons learnt from the pilot project were incorporated into the project plan for the next stage – the Enterprise-Wide rollout. Reports from ESM had to be produced and managed and could easily become a time consuming task. In preparation for this, A&R employed one full time employee (FTE) to work with me. His role was to foster and grow the relationships with the SysAdmins, and to take over the general ESM administrative and security analysis tasks on an on-going basis; his role to be continued long after the implementation was completed.

3.4 Deployment tasks

The tasks required for the successful deployment can be summarised as follows:

a. Create a full document set for the Implementation

b. Create ESM Agent Rollout Pack, FTP server (if necessary), and notes for agent deployment

c. Build and configure more ESM managers

d. Configure various policies & domains on all managers

e. Install all agents and bring to compliance (approx. ½ day per server.)

f. Create Exception reports (for agents that cannot fully comply)

A full document set was put together for the ESM implementation including ESM Security Analyst Guide; Administrator Guide; Agent Installation, Backup guides; Consol (GUI) installation and user Guide; Security Update guide; Installation Checklists; Project Plan; Overview presentations; Communication flow diagrams; Database of ESM Servers; etc. Some of these documents are available on the Symantec's Web site [ESM55m]; others were created during the course of the project.

ESM Domains were set up based on SysAdmin groupings, geographical locations and a single domain of highly sensitive servers. Servers were grouped into domains and ESM policies were ran against each domain, one domain at a time, so as to be manageable to the SysAdmins and to my growing team. Agent installation packs were created with software and instructions for the SysAdmins. SysAdmins were required to download the installation packs from an FTP server that I built specifically for this purpose. They were then expected to install the agents and ensure that they were registered with the appropriate ESM manager. Full instructions for the process were provided in the packs.

On initial Policy compliance checks on servers, the SysAdmins and System Owners were given reports relating to their servers' compliance. From our pilot we realized that about 75% of the non-compliance issues could be automatically fixed from the ESM GUI. The other 25% had to be done manually or scripted. Scripts developed in the pilot were distributed to the SysAdmins. Our reports include details of the non-compliance, the fix, whether it can be automatically fixed by ESM or not, the justification for the fix, our notes (base on the use of the server), etc. For each non-compliance, the SysAdmin responded either for it to be fixed automatically by ESM (where this was available) or that he will handle the task manually, and a date for completion.

If a SysAdmin had an issue with a particular change, he was required to request a dispensation with full justification why a deviation from the compliance check was necessary. The A&R team (i.e. me) reviewed each dispensation and, in conjunction with the system owner and SysAdmin, a decision was made on how long the dispensation would last before the host must comply with the company policies. ESM caters for this by allowing messages to be suppressed (i.e. the check is still done, however the compliance message is not shown on reports, until it expires or the suppression removed).

3.5 Ongoing and Maintenance tasks

As with any project of this magnitude, there are other tasks that must be fulfilled to maintain the systems long after the implementation project is complete. To facilitate this the following tasks were completed:

a. Differing schedules were set up for the different policy types that would run on different domains.

b. Set up and test a backup and restore strategy for the ESM managers;

c. Set up a Security Update process so that managers and agents can be updated on a regular basis with updates from Symantec;

d. Extend the 'server population' policy so that A&R became more involved and informed on any new IT projects, etc. This allowed A&R the ability of providing security advice early in life cycle of new projects.

The FTE person involved with me during the implementation phases was more involved with the trends and analysis of the reports produced by ESM, and chasing up SysAdmins to complete tasks by the required deadlines. As the ESM implementation matured, his role would change to incorporate the following:

e. Working with the various SysAdmins to investigate changes in compliance

f. Monitoring general compliance of all agents

g. Perform ESM Upgrades and Updates

h. Ensure ESM policies continue to match TELco written policies

3.6 Conclusion

The entire process took 12 months to complete. The Pilot phase was completed within the first 3 months. The Enterprise-Wide rollout took the rest of a year to complete. Despite the hard work and the time it took, the end product was that a relatively small security team could effectively manage and control the security of the environment, and ensure SLA targets are met.

To summarise briefly some of the necessities for any such project to be a success:

a. The SysAdmins and systems owners must 'buy-in' to the solution – it is their involvement that will make it a success.

b. Pilot the solution, fine-tune the requirements, and the reporting at this stage. The task is much more difficult at the enterprise wide rollout phase

c. Phase the rollout and even the compliance checks on each server by using progressively tighter policies.

d. Consider a 'roadshow' or similar to raise the awareness and get buy-in from management and SysAdmins prior to the main rollout.

e. Find ways to keep track of the server population, especially new servers entering the network.

f. Keep proper documentation – it will help you in the long haul and keep you out of trouble, especially when dealing with a third party supplier.

g. Have escalation procedures and a process for dispensation – Not every server may be able to comply with the policy straight away.

4 Root Account Audit

One of the things that is inherent when an IT outsource occurs is passing control of the 'god' accounts of the systems to the IT Services Specialist. The Unix example of a 'god' account is the 'root' Account.

Historically, the A&R department would be provided with a written list of root passwords from various administrators. These would be locked in the vault in a secure room for safekeeping. As addressed by Reeves [RS0108], keeping passwords in a locked safe is fraught with problems. I found it difficult to imagine that this 'trusting' process would continue when the systems were outsourced to IBM.

With in-house systems, there are no real SLAs, and it did not really matter who typed the wrong command, as long as the services can recover quickly. Incident response and forensics are easier to perform with less 'red tape' when dealing with your own staff. It gets a little more difficult when you need to involve a third party. The administrator would normally be local to the server, so allowing policy settings such as 'no root logins except at the console' was acceptable.

4.1 The Problems with Conventional Methods

The main issue was the level of detail offered by the Unix audit function for privileged root accounts. For example, if an incident occurred that was similar to the identity theft case as reported by ITWorld [IT0211], our forensic analysis (probably taken from ESM reports) would show that 'root' was logged in and had access to the records in question. But who exactly is 'root'? Would you take all of IBM to court, or just the 80+ SysAdmins who were on duty at the time? As a matter of fact, the courts may find that TELco may be liable since they cannot positively identify the perpetrator.

To solve this we needed to provide audit information on the use of the root accounts with sufficient granularity to identify an individual.

The obvious choices would be utilise Unix commands 'sudo' and/or 'su' [UXMAN] both of which allows the user to log in as themselves then escalate there privileges when required. A fairly comprehensive paper on SUDO for remote access via SSH is available at the SANS reading room [FL0106] and there are many Internet references for SU, my favourites being the ITWorld ones [IT0206]. The beauty of these solutions is that accesses can be logged on the system (or preferably to a syslog facility) with sufficient detail to trace back to a single person.

Using the JV team, since they existed as the brokers for issues between TELco and IBM, I proposed different solutions to the SysAdmins. My solutions all used SU and/or Sudo, and a total ban from direct Root logins (except via the console). Each proposal was 'shot down in flames' by the SysAdmins. The arguments were as follows:

a. Direct root access was required under certain plausible scenarios, e.g. a full filesystem or when a user's home directory is on an unreachable NFS share.

b. Remote access will be required due to the geographic spread of the servers and the 24/7 support that was implemented. Using policy controls such as 'limiting root to just console login' is not an option.

c. There are 80+ SysAdmins. To get the granularity I needed, each must have an account on each box, since they can no longer share accounts.

d. It is considered bad practice to have more than 24 accounts on a Unix server, especially when many may not in use. It can become increasingly difficult after that to manage user accounts and detect an impostor [GS0112].

e. Password management becomes an issue. Leavers, Joiners and role changes all cause problems with this solution, especially with the churn rate of Unix SysAdmins.

f. Weak Passwords will always be an issue but will be magnified in this solution. SysAdmin will tend to use the same password on all 500+ servers. Some will be using weak passwords, which makes all systems vulnerable to password attacks.

4.2 A solution fit for purpose?

I needed another solution; one that was quick and easy to implement, acceptable by the SysAdmins yet provide us with the degree of granularity that was required for due diligence.

I enquired about their current process. If a SysAdmin needed to access a particular server, he would follow these steps:

a. Log on to a share on the network

b. Open a particular file on the share that contained server names and passwords.

c. A simple search for his server displays the required password(s)

d. Notes the password and logs off the share

e. Logon to the Unix server using the generic SysAdmin or Root username and the noted password.

This was a very simple process and it worked for them. From time to time, the SysAdmins would update the list of Usernames/Passwords and keep a

backup copy off-line. This was the same process used at TELco and it was the same list that was passed to the A&R team for safekeeping. The only difference is that it was a single list now, as IBM placed all the SysAdmin into a single group.

Eureka! It was simple! With just a few tweaks to their existing process, I was able to satisfy all concerned. My new process would use a database (preferably encrypted, on a server in a physically secure facility), with secured remote access facilities such as the Cable & Wireless SecureDial® service [CW0207] or using other secure remote access facilities such as a VPN or even SSH as described by SysAdmin™ Journal [DF9801].

In its simplest form the new process is as follows:

a. Secure login to Database using your own personal account

b. Enter an auditable entry including reason for needing the password, etc

c. Query Server name and password type required (root or SysAdmin), database returns password.

d. Note the password – this will be used to log onto the server.

e. Change the password on the database for that server. Keep a note of this second password as well. Log out of the database.

f. Login to the Unix system with the required Username and the noted password from d.

g. Change the password on the server to the new password that you entered on the database in e.

Ok, so I have added 2 new steps to the process, namely e. and g., and I have changed b. drastically. Now I needed to test that this process will work in the real world. Using VB (and some skills I developed while reverse engineering the Melissa Macro Virus) I created a script that did the following:

a. Captured the SysAdmin's login identity (i.e. Username) and date/time of access.

b. Prompted for host name or IP address of the server that the SysAdmin wanted to access, a reason for access and the username of the Unix account that is associated with the password..

c. Prompted for the SysAdmin to enter a new password and confirm new password. This password will be the new password on the Unix host.

d. Stored the hostname and/or IP address of the Unix Server, the SysAdmin's identity, username of the Unix account, date/time and reason for access into a text based log file.

e. Displayed the hostname and/or IP address of the Unix Server, the Unix account name, current server password and the new server password – with brief instruction with what the user should do next i.e. 'note both

passwords and account details, log on to the server immediately and change the password, etc.

The above was sufficient for our testing purposes as it tested the process without too much technical solutions in the way. If the script failed the users can easily fall back to their original mode of work.

4.3 Dissecting the Process

On analysis I found that the simple change in process did not affect the SysAdmins to any great extent. From the logs produced by the simple VB script, manually correlated with the output from our ESM reports, I could tell which SysAdmin logged into which server and when they did it. Mission accomplished. However, we found that there were many other pros and cons to this solution as follows:

a. The password was attached to whichever user who made the password change. Only that user is expected to know the new password for a particular server at any point in time, and can login as many times to the server without changing the password. If a second SysAdmin needed to access the same server, he would need to go through the process and change the password. Both users could be logged in, but if the first user logged out he would need to go through the process again. Even in this scenario I knew which 2 users were logged in.

b. On busy servers, the password changed many times per day as different SysAdmins logged on. This could eventually cause the use of weak passwords, and creation of large password histories within a short space of time.

c. The failed logging attempts rose quite sharply (as would be expected) on busy servers. In some cases it had to be raised from 3 attempts to 5.

d. There were lots of cases where the new password did not make it from the database to the server. Since the script kept the last 10 passwords used on the database this was not too much of an issue, and it was easy to identify the culprits. Of more concern, was when the passwords in the database did not match the password on the server due to a typo. Implementing 2 or more usernames per server with root privileges solved this issue so that one account could reset the password for the other.

e. Forgotten passwords was never a problem – just run the script and get another.

f. Dealing with SysAdmin churn was easy – once their user account has been disabled, they will not be able to get the new passwords.

g. This process is not expected to increase the number of privileged accounts on the system.

4.4 Further Development required

Unfortunately, this was as far as I was involved with this particular part of the project. A&R were getting the information they required. A 'real' VB development programmer tidied up my scripts and added cryptographic protection to both the password file and the log file. The use of the script became standard procedure as it fulfilled the requirements of TELco and IBM.

For me, if I had the time, I would have developed it into something sexier, probably using some of the web-enabled automatic password generation and update ideas used by Holbrook [HM0201] and PGP encryption used by Reeves [RS0108] with some facility for remote access and strong authentication as with SecureDial [CW0207], SSH [DF9801] and/or VPNs. I would use encrypted databases (instead of flat files) that could be clustered or run in high availability at different geographical points on the network to protect from hardware (single point of) failure and useful in disaster recovery.

5 Summary

In the wake of an IT outsource and the many things that need to happen for it to be successful, the checklist will always have a single line item for security. To put the tick in the box the enterprise need to have some assurances that the IT Services Company is keeping your information safe. The only sure way of doing this is by having a way to check the security status of the IT equipment, and the ability to capture audit and accountability information whenever those systems are accessed.

Appendix A: Estimated Times for Fixing Security issues Identified by ESM

This document tries to estimate the time (in man hours) it takes to bring an average Unix machine in line with TELco and ESM security policies. The report makes the following assumption:

a. The Unix programmer/analyst is capable of carrying out system administration tasks and script writing on one or more Unix o/s flavour.

b. All security (and other) patches are applied to the o/s as part of the normal daily Unix Administration and will not be considered in these timings.

c. Estimated times are based only on ESM Ratings 4 (Critical), 3 (Major) and 2 (General) security issues.

d. The System Analyst is familiar with the machine, applications and machine/application usage and capabilities.

Ratings

Table showing Breakdown of ESM Ratings

ESM Rating	Avg. No. of messages per machine	Avg. No. of different message types	Percentage ESM can fix remotely	Percentage split per message type	Notes
4 Critical	30	3	95	95/3/2	'Special Device file with other access' is the largest percentage and can be fixed by ESM
3 Major	11	2	90	50/50	'Setuid/Setgid to owner' and 'Guessed User password' are the main security risks here.
2 General	140	8	60	40/40/5/5/4/3/2/1	File permissions, file attributes and U-mask are the culprits here.

Timings

Agent Installation takes about 5 minutes and should not take longer than 10 minutes. The instructions are clear and simple. **Total time (max): 10 minutes.**

ESM Remote Correction Utility can fix (on average) more than 75% of the security issues raised by the ESM 2,3 and 4 ratings. The timing here depend on how long it takes the System Analyst to decide whether or not the ESM recommended fix is acceptable. If they are acceptable then it is a matter of a mouse-click to fix them all. If they are not then either they are fixed manually or have it suppressed by ESM with a dispensation served. **Total time (max): 60 minutes.**

Manual Correction is needed to fix up to 25% of the security issues. This can be time consuming, but I have noted that many items are repeated many times over on one machine. Script writing can greatly reduce the time it takes to fix repeated items and moreso if scripts can be written so that they are re-used on other machines.
If scripts are utilised effectively then on an average machine this should take no longer than 60 minutes (which includes items that should not or cannot be scripted). **Total time (max): 140 minutes.**

Conclusion:
My (generous) estimation indicates that it should take **no longer than 3½ hrs** to resolve the security issues highlighted by ESM on an average machine by a competent Unix systems analyst.

Appendix B: References

[BMCPEM] BMC Patrol® Enterprise Manager http://www.bmc.com/products/proddocview/0,2832,19052_19429_23191_7266,00.html (February 3, 2003)

[BVESS] Bind View's Enterprise Security Solution. URL: http://www.bindview.com/solutions/Security/ (February 3, 2002)

[CA9809] Craig,Andrew, "IBM Wins Giant Telecom Outsourcing Deal" TechWeb News. September 2, 1998. URL: http://www.techweb.com/wire/story/TWB19980902S0004 (February 2, 2003)

[CW0207] Cable and Wireless SecureDial Service Description. July 2002. URL: http://www.cw-reference.com/consult/s-manual/sm-one/market/sdialsd.pdf (February 3 2003)

[DF9801] De La Vega, Francisco M. "Secure Remote Sessions (SSH)". SysAdmin Journal. January 1998. URL: http://www.samag.com/documents/s=1195/sam9801d/9801d.htm (February 3, 2003)

[ESM55] Symantec Enterprise Security Manager™ 5.5 Key Features page. (previously known as Axent Enterprise Security Manager).URL: http://enterprisesecurity.symantec.com/products/products.cfm?productid=45&EID=0 (February 2, 2003)

[ESM55m] Symantec Enterprise Security Manager™ 5.5 Manuals page.URL: http://www.symantec.com/techsupp/enterprise/products/sym_esm/sym_esm_55/manuals.html (February 2, 2003)

[FL0106] Forbes, Liam, "Sudo and SSH: A Scheme for Controlling Administrator Privileges and System Account Access" SANS InfoSec Reading Room, June 11, 2001. URL: http://www.sans.org/rr/authentic/sudo.php (February 2, 2003)

[GS0112] Shaffer, George "Hardening OpenBSD Internet Servers" GeodSoft Website Consulting December 15 2001 URL: http://geodsoft.com/howto/harden/OpenBSD/users_files.htm (February 4 2003)

[HC0102] Hinton, Craig "Symark Powerpassword". SC Magazine. Jan 2002. URL: ftp://ftp.symark.com/unix/pr/sc_powerpassword_review.pdf (February 3 2003)

[HM0201] Holbrook, Mark "The web password page" SANS InfoSec Reading Room. January 26, 2002. URL: http://www.sans.org/rr/authentic/web.php (February 3 2003)

[ISS621] Internet Security Systems Internet Scanner 6.2.1 Documentation. URL:http://www.iss.net/support/documentation/docs.php?product=7 (February 2 2003)

[IT0206] Henry-Stocker, Sandra. Various(to June 2002) ITworld.com http://www.itworld.com/nl/unix_sys_adm/06262002/ http://www.itworld.com/nl/unix_sys_adm/01232002/ http://www.itworld.com/nl/unix_sys_adm/01092002/ (February 3 2003)

[IT0211] "Identity Theft raises questions about Security" Itworld.com November 27 2002. URL: http://www.itworld.com/Sec/2052/021127identitytheft/ (February 3 2003)

[OM0106] O'Neill Michael, "UNIX System Security in a Large Enterprise Environment – Axent ESM" SANS InfoSec Reading Room. June 22, 2001. URL: http://www.sans.org/rr/tools/axent_esm.php (February 2 2003)

[RS0108] Reeves, Shelby, "Secure Password Storage" SANS InfoSec Reading Room, August 14, 2001. URL: http://rr.sans.org/casestudies/storage.php (February 3 2003)

[TIVOLI] IBM's Tivoli Security Management Solutions. URL: http://www-3.ibm.com/software/tivoli/solutions/security/ (February 3 2003)

[TWSLA] "How to Evaluate SLA Compliance" The Web Host Industry Review. URL: http://thewhir.com/reseller/articles/esla.cfm (February 3 2003)

[UXMAN] SU Man Pages. URL: http://unixhelp.ed.ac.uk/CGI/man-cgi?su Sudo Man Pages. URL: http://hegel.ittc.ukans.edu/topics/linux/man-pages/man8/sudo.8.html