Global Information Assurance Certification Paper

Detection and Recovery from a Major Security Breach

Richard Hanschu

Completed in partial fulfillment of
GIAC Securing Unix Certification (GCUX)

Network Security 2000

I. Executive Summary

Overview: This paper is the result of a comprehensive security audit undertaken of a system named Pearl. This system functioned as the main firewall/gateway for a medium-sized company that was quickly progressing into the B2B service market on the Internet. During the course of this audit, major vulnerabilities were discovered in the system configuration. Further study lead to the discovery of a rootkit installed on the machine. At this point the focus was changed from auditing the system to damage control and restoration of a higher level of security. Instead of providing mere recommendations for improving security, this paper documents the actual steps taken in discovering and recovering from this security breach over the last month.

Security Vulnerabilities Discovered:

- Poor/non-existent formal security policies.
- Dangerous water based fire suppression and flaws in physical security
- Insufficient backup and disaster recovery procedures.
- Multiple unneeded and insecure services offering immediate root compromise with scripted attack tools.
- Very poorly written firewall rulesets not blocking critical attack routes.
- Installed rootkit that had trojaned critical binaries such as login and rpcbind. Rootkit included sniffers for aquiring further passwords.

Steps taken to recover from security breach:

- Immediate removal of rootkit files and replacement of known trojaned binaries.
- Restriction of firewall ruleset to limit further exposure
- Installation of tripwire to monitor system for futher suspicious activity until it could be replaced
- Redesign of network layout for improved security
- Building two new hardened firewalls to replace compromised system
- Establishment of backup and disaster recovery plans
- Second audit of newly installed firewalls to insure secure configuration

Recommendations for further improvements:

- Installation of non-water based fire suppression equipment
- Closure of above ceiling access into server room
- Installation of IDS system to monitor traffic on protected DMZ
- Continued upkeep of security patches on high risk systems

II. Initial Audit

Description of system: The system audited (pearl) was a SUN Microsystems Ultra1 machine running Solaris 2.6. This machine was running Checkpoint Firewall-1 (version 4.0) and functioned as the main security gateway for XYZ.com. It had been setup by a previous administrator, and not actively maintained with security in mind. Given its importance to the security of the entire network, it was chosen for a through audit.

Physical Security: The machine was located in a locked server room. Access to the room was controlled by keycard operated electronic locks mounted in a secure solid-core door. Access to the room was limited to a short list of administrators and trusted developers. Inspection under the raised floor revealed no access methods. Inspection of the dropped ceiling revealed that there is the possibility of access in certain locations where the wall does not extend entirely from floor to ceiling. The room was further protected with an alarm system with motion and sound sensors activated nightly. Two cameras recorded all activities 24 hours using tapes changed daily. Fire suppression was a typical heat-activated water system common in office buildings.

Security Policy: No formal security policy was in place concerning the operation and maintenance of this system. The previous administrator often left the machine logged in as root with the Firewall-1 ruleset configuration GUI on the screen. Only the administrator had an account on the machine, but no policy existed concerning password quality or rotation for either the user or root accounts.

Backup and Disaster recovery plans: Occasional backups of the system had been made in the past to tape. The backups were not frequent or organized, nor were the tapes stored off-site. There was no disaster recovery plan in place concerning this machine and there was no spares readily available for dealing with possible hardware failures.

OS Audit: Given that this machine was designed as a security gateway between the Internet and the internal networks, looking at running services was a logical place to start. Appendix A contains a condensed version of inetd.conf from pearl. The security vulnerabilities from these inetd services are almost to numerous to list. A summary of the worst follows in decreasing order of severity:

Service	Vulnerability
ttdbserverd, statd, cmsad	Known root compromises in some versions (CERT CA-98.11, CA-99.05, CA-99.08[1])
rexecd	Allows remote execution without reliable authentication (CVE : CAN-1999-0618[2])
sadmin	Known bug allowing arbitrary commands run remotely (CVE : CVE-1999-0626)

[1] This citation format refers to vulnerabilities detailed at http://www.cert.org
[2] This citation format refers to entries in the Common Vulnerbilities and Exposures dictionary found at http://cve.mitre.org/

rlogin services	Allows access to system utilizing weak .rhosts authentication (CVE : CAN-1999-0651)
Telnet and FTP	Vulnerable to sniffing of passwords in clear text (CVE : CAN-1999-0619)
Misc RPC services:, userd, sprayd, quoted, nlockmgr,	Possible DOS attacks and information leaks about system
echo, chargen, daytime, finger, walld	Possible DOS attacks and information leaks about system

This is a pretty bleak situation. A scan of the running processes reveals even more services running as daemons (Appendix B). Several of these services also have known vulnerabilities:

Service	**Vulnerability**
Sendmail SMI-8.6/SMI-SVR4	Multiple known vulnerabilities leading to root compromise in this very old version. (CVE : CAN-1999-0203). System not set to prevent mail relay.
SNMP	System responds to community names public and private. Possible information leak (CVE : CAN-1999-0517)

Since this machine was configured as a firewall, perhaps none of the ports these services listen to were available due to the firewall configuration. An Nmap scan was run on the machine from a remote system that should have been considered hostile by the firewall. The results are shown in Appendix C. As is apparent, many of these services are still shown as open to the outside world. In particular, rlogin, exec and RPC services are world accessible. At this point an audit of the firewall ruleset became the next logical step.

Audit of software running on system: At this point, several critical software packages were audited to see if known vulnerable versions were running and to determine why the firewall was allowing critical traffic to pass.

Sendmail: The sendmail version running was a SUNOS package version 8.6. Multiple vulnerabilities are known in this version. As configured, this system did not complain when issued the command :MAIL FROM: |testing This probably means that it is possible to send mail that will be bounced to a program, which allows anyone to execute arbitrary commands. In addition mail relay was not turned off.

Tooltalk Database (ttdbserverd), Calender manager (cmsd) and Statd (statd): These were determined to be unpatched versions vulnerable to immediate root compromise. (see CERT CA-98.11, CA-99.05, CA-99.08).

Checkpoint Firewall-1: The firewall ruleset was thoroughly examined. Since this isn't a firewall audit exercise, I will briefly summarize the findings. In general the firewall ruleset seems to have been developed by adding in services that "made things work" instead of limiting services to only those needed. A "NOISE" group of services was found with ANY:ANY access that included TCP111 (portmapper), TCP135-139 (Netbios services) as well as high TCP and UDP ports (RPC services). This allowed many types of dangerous traffic to pass the firewall. There were many services allowed that I have never even encountered, leading me to suspect that they were added "just in case". In addition, the there were several outstanding security patches to the firewall software that hadn't been applied (4.0 SP1)

Forensic audit of file system: Very quickly it had become apparent that this system had major exploitable vulnerabilities. At this point, a CD-R was created with a known "clean" toolkit from a fresh install of Solaris 2.6. This toolkit included copies of ls, netstat, ifconfig, lsof, top, find, du, chown, chgrp, chmod, sh, tcsh, tar and gzip[3]. The cdrom was mounted and the shell path changed so that these binaries would be utilized instead of those on the system. A ps –ef was done with the "native" system ps command and the "clean" one and the results run through diff. This revealed an immediate discrepancy, a program called "./sys222" that wasn't revealed by the "native" ps. Lsof showed this file residing in /var/spool/.recent (which also wasn't revealed by "native" ls). The system had clearly been rooted. Examination of the files in this directory and searches on http://packetstorm.securify.com revealed that this was a variant of the sun2.rootkit. The program revealed by the ps check was a IRC proxy server that was running on the machine (but actually blocked by the firewall). This root kit replaces ls, ps, find, netstat, rpcbind, and tcpd with trojaned copies that either hide the presence of the kit or provide backdoor access to the system. In this particular case, the startup script was modified so that it didn't replace or kill rpcbind (the copy found on packetstorm does). It also contains packetsniffers and a smurf attack utility. Neither of these was found running on the system.

Initial Damage Control: At this point, the security audit was stopped (there was little point in finding more possible vulnerabilities when the machine was clearly compromised and the reality of situation required action). This machine was absolutely essential for the functioning of the business and so couldn't be shutdown without a clear plan of action. The files in the root kit were chmod 000 and moved to a different location. Trojaned binaries were replaced with known good copies from the CD-R. Tripwire v1.3.1 was installed on the machine to monitor for further suspicious activity. Firewall-1 rulesets were drastically reduced to disallow almost all traffic to and from the machine from the external Internet.

III. Steps Taken to Recover from Security Breach

At this point, the companies' main security gateway was revealed to have been compromised with a rootkit containing a packet sniffer that could have monitored all of the traffic entering and leaving the network as well as a large percentage of the traffic on

[3] Unix Forensics, John Green. Sans Institute

the internal network. All root passwords were immediately changed on all machines. A quick audit was conducted on all machines with the clean CD-R toolkit looking for signs of this or similar rootkits. No other machines have been found to be compromised. Plans were made to totally replace the compromised machine with a clean system as soon as possible as well as to redesign the network structure in order to provide a more secure interface with the Internet.

Network Redesign: The original network was created with a central hub and spoke network design (Fig. 1). During replacement of the security gateway, the network was redesigned into a more efficient two firewall design with a protected DMZ[4] (Fig.2,)

Figure 1. Original Network Design

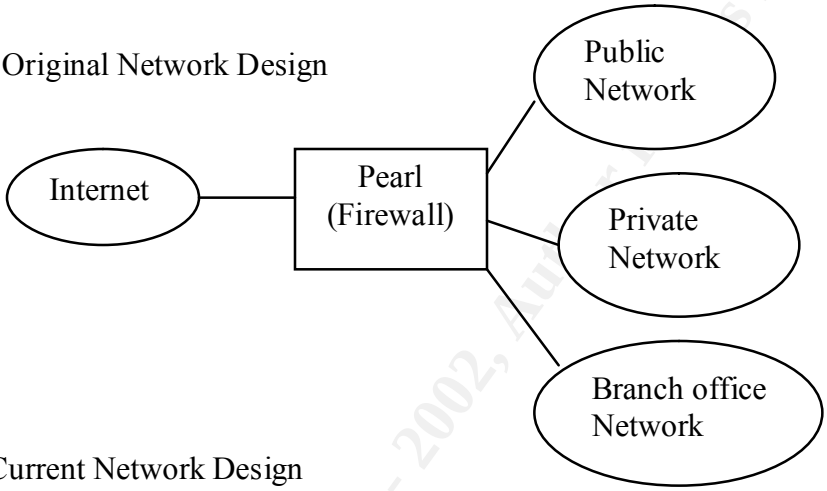
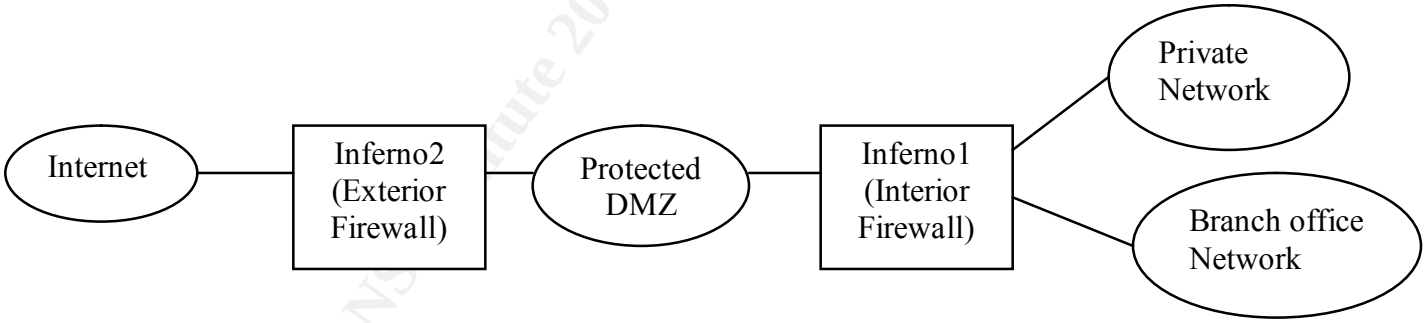


Figure 2. Current Network Design



This perimeter defense design is superior because it allowed different firewalls with different responsibilities. The exterior firewall (Inferno2) protects the Internet servers on the DMZ from general access from the Internet as well as defense in-depth for the second interior firewall (Inferno1) which protects the private network with a more stringent ruleset as well as acting as a NAT gateway to the Internet

Firewall System Hardware and Physical Security: Both firewall machines were identically configured with the same hardware and software. The systems ran on AMD K5/400 machines with 128 MB RAM and Intel Pro100 NIC cards. The machines were

[4] Firewalls 101: Advanced Perimeter Protection and Defense In-Depth, Lance Spitzner. Sans Institute

located in the same server room described above. After initial software install was completed, all removable media drives were removed from the system and the case was locked.

OS and software configuration: The Mandrake 7.0 distribution of Linux was installed. During the install script, the highest security level was chosen. By default, this does not active inetd or any other services. All required services must be explicitly enabled. MD5 shadow passwords were enabled. A very minimal custom server install was chosen with only core required packages. In order to facilitate creation of later tools, gcc and it's requirements was installed. Later after software configuration was completed, gcc was removed. A strong root password was used and one user account with a strong password was created. Passwords were checked with the pam_cracklib module[5]. Lilo and BIOS were not protected with a password due to the requirement that the machine reboot unattended. Immediately after OS installation, the following security patches were installed:

dump-0.4b18-1mdk.i586.rpm	pam-0.72-7mdk.i586.rpm
fdutils-5.3-11mdk.i586.rpm	pam-devel-0.72-7mdk.i586.rpm
glibc-2.1.3-14mdk.i586.rpm	pam-doc-0.72-7mdk.i586.rpm
glibc-devel-2.1.3-14mdk.i586.rpm	perl-5.00503-11mdk.i586.rpm
glibc-profile-2.1.3-14mdk.i586.rpm	man-1.5g-15mdk.i586.rpm

(note: kernel patches not installed due to later re-compile of kernel from fresh sources)

This brought the system to a "fully patched" state according to Mandrake Software. Next ssh2 v 2.2.0 was acquired from http://www.ssh.com, compiled and installed with tcpd support (--with-libwrap; TCP wrappers is installed by default in the Mandrake distribution). Ssh2 was setup to run as a daemon from rc2.d. TCPWrappers was configured via /etc/hosts.allow and /etc/hosts.deny to only accept logins from a known administrative console. Since inetd was disabled, no access was allowed to the system via telnet, rlogin or ftp. The system was configured entirely at the console without network connection until ssh2 was available. The following banner was added to tcpd for ssh2 as adapted from CIAC[6]:

```
**************************************************************************
                NOTICE TO USERS

Use of this system constitutes consent to security monitoring and testing.
All activity is logged with your host name and IP address.


**************************************************************************
```

Since these machines were designed as security gateways, no other services such as SMTP, NFS, DNS or SMB servers were installed.

The next step was to install IPSEC network layer for VPN access. The source code for FreeS/WAN (v1.7) was downloaded from www.freeswan.org. Since Mandrake 7.0 installs the 2.2.14-15mdk kernel, the kernel source for 2.2.16-9mdk was downloaded

[5] http://www.kernel.org/pub/linux/libs/pam
[6] http://ciac.llnl.gov/ciac/bulletins/j-043.shtml

from a Mandrake mirror. The most recent kernel was acquired due to a known kernel bug in versions prior to 2.2.16 allowing root access for local users.[7] The new kernel and FreeS/WAN sources were MD5 checksumed against signatures provided at the source websites. The new kernel was compiled with FreeS/WAN IPSEC enabled and then installed.

Next Postfix was installed as a MTA in order to enable the gateway to send security notices to the administrator. Postfix was acquired from http://www.dynamite.org/postfix/start.html and installed via rpm. It was configured to run in Null client mode where it can only send mail to the domain MTA and does not listen on TCP25[8]. After the installation of Postfix, Psionic Logcheck[9] was acquired and installed. It was configured to send messages to a frequently monitored security email drop box on the administrative station.

After all software was installed, Tripwire[10] (v2.2.1) for Linux was installed. This version encrypts configuration files and reference databases, reducing the need to store these on read-only media. Reference databases were created and a cron job was created to run the tripwire audit once a night and mail the results to the security email dropbox.

Backup and Disaster Recovery: A regular backup system to directly attached tape drives was instituted with tapes stored off-site. A third physical machine of identical configuration is stored on site for either spare parts or total replacement of a damaged system within minutes. The disk images are updated on this backup system whenever a major change is made to the production systems.

Firewall Configuration: The kernel packet filter IPchains was utilized to create the firewall ruleset. I will only briefly describe the techniques and philosophy behind ruleset creation. By default all traffic into and out of to all interfaces was denied. Anti-spoofing rules were created preventing each interface from receiving traffic from networks which should be on "the other side" of the firewall or appear to come from the firewall itself. Only known required services were allowed to pass the incoming filters. Since IPchains is not a stateful firewall, upper return ports could not be totally blocked. Rules were created to allow return traffic (SYN-ACK) only from machines to which connections (SYN) could be made. All other SYN or SYN-ACK traffic was denied. Access to the firewall itself was only allowed via ssh from a secured administrative station on the private network (in addition to TCPwrapper rules which enforce this same policy).

IV. Secondary Audit

Due to extreme time constraints and an intense desire to remove the known compromised firewall, the machines were put into production at this point. After all of the resulting problems were worked out, a second security audit was conducted.

---

[7] http://www.linuxsecurity.com/advisories/advisory_documents/other_advisory-476.html
[8] See http://www.dynamite.org/postfix/faq.html#null_client for Null client configuration setup
[9] http://www.psionic.com/abacus/logcheck/
[10] http://www.tripwire.com

Security Policy: A security policy has been established in reference to these machines. Only a system administrators have user accounts on the machine, and PAM has been configured to check the password on these accounts for strength. Password rotation is enforced via PAM every two months. Access to the root account is only allowed at the console or via su after login via ssh only.

OS audit: An audit of the processes running on the machine reveals none of the services considered vulnerabilities on the previous firewall (Appendix D). No inetd.conf or portmapper super-service exists on this machine, so there can be no vulnerabilities from this source. An self explanatory audit of /etc/passwd is included in Appendix E showing very limited user accounts (and use of MD5 shadowed passwords implied). Given that no user access is allowed to the machine other than administrators (only one currently), the risk of privilege escalation from a software flaws is greatly reduced.

Nmap was used to scan both of the firewalls from internal and external systems. The Nmap output is not included due to its extreme simplicity. No open ports were detected unless the scan was conducted from the administrative station, and in this case, only tcp/22 (ssh) was open. All other ports registered as filtered. A Nessus scan (including DOS attacks) was also negative.

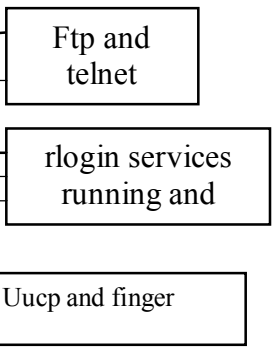
V. Further Recommendations:

At this point, a highly insecure "security gateway" has been totally removed from the system and replaced with a set of secured firewalls. Further recommendations for increasing the security of the system are mostly based on the flaws found in the physical security of the machines. The holes in the plenum above the dropped ceiling should be sealed to prevent bypass of the door locks. This will also be required in order to upgrade the fire suppression system to a Halon style system. If a Halon system is not feasible, at the very least a two-stage water system with a dead man switch should be installed.

In order to maintain the security of the system, future security patches should be obtained where needed and installed. For the security of these machines and those that they protect, and IDS system such as SNORT or SHADOW should be installed on the interior firewall monitoring the protected DMZ.

Appendix A: Inetd.conf on pearl (comments edited out for brevity)

```
ftp     stream tcp    nowait root   /usr/sbin/in.ftpd      in.ftpd
telnet  stream tcp    nowait root   /usr/sbin/in.telnetd   in.telnetd
name    dgram  udp    wait   root   /usr/sbin/in.tnamed    in.tnamed
shell   stream tcp    nowait root   /usr/sbin/in.rshd      in.rshd
login   stream tcp    nowait root   /usr/sbin/in.rlogind   in.rlogind
exec    stream tcp    nowait root   /usr/sbin/in.rexecd    in.rexecd
comsat  dgram  udp    wait   root   /usr/sbin/in.comsat    in.comsat
talk    dgram  udp    wait   root   /usr/sbin/in.talkd     in.talkd
uucp    stream tcp    nowait root   /usr/sbin/in.uucpd     in.uucpd
finger  stream tcp    nowait nobody /usr/sbin/in.fingerd   in.fingerd
```



```
#systat stream  tcp     nowait  root    /usr/bin/ps             ps -ef
#netstat        stream  tcp     nowait  root    /usr/bin/netstat        netstat -f inet
time    stream  tcp     nowait  root    internal
time    dgram   udp     wait    root    internal
echo    stream  tcp     nowait  root    internal
echo    dgram   udp     wait    root    internal
discard stream  tcp     nowait  root    internal
discard dgram   udp     wait    root    internal
daytime stream  tcp     nowait  root    internal
daytime dgram   udp     wait    root    internal
chargen stream  tcp     nowait  root    internal
chargen dgram   udp     wait    root    internal
100232/10       tli     rpc/udp wait root /usr/sbin/sadmind     sadmind
rquotad/1       tli     rpc/datagram_v  wait root /usr/lib/nfs/rquotad  rquotad
rusersd/2-3     tli     rpc/datagram_v,circuit_v        wait root /usr/lib/netsvc/rusers/rpc.rusersd    rpc.rusersd
sprayd/1        tli     rpc/datagram_v  wait root /usr/lib/netsvc/spray/rpc.sprayd       rpc.sprayd
walld/1         tli     rpc/datagram_v  wait root /usr/lib/netsvc/rwall/rpc.rwalld       rpc.rwalld
rstatd/2-4      tli     rpc/datagram_v wait root /usr/lib/netsvc/rstat/rpc.rstatd rpc.rstatd
#rexd/1         tli     rpc/tcp wait root /usr/sbin/rpc.rexd     rpc.rexd
#ufsd/1 tli     rpc/*   wait    root    /usr/lib/fs/ufs/ufsd    ufsd -p
100221/1        tli     rpc/tcp wait root /usr/openwin/bin/kcms_server  kcms_server
fs              stream  tcp     wait nobody /usr/openwin/lib/fs.auto    fs
100235/1 tli rpc/tcp wait root /usr/lib/fs/cachefs/cachefsd cachefsd
kerbd/4         tli     rpc/ticlts      wait    root    /usr/sbin/kerbd  kerbd
printer         stream  tcp     nowait  root    /usr/lib/print/in.lpd   in.lpd
dtspc stream tcp nowait root /usr/dt/bin/dtspcd /usr/dt/bin/dtspcd
xaudio   stream tcp   wait root /usr/openwin/bin/Xaserver Xaserver -noauth -inetd
100068/2-5 dgram rpc/udp wait root /usr/dt/bin/rpc.cmsd rpc.cmsd
100083/1 tli rpc/tcp wait root /usr/dt/bin/rpc.ttdbserverd /usr/dt/bin/rpc.ttdbserverd
```

Multiple unnecessary services running. Possible DOS attacks vulnerabilities

sadmind

Unnecessary rpc services… possible information leaks and vulnerabilities

Cmsd and statd

Tooltalk DB server

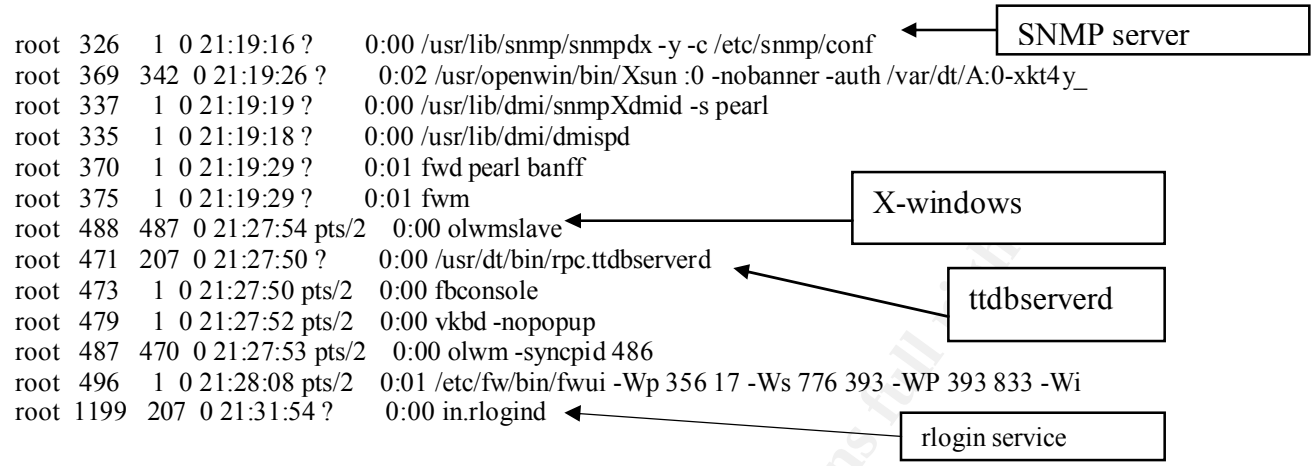
Appendix B: ps –ef output from pearl *(condensed*… some routine and X related processes removed)

```
  UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0 21:17:24 ?       0:00 sched
    root     1     0  0 21:17:27 ?       0:00 /etc/init -
    root   106     1  0 21:18:46 ?       0:00 /etc/fw.boot/fwboot bootd
    root   393     1  0 21:20:01 pts/0    0:00 /usr/lib/saf/ttymon -g -h -p pearl console login:  -T sun -d
/dev/console -l co
    root   296     1  0 21:19:14 ?       0:00 /usr/sbin/vold
    root   176     1  0 21:18:52 ?       0:00 /usr/sbin/rpcbind
    root   207     1  1 21:19:05 ?       0:01 /usr/sbin/inetd -s
    root   212     1  0 21:19:05 ?       0:00 /usr/lib/nfs/statd
    root   214     1  0 21:19:05 ?       0:00 /usr/lib/nfs/lockd
    root   263     1  0 21:19:13 ?       0:00 /usr/lib/lpsched
    root   233     1  0 21:19:06 ?       0:01 /usr/sbin/syslogd
    root   364   326  0 21:19:22 ?        0:00 mibiisa -p 32796
    root   284     1  0 21:19:14 ?       0:00 /usr/lib/sendmail -bd -q1h
    root   253     1  0 21:19:12 ?       0:00 /usr/sbin/nscd
    root   377   342  0 21:19:30 ?        0:00 /usr/dt/bin/dtlogin -daemon
    root   342     1  0 21:19:19 ?       0:00 /usr/dt/bin/dtlogin -daemon
```

Checkpoint Firewall

Rpcbind and inetd super services

A mail and print servers on a firewall?



Appendix C: Nmap output of scan of pearl from external machine

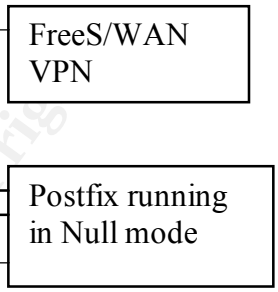
```
Starting nmap V. 2.30BETA17 by fyodor@insecure.org ( www.insecure.org/nmap/ )
  Interesting ports on  (****):
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
25/tcp     open        smtp
37/tcp     open        time
79/tcp     open        finger
111/tcp    open        sunrpc
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
6000/tcp   open        X11
32771/tcp  open        sometimes-rpc5
32772/tcp  open        sometimes-rpc7
32773/tcp  open        sometimes-rpc9
32774/tcp  open        sometimes-rpc11
32775/tcp  open        sometimes-rpc13
32776/tcp  open        sometimes-rpc15
32777/tcp  open        sometimes-rpc17
32778/tcp  open        sometimes-rpc19
32780/tcp  open        sometimes-rpc23

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=38275 (Worthy challenge)
Remote OS guesses: Solaris 2.6 - 2.7, Solaris 7
```

Appendix D: Process list from inferno

```
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Nov05 ?        00:00:02 init
root         2     1  0 Nov05 ?        00:04:36 [kflushd]
root         3     1  0 Nov05 ?        00:09:20 [kupdate]
```

```
root       4    1  0 Nov05 ?       00:00:00 [kpiod]
root       5    1  0 Nov05 ?       00:02:35 [kswapd]
root       6    1  0 Nov05 ?       00:00:00 [mdrecoveryd]
root     422    1  0 Nov05 ?       00:00:00 /usr/local/sbin/sshd2
root     436    1  0 Nov05 ?       00:00:01 crond
root     563    1  0 Nov05 ?       00:00:00 /usr/local/lib/ipsec/pluto --debug-none
root     651    1  0 Nov05 tty1    00:00:00 [login]
root     652    1  0 Nov05 tty2    00:00:00 [mingetty]
root     653    1  0 Nov05 tty3    00:00:00 [mingetty]
root     654    1  0 Nov05 tty4    00:00:00 [mingetty]
root    1842    1  0 Nov09 ?       00:00:00 /usr/lib/postfix/master
postfix  1845  1842  0 Nov09 ?       00:00:00 qmgr -l -t fifo -u
root    18744 18743  0 11:23 pts/0    00:00:00 -bash
postfix 25930  1842  0 17:18 ?       00:00:00 pickup -l -t fifo
```



Appendix E. /etc/password on inferno

```
root:x:0:0:root:/root:/bin/bash
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
postfix:x:101:104:postfix:/var/spool/postfix:
richardh:x:102:104:richardh:/home/richardh
```