Securing a Novell Nterprise Linux Services Server: Step-by-Step

Al Maslowski-Yerges

SANS GIAC-GCUX Practical Assignment 2.0 Option 1
July 1, 2004

Abstract

The purpose of this paper is to create a step-by-step plan for implementing a secure installation of specific components of Novell's Nterprise Linux Services 1.0 (NNLS). This server is needed primarily to extend network services to remote users in a secure and easily used manner without the complexity VPN's of any kind. First I will address the specifications for the server in terms of hardware and software needs. Secondly, I will outline the risks that are likely to be present for this server in this role along with a plan to mitigate those risks as much as possible. Next I will document in detail how the server can be installed and configured to meet that mitigation plan. I will also outline and document some of the ongoing maintenance procedures that need to be employed in order to keep the server secure and functioning well in its role. Lastly, I will document testing procedures that will help to verify that the server is reasonably secure from the risks outlined as well as point out additional measures that might need to be taken to further secure the environment.

Server Specification

1. **The Need –** This server is being put into service in order to satisfy several needs of remote workers. Remote workers need access to networked data and services from any location with the flexibility to connect using any workstation available. The solution must be simple to use with very little to download or configure for the end user. They also need a collaboration platform that will allow them to work effectively with others on their teams regardless of location or installed software. The solution should look the same whether on the road or in the office. They want to be able to share files, a calendar, and have a place for shared discussions/postings. Several key employees have recently lost laptops or had computer crashes resulting in a great deal of lost time and productivity for the company. Some of the data lost may not be anywhere else. Workers need a way to access their data from anywhere (i.e. during a flight) while ensuring that the data is also backed up and safe. Above all, the solution needs to be secure. The company has a great deal of intellectual property that must be protected.

2. **The Problem –** The company has had a VPN solution in place for some time, but that solution has failed to live up to the expectations of all users and has proven difficult to use and hard to support. It has become clear that the solution does not really meet the needs. It is hard to set up and requires a specific client that must be configured by the IT department. Therefore it can't be installed on many of the computers that might be available to remote users. Collaboration is really non-existent and employees are frustrated with trying to FTP files back and forth or send them through the e-mail system with its limits on attachment sizes. There is no consistent solution for backing up remote user data. Many users have been using ZIP disks or CD-RW media to backup important data but versioning and control has become a real problem.

Other users have resorted to sending copies of important documents to themselves in e-mail to external e-mail accounts. Security is very lacking and important data is seeping out of the company daily through the practices remote workers and others have adopted in order to try to meet their data access needs.

3. **The Chosen Solution –** Novell's Nterprise Linux Services 1.0 has been chosen to solve these problems. Here is how it matches up to the needs discussed.
   3.1. Easy to use/portable – For many functions all that is needed is a web browser with java support. If additional functionality is needed a small agent can be automatically downloaded and installed.
   3.2. Collaboration/File sharing – The Virtual Office component which is also web based allows secure easy sharing of files, calendars, and discussions. Since it is a web interface it looks the same no matter where it is accessed from. eGuide offers a web based "Yellow Pages" to find co-workers and speed collaboration.
   3.3. Data availability and backup – Two parts of the suite offer solutions in this area. iFolder allows data to be synchronized in a secure manner between multiple workstations and the server keeping the data safe and available. NetStorage is a web based application that makes iFolder as well as other networked data sources available to users from anywhere with a web browser.
   3.4. Security – All of the solutions use SSL encryption so the data and user information are safe. It also improves security by reducing the need to "go around" the network by e-mailing important data to external mail accounts or carrying the data on ZIP disks, CD's, or even laptops that can be lost, stolen, or destroyed. Since the authentication is tied into the corporate directory solution, no additional id's or passwords are needed either. Additional authentication mechanisms (such as "smart cards") can also be employed if desired.

4. **The Software Required/Chosen –** In order to support this solution SuSE Enterprise Linux Server version 8 was chosen as the base operating system. It will be patched to the current service pack available, SP3, with additional patches as available from SuSE. Novell Nterprise Linux Services (NNLS) 1.0 will be installed as well. NNLS is composed of both open source packages and Novell controlled packages. The packages we are installing are listed below along with their version number as of this time. Additional packages are available as part of NNLS 1.0 but will not be installed.
   4.1. Apache 2.0.45
   4.2. Java JVM 1.4.1_02
   4.3. Tomcat 4.1.24
   4.4. eDirectory 8.7.3
   4.5. DirXML 1.1a
   4.6. eGuide 2.12

4.7. Samba client 2.2.8a
4.8. iFolder 2.1.2
4.9. NetStorage 3.0
4.10. iManager 2.0.2
4.11. iPrint 5.0
4.12. Virtual Office 1.0.1
4.13. Red Carpet Client 1.4

5. **Other Supporting software –** In order to install and run the services listed above a couple of other pieces of software must be installed and running.
   5.1. Open SLP 1.0.11-1
   5.2. A full list of all Novell supplied software and version is found in the Appendices. This list reflects the version numbers after all available patches have been applied.
6. **Hardware requirements –** In order to run all of these services effectively for the roughly 100 people who will use it regularly, we need a fairly robust server. Base on this here are my recommendations for the hardware.
   - Pentium 4 2 -3 Ghz
   - 1-2 GB of RAM
   - Disk space 72GB+ minimums: /var – 350 MB, /opt – 100 MB, /usr – 310 MB
7. **Network Access – Who and from where?**
   7.1. Gateway server – This server will not need to allow local login to the OS and will not hold user home directories or any directly accessible user data. It will be a gateway into the rest of the network and serve up resources found on the internal network. To accomplish this, the server will be accessible from both the Internet and the internal network. It will be available to all network users.
   7.2. Admin only – Direct login to the server will only be allowed for a few select admin users over SSH to allow for maintenance and monitoring. This SSH access will also be allowed from the Internet for remote administration.

Risk Mitigation Plan

Based on the role of the server, I will now identify what the most likely avenues of attack might be as well as a general plan for protecting against these attacks/vulnerabilities.

1. **Most likely sources of attack –** In order to most effectively plan for the protection of the server, we need to identify the most likely sources of attack so that we can concentrate our efforts on protecting from these attacks.

   1.1. Web Server based attacks – Since all of the main services provided by this server are web based, HTTP/S will need to be open in the firewall. This provides multiple methods of attack from cross site scripting to webdav attacks to Apache or Tomcat buffer overflow vulnerabilities. Apache and Tomcat will have to be carefully protected.

1.2. Password Guessing/Brute Force – The services available to users will be password protected so there will be multiple interfaces available for entering user id's and passwords. This presents an opportunity for an attacker to try to guess user id's and passwords of valid users to try to get access. We must protect this as well as possible without losing functionality.

1.3. Internal Attacks – Internally many more ports will be open to allow for eDirectory replication, connection to other data sources for DirXML, and other purposes. This potentially makes the number of internal avenues of attack much greater. The server will have to be protected from the internal network as well.

1.4. Denial of Service – Denial of service is also a likely style of attack that we should account for. An attacker may just want to overwhelm the system to take it offline and make it unavailable for our users.

2. **Plan to protect –** Based on these avenues of attack, here are some general methods by which I plan to protect against those attacks.

2.1. Physical Security – There is almost no way to ensure that a server is secure without physical security being maintained. This server should be behind an access controlled door that only administrative personnel can access. The BIOS should be protected and the server console should be locked and logins restricted from additional interfaces (i.e. serial interfaces)

2.2. Network/Host/Application based firewalls – Multiple firewalls will be protecting this server from outside access. Only HTTP, HTTPS, and iPrint will be allowed through the firewall to this server. Outbound access from this server will also be limited to only the ports needed for Red Carpet patch management out to Novell's site. All outbound access is very limited by the external firewall in the organization, further protecting the site. This server will be located on a "DMZ" network such that access to the internal network will be filtered by another logical firewall layer. An Apache module called "mod_security" with some application specific IDS/firewall functionality will be configured and implemented.

2.3. Unneeded services shut down – All daemons not necessary to the function of the server will be shut down and disabled to reduce the avenues of attack.

2.4. Patch Management – Patching the software installed is one of the most important tasks. Patching will be centrally managed and conducted on a scheduled, periodic basis to protect against newfound vulnerabilities. Both application and OS related patches will be monitored very closely.

2.5. Logging/Log analysis – This server will maintain extensive logging and will forward its logs to a central syslog server for monitoring and alerting.

2.6. Network/Host IDS – The network already employs network based IDS which will be adjusted to incorporate any changes necessary to effectively watch this machine. Tripwire will be installed on the host to give a method of detecting changes to key files on the system. A method of updating and monitoring Tripwire will be employed. We may want to explore a more full featured host based IDS solution in the future as well.

2.7. Backup routine – Both a "Gold" level backup of the original configuration and daily backups of user data will be maintained to protect the user's data as well as assist in identifying any malicious changes made to the system.

2.8. Vulnerability scans – A periodic vulnerability scan will be conducted on the system to identify any ongoing or newly identified vulnerabilities. This data will be analyzed and a plan of action drawn up and implemented to mitigate the risk from any newfound or ongoing vulnerabilities identified.

2.9. Account Lockouts/Passwords – Because of the multiple login screens available it is important that all user accounts are set to enforce a lockout time period after 3 – 5 unsuccessful login attempts. There is a danger of allowing a denial of service attack if an external attacker can prevent a user from logging in by purposely locking the account but that is better than allowing brute force password guessing. Also all users will be reminded about corporate password policy and good password habits that they have already been instructed in. Once again, we may want to explore the use of client certificates or "smart cards" to offer an additional layer of defense here.

2.10. Kernel Tuning – In order to protect against a number of denial of service attacks, some kernel tuning will be done to change how the server responds to certain types of potentially malicious network connections.

2.11. Stay supportable – An overarching requirement that must be carefully guarded even as we secure the system is that the system must remain "supportable" by Novell. No changes can be made that will void Novell support. If changes are made, they must be easily reversed so that Novell can properly support this installation with us.

2.12. Consider moving DirXML – The "meta-directory" functionality of DirXML makes this a very sensitive server because this server will be "trusted" by other entities to make changes to multiple types of directory data including user id's and passwords. The original design calls for this functionality to be on this server along with the other NNLS components. I suggest moving this functionality to another server to make the configuration less critical in the environment and to reduce the amount of damage that a compromise could do to the rest of the systems. However, for this design I am keeping this requirement intact and will work to offer protection for it.

8. Summary – Risk Mitigation.

Security is always a balance between risk and function. The services that will be supplied by this server are desperately needed by the business and will be used extensively by some of the users in the highest positions in the enterprise.

Given the role of this server the multiple layers of security we have planned are both appropriate and necessary. As a gateway into the network and an important resource for remote staff it should be carefully protected and managed.

Installing and Hardening the Server

Based on the plan outlined above I will now go through the specific steps necessary to install and harden the server and the applications that exist on it. First I address the installations of the operating system itself and the NNLS software. Next, I address the OS and the application hardening steps we must go through.

Operating system installation

The install should be from manufacturers CD media (SuSE Enterprise Server 8) with the normal install routine and default prompts except as noted below. The network cable **MUST** remain unplugged until after patching has taken place and several services have been disabled!

1. **Partitioning as follows** (choose "expert" option, adjust sizes for production)

Partition	Size	Format
/	2GB	EXT3
/usr	2GB	EXT3
swap (in extended partition)	900 MB	swap
/var (in extended partition)	106GB	EXT3

1.1. This partition strategy was chosen for the following reasons:

1.1.1. Fairly large root partition to hold root's home directory for downloads/etc… and also so additional software can be added as necessary. Depending on your sites needs you may want to reduce or enlarge this partition.

1.1.2. The "usr" partition is kept separate so it can be mounted read only (RO) under normal conditions to protect binaries from easy attack.

1.1.3. The "swap" partition should help performance if memory utilization becomes high. This is not as likely in today's server environments but it is still good practice to create a fairly large swap partition.

1.1.4. The "var" mount point is kept on its own partition for several reasons. First, this is where all of the data for iFolder, NetStorage, eDirectory, and Virtual Office is stored. This configuration makes it much easier to cleanly backup and restore this data. Secondly, log files may also stored on this partition (depending on what logs are sent to a syslog server and I suggest all log files are) so the old trick of filing up the file system to take down the server will not work.

2. **Package selection.** – The goal is to try to install as small a system as possible to reduce the possibility of adding new and different attack vectors and to keep the system as stable and fast as possible. As a base to work from, select "minimal graphical system – without KDE". Then add the

following packages/groups:

2.1. KDE – this will be a server but may be easier to manage at the console with a graphical interface. Run level will be set to 3 to prevent Xwindows from running on the server at all times. Other "X" security parameters will be added later.
2.2. C/C++ Compiler & Tools – This would not go on a normal production system! A later step below removes most of the most dangerous parts of these packages but it would be best not to include them at all. In my environment it was necessary to have them so I could compile several drivers and programs and install them. Once compiled, the drivers can be moved from system to system or installed again on similar hardware so the compiler and libraries would not be necessary.
2.3. gettext – necessary for the NNLS installation script to run.
2.4. xntp – needed for time synchronization which is <u>very</u> important to eDirectory health and function.
2.5. YaST2 configuration tools with modules for security check and SuSEfirewall2 management as well as regular management.
2.6. SLES Administration Tools – for server management but only the necessary tools.
2.7. Man pages – always useful in day to day administration and not a great danger.
2.8. sudo – good tool for logging and managing access to the server for administrators.
2.9. Tripwire – to use for system integrity checking in our maintenance section below
2.10. Logsurfer – to assist in managing and monitoring log files if kept on this server (again I strongly suggest putting the log files on a central syslog server instead, but if you must log to this server you probably also need the tools to monitor them.
2.11. Do NOT install – Do NOT install the web server, tomcat or any other modules that will be supplied by the NNLS install. We want only the specific modules and configuration supplied by Novell so that no unknown/unneeded software is installed.

3. **Boot Loader**: Keep the standard configuration here. Later we will password protect Grub from changes and encrypt that password in the configuration files for protection.
4. **Root password** – Choose the "expert" button here and change the encryption to Blowfish or MD5 instead of the default, easy to crack DES encryption. Also be sure to pick a complex password of 8 characters to make it hard to crack.
5. **Additional Users** – Here it is a good idea to add one additional user that will serve as your administrative account on the machine. Later root login will be restricted to the console only and will not be allowed from the network in any manner. You need an account to connect with so that you can "SU" to root or preferably use "SUDO" to manage the server once connected. When adding

the user also choose Blowfish or MD5 encryption and edit the password settings to require an 8 character password, set the password to expire in 180 days, and set the minimum age of the password to be 2 days. These will also be the defaults we will set for any new accounts added to the machine. You probably want to have root's e-mail forwarded to this account as well for ease in monitoring.

6. **Video** – Set the video card and monitor to something appropriate for your hardware. Be careful with this or Xwindows may not start at all upon finishing the configuration.
7. **Network setup** – Edit the configuration to have a static IP address, a valid host name and domain name, gateway, and dns servers. Also, edit the routing tables to add a multicast route! This will be needed later for NNLS to work properly with SLP (Service Location Protocol – a protocol used for discovering and connecting to network based services over TCP/IP) The information should be added as follows:
   - **7.1.1. Destination**: 224.0.0.0
   - **7.1.2. Next Hop**: 0.0.0.0
   - **7.1.3. Mask**: 240.0.0.0
   - **7.1.4. Interface**: eth0

   7.2. Single Interface: This configuration guide assumes that you will have only one interface configured in the server. This server's role as described above does not include any need for routing and should be kept simple in this regard to make the firewall configuration more simple and secure. A second interface should not be added unless it is critical to functioning in your environment. If a second interface is added, IP forwarding should remain off.
8. **No modem, ISDN, or Printer configuration**. Follow the rest of the install as required in your environment.
9. **Reboot and verify that all accounts can successfully log in at the console.**
10. **Patching –** Install SP3 for United Linux from CDROM media (NO NETWORK YET).

    10.1. Mount the cdrom. Then run install.sh from the root of the media. Choose the default install option to update all packages and the kernel.
11. **Disable services –** Use the chkconfig command to list and then disable running services that are not needed. Here is an example from my server.

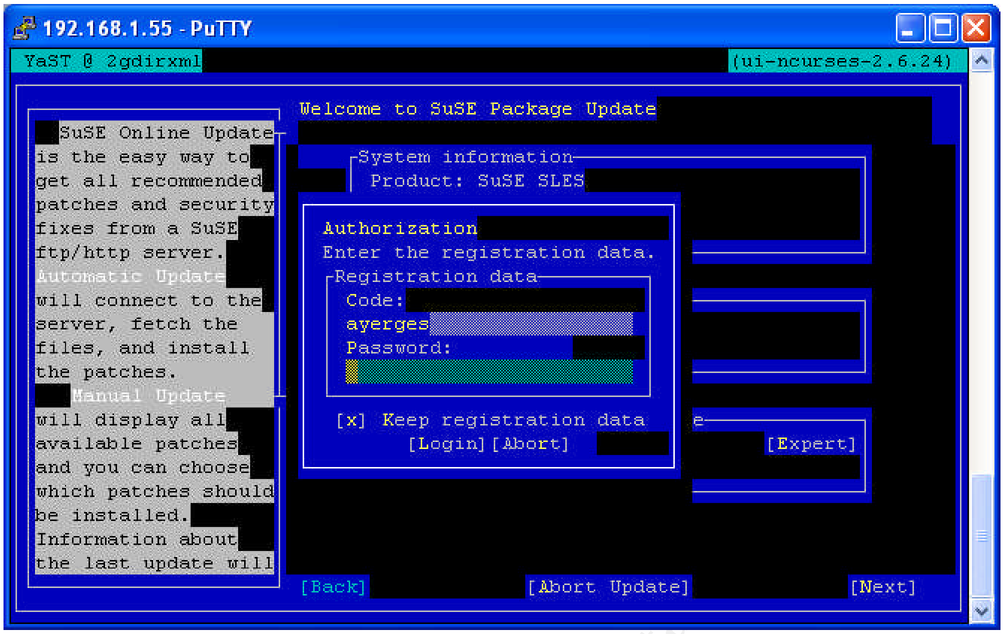
```
2gdirxml:/ # chkconfig |grep on |more
alsasound              on
atd                    on
boot.clock             on
boot.crypto            on
boot.cycle             on
boot.idedma            on
boot.ipconfig          on
boot.isapnp            on
boot.klog              on
boot.ldconfig          on
boot.localfs           on
boot.localnet          on
boot.lvm               on
boot.md                on
```

All services not needed should be turned off. For SuSE, the boot.* services are needed but some that should be turned off are: portmap, hotplug, hwscan, xdm xinetd… These can be turned off with a command such as

“chkconfig portmap off”

For SuSE, the different run levels are automatically taken into account in the chkconfig command. This is convenient but you would have to edit the init.d/rc3.d/ and other run level symbolic links directly if you wanted something to start in run level 5 but not in run level 3 for instance.

12. **Additional patching** – At this point most critical vulnerabilities have been taken care of so the network cable should be plugged in and then additional patches should be applied as available from SuSE. A software support contract is required to get these patches. I suggest using Yast Online Update (YOU) to get these patches and keep the system up to date on all OS related patches. Here is an NCURSES screen showing part of the process after typing the command “you”



```
YaST @ 2gdirxml                                        (ui-ncurses-2.6.24)

SuSE Online Update      Welcome to SuSE Package Update
is the easy way to
get all recommended       System information
patches and security       Product: SuSE SLES
fixes from a SuSE
ftp/http server.          Authorization
Automatic Update          Enter the registration data.
will connect to the        Registration data
server, fetch the            Code:
files, and install           ayerges[redacted]
the patches.                 Password:
 Manual Update               [redacted]
will display all
available patches          [x] Keep registration data
and you can choose              [Login] [Abort]                  [Expert]
which patches should
be installed.
Information about
the last update will
                        [Back]           [Abort Update]            [Next]
```

Installation of NNLS (Novell Nterprise Linux Services)

This installation will not be covered in detail. Refer to the documentation on Novell's web site for more detailed instructions. (http://www.novell.com/documentation/lg/nnls/index.html?page=/documentation/lg/nnls/install/data/front.html#bktitle) Information specific to a successful installation and information related to security settings will be the main focus of this section.

1. **Pre-flight configurations/installations** – Several tasks must first be performed to prepare for the installation.

   1.1. Hosts and DNS - Edit the hosts file to make sure that the proper format is present as follows in this example. Make sure specifically that the localhost.localdomain localhost is as shown without any real hostname in it and that the correct domain name and host name entries are listed following the IP address. The install routine relies on this information being correct.

```
2gdirxml:/ # vi /etc/hosts



127.0.0.1    localhost.localdomain    localhost
192.168.1.55    twodirxml.2gnetworks.com    twodirxml
```

   DNS (both forward and reverse) entries should also be confirmed and tested from the console of this host.

   1.2. Download and install OpenSLP – This is not absolutely necessary but it gives you more options with regard to SLP. If SLP is not already installed, the

NNLS installation routine will install a Novell version called “slpuasa” with somewhat limited functionality in that it can’t act as a “Directory Agent.”

2. **Installation and Component Selection** – Install by running the “install.sh” script at the root of the CDROM. Choose a “custom” installation. The following components will be installed on this server as discussed above, disabling Linux User Management (LUM), Samba, and NetMail:
    2.1.1. Apache.
    2.1.2. JVM
    2.1.3. Tomcat
    2.1.4. eDirectory
    2.1.5. DirXML
    2.1.6. eGuide
    2.1.7. iFolder
    2.1.8. iManager
    2.1.9. iPrint
    2.1.10. Virtual Office
    2.1.11. Red Carpet Client

2.2. Ports – The installation and function of all of the NNLS services depend on specific access to ports on the server. This is very important for successfully installing a functioning server that is still secure. Below is a list of the modules/functions and the ports that they operate on. These services will have to be allowed through the firewall as described in the table. “←” denotes access inbound to the server. “←→” denotes inbound and outbound access to/from the server. “→” denotes outbound access only from the server. We will use this later for firewall configuration.

Service	**Description**	**Port Protocol**	**FW access.**
NCP	Netware Core Protocol	524 TCP/UDP	Internal nets only ←→
Ldap	Light Weight Directory Access	389 TCP/UDP	Internal nets only ←→
Ldaps	LDAP over SSL/TLS	636 TCP/UDP	Internal nets only ←→
iMonitor	Monitoring and management of eDirectory	8008 TCP	Internal nets only ←
iMonitor SSL	Same as above over SSL/TLS	8010 TCP	Internal nets only ←
WebAdmin	For management of Red Carpet	8018 TCP	Internal nets only ←
WebAdmin SSL	WebAdmin over SSL/TLS	8020 TCP	Internal nets only ←
Srvloc	SLP Service Location protocol	427 TCP/UDP	Internal nets only ←→
http	Web access for	80 TCP	Internal and

	Virtual Office and iFolder. All except iFolder redirect to https port.		External nets. iFolder needs http but data and passwords are encrypted by application. ←→
https	Web access for Virtual Office, eGuide, NetStorage, iManager, iPrint	443 TCP	Internal and External nets. Secure Access to office resources through a web browser ←→
IPP	Internet Printing Protocol	631 TCP	Internal nets ←
DirXML	XML file transfers for Meta Directory management	8080, 8090, 8009 TCP (others may need to be added for additional systems	Internal nets only ←→
Samba/CIF/MS networking	Allows connection to internal server shares	135, 137, 138,139, 445 TCP/UDP	Internal nets →
Ntp	Network Time Protocol	123 TCP/UDP	External nets →
Syslog/syslog-conn	Transfer of system log files to secure logging server.	514 UDP, 601 TCP/UDP	Internal nets →
ICMP	Internet Control Message Protocol	ICMP	Limited to specific types. 0,3,11 ← 8,3,11 →

3. **Finish Install/Post install configuration** – Follow the Novell documentation cited above to finish the installation as appropriate for your environment and configure test accounts and connections as they will be when the server is in full production. (Preferably this will all be done in an isolated test network that very closely mimics your full production environment) Note: the LDAP port specified for all LDAP communications should be 636. This is the default, but pay attention to this as you run the installation. LDAP is key to most of the server operations and is used heavily. We want to be sure that it is always accessed over a secure SSL/TLS channel.

   DirXML drivers should be set up, e-mail links in Virtual Office, NetStorage

storage locations, and any other server configurations necessary should be made.

We want to be sure the server functions as desired before additional hardening steps are taken so that we know if our actions have caused the problem or if the software is just misconfigured/non-functional.

4. **Confirm server function** – Full documentation is beyond the scope of this document, but all server functions should be tested to confirm that they work before further work is done to secure the server. Then after each section, this functionality should be tested again. I suggest using a checklist something like the following, edited to fit your environment. You will typically want to replace the IP address in the URL's with the full DNS name of your server as it was installed. Access should be checked from internal IP addresses only until the server has been fully secured and tested. Access from the Internet can only be tested after all security steps have been taken.

Item	URL/Process	Result
Confirm eDirectory health/synchronization	https://192.168.1.55:8010/nds/summary	□ YES □ NO
Can access server home page	https://192.168.1.55	□ YES □ NO
Can access NetStorage	https://192.168.1.55/NetStorage	□ YES □ NO
iFolder works via client and web access	https://192.168.1.55/iFolder/applet/java.htm	□ YES □ NO
iPrint works both as a client and as a print server	https://192.168.1.55/ipp	□ YES □ NO
Can access eGuide and look up data	https://192.168.1.55/eGuide/servlet/eGuide	□ YES □ NO
iManager administration functions properly	https://192.168.1.55/nps/iManager.html	□ YES □ NO
Red Carpet	will connect and check for updates (managed through iManager)	□ YES □ NO
DirXML synchronization still works	Change user details, add, remove, and change a password for a user from both data stores and confirm that it functions and follows your established rules.	□ YES □ NO

Securing the base operating system environment

1. **GRUB Boot Loader** – Password protect the boot loader to prevent editing of the boot environment or passing kernel level commands to the system at boot time. Use the md5crypt command within GRUB to encrypt a password. Then use this hash to edit the menu.lst file and insert the password line as shown below. Be sure **NOT** to use the same password as root or any other user

password on the system. Always test the functionality to be sure the password was typed correctly. You don't want to test it when you are at the console and need to change a boot parameter in a panic!

```
# grub

    GRUB  version 0.93  (640K lower / 3072K upper memory)

 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the possible
   completions of a device/filename. ]

grub> md5crypt

Password: *******
Encrypted: $1$vUYoM$OAxm9NVNUBsCeP1dl50

grub>quit

vi /boot/grub/menu.lst

color white/blue black/light-gray
default 0
timeout 8

password --md5 $1$vUYoM$OAxm9NVNUBsCeP1dl50
title linux
   kernel (hd0,0)/boot/vmlinuz root=/dev/hda1   vga=773
```

1.1. BIOS – If your hardware supports it you should also password protect changes to the BIOS to prevent changing the boot order of the device. In production booting from CD or floppy should be disabled.

2. **Tuning Network Kernel Parameters** – There are a few parameters that can be applied to the kernel through the proc file system to improve protection of the server. Several sources (Linux Security Quick Reference Guide: http://www.tldp.org/REF/ls_quickref/QuickRefCard.pdf, Gentoo Linux Security Guide: http://www.gentoo.org/doc/en/gentoo-security.xml) suggest editing/creating a file /etc/sysctl.conf and inserting several parameters into this file to accomplish this. I found that the file did not exist and that the changes put into it were not applied (SuSE has a different method). After more research I found how SuSE implements this functionality. When SuSE boots, it executes a script /etc/init.d/boot.ipconfig. This script utilizes several variables and settings in /etc/sysconfig/sysctl to write settings to the proc file system to modify the behavior of the network stack. (Suse 9: The Boot process: http://www.openskills.info/view/boxdetail.php?IDbox=944&boxtype=distro) I edited both files to accomplish the same goals as the sysctl.conf parameters

because all settings I wanted were not built into the files.

2.1. Parameters to set - Below are the parameters that I suggest setting:

```
net.ipv4.ip_forward = 0   -- Disables IP forwarding.
net.ipv4.conf.all.accept_source_route = 0 – Disables source routing.
net.ipv4.tcp_syncookies = 1 – TCP syn flood protection parameter.
net.ipv4.tcp_max_syn_backlog = 4096 Additional TCP syn flood protection.
net.ipv4.conf.all.rp_filter = 1 Enables anti-spoofing protection.
net.ipv4.conf.all.send_redirects = 0 Disables the sending of ICMP redirects.
net.ipv4.conf.all.accept_redirects = 0 Disables receipt of ICMP redirects.
net.ipv4.conf.default.accept_redirects = 0 Disables ICMP redirects for newly activated.
```

2.2. /etc/sysconfig/sysctl – Modify this file to add these options along with the default configuration options. The contents of the edited sysctl file I used are in the appendices at the end of this paper.
2.3. /etc/init.d/boot.ipconfig file to activate these settings on boot. This is also in the appendices at the end of this paper.

3. **Warning Banners** – include a warning message for all direct methods of connection to the server.
3.1. /etc/motd – This will display for each login after it is successful. You may have to create this file if it is not there.
3.2. /etc/issue – This file is displayed during interactive login at the console and should also have a warning. In addition to the warning it displays information about the OS version and patch level. This gives away too much information and should be edited out. Don't give away any information about the OS at all. An example of some possible contents for the file is below

```
No Version given.

This system is for authorized use only. All activity may be monitored and/or logged.
```

3.3. /etc/issue.net – This is the same as /etc/issue except that it is what is displayed to users logging in remotely with Telnet, or FTP. You should put the same warning in it as in /etc/issue. It can also be the Banner for SSH connections but the ssh configuration must be edited for this to happen. Below is the portion of the file that must be changed to point the banner at the

/etc/issue.net file.

```
# vi /etc/ssh/sshd_config

…….
# no default banner path
Banner /etc/issue.net
#VerifyReverseMapping no

# override default of no subsystems
```

4. **Additional SSH configuration** – Since we are in the sshd_config file there are some other changes that should be made here. SSH/SCP have been designated as the only remote access protocols allowed to this server but it should be further secured from its defaults. In addition to setting a banner as we did above, it should be restricted to version 2 of the protocol only. SSH version 1 has some inherent weaknesses and so should be avoided. Below is an excerpt of the sshd_config file showing the configuration lines that should be unremarked and set as they are shown. Most settings are fairly self explanatory. No hosts should be automatically trusted through the rhosts types of authentication or even with a machine based certificate as with the RSA variants. Root should not be allowed direct access. For administration, you should connect to the machine as a regular user and then SU to root for additional needed rights.

```
#Port 22
Protocol 2
#ListenAddress 0.0.0.0
#ListenAddress ::
SyslogFacility AUTH
#
#LoginGraceTime 600
PermitRootLogin no
#StrictModes yes
RhostsAuthentication no
# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
PermitEmptyPasswords no
```

5. **Further Securing Remote Login** – In addition to the restrictions we made on SSH, we should also further disable remote interactive login for root in case mistakenly or maliciously telnet or some other method of tty access was

enabled again. To do this we will concentrate on the /etc/securetty file. All lines except the TTY1 should be commented out. This is needed for console access. SSH is running its own daemon and is not affected by these settings.

```
# This file contains the device names of tty lines (one per line,
# without leading /dev/) on which root is allowed to login.
#
tty1
#tty2
#tty3
#tty4
#tty5
#tty6
# for devfs:
#vc/1
#vc/2
#vc/3
#vc/4
#vc/5
#vc/6
```

5.1. Now this file should be protected by executing the following:

5.1.1. "chown root:root /etc/securetty" (should already be owned by root but this is a safety measure)

5.1.2. "chmod 400 /etc/securetty" (this makes it so that only root can read the file and nobody can write to it, even root, until root chmod's the file with more permissions again.

6. **Tighten settings in inittab** - /etc/inittab has several settings in it that should be tightened next. We will disable Ctrl-Alt-Delete from shutting down the server, edit the default run level, protect the server even in Single User mode, and disable extra console login daemons (Ctrl-Alt-Fx) to further protect console access. See the settings made below.

```
# The default runlevel is defined here
id:3:initdefault:

# First script to be executed, if not booting in emergency (-b) mode
si::bootwait:/etc/init.d/boot
# what to do in single-user mode
ls:S:wait:/etc/init.d/rc S
~~:S:wait:/sbin/sulogin

# what to do when CTRL-ALT-DEL is pressed. Comment to disable.
#ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

6.1. The "3" in the id:3:initdefault line designates that the default run level is level 3 which does not load the GUI. The GUI can be loaded as necessary

with the “startx” command but should not remain loaded or load by default on the server.
6.2. The line beginning with ~~:S is the command for what to do in single user mode. (i.e. typing “single” as a boot parameter in grub – which now requires password access anyway). Change the “respawn” command to “wait.” This will prompt for the root password before continuing. This may seem pretty secure but it can be bypassed as well by booting with the command parameter “init=/bin/bash” so it is very important to remember to maintain strong physical security and set that grub password as discussed above.
6.3. The “ca::ctrlaltdel:/sbin/shutdown –r –t4 now” line is the command to execute when Ctrl-Alt-Delete are pressed. This should be commented out as shown to disable this functionality and prevent someone with physical access from shutting down the machine without a valid login.

7. **User Account Security** – Several things should be changed from the defaults with regards to user accounts. I will discuss each and give examples of these changes and commands.
7.1. Password Settings – by default the password settings are quite weak. They never expire, the minimum length is 5, and there is no limit to how quickly the password can be changed again. These 3 things will be changed as shown below to provide some extra security to the user account settings. There is an option to auto-lock expired accounts in the file /etc/default/useradd. It should have a line “INACTIVE= 60” to automatically disable accounts that have expired. The next step is to change the defaults in /etc/login.defs:

```
PASS_MAX_DAYS   180
PASS_MIN_DAYS   2
PASS_MIN_LEN    8
PASS_WARN_AGE   7
```

The next step is to change these settings on existing user accounts. The settings put in the file above only apply to users created after the settings have been saved. In order to change the settings for existing users we will use the “chage” command in the following awk script. Non-system accounts start at 500 and above so this command will set these values on only those users. Root and other system accounts are not affected.

```
# awk –F: ‘$3 >= 500 { system (“chage –M 180 –m 2 “ $1) }‘ /etc/passwd
Aging information changed.
```

7.2. Change default home directory umask - Within this same login.defs file there is a line to set the default umask setting for newly created user’s home directories. The default unmasked settings for all files created are 666 and for directories are 777. The default umask is set to 022 which means that all files created by that user will have permissions of 644 set on them and directories created by that user will have permissions of 755. This means that the files and directories are world readable. When applied to new user home directories each new user's directory is world readable. This is not the behavior we want so change the umask to 077 so that new user home directories created will have permissions of 700 to protect them. Also, change the permissions on the two current home directories (root, <adminuser>) with:

```
chmod 700 /root
chmod 700 /home/<adminuser>
```

7.3. Change default operating umask – There are some other places where the default umask should be edited so that by default users and root do not create world readable files and directories. These are set in the system profile as well as (for most distributions) in shell "resource" files.

7.3.1. System profile – first we will change the default by adding the file /etc/profile.local and then we edit it and add in a line for umask 077

```
#touch /etc/profile.local
#vi /etc/profile.local

# /etc/profile.local for SuSE Linux
#The user file-creation mask changed here to restrict files created
#to more secure setting. Since profile.local is read AFTER
# "profile" This should take precedence.
umask 077
```

7.3.2. Shell files – Unlike RedHat and others, SuSE does not typically put umask settings into user .bashrc .cshrc or .profile files so the system profile applies to all users unless a different setting is put into these files. Also /etc/profile will get overwritten when the system is patched or updated so the proper place to put the umask setting is in a file /etc/profile.local. The setting in /etc/profile.local overrides the setting in /etc/profile because it is read later in the boot process. (Unofficial SuSE FAQ. *SuSE 7.3 Bash Initialization*: http://susefaq.sourceforge.net/articles/bash.html)

7.4. Purging Unnecessary Accounts – by default some unnecessary accounts are added to the system. These can be purged if you are sure none of them are needed. However, in my testing this can break components of NNLS. I suggest purging only the following users: games, news, and uucp. I suggest only removing the groups: games and uucp. Since this could be very hard to recover from, I suggest making a copy of the passwd, shadow, group files before making any changes and then running through your testing checklist before removing those files.

7.4.1. copy the files – Use the following command to make backups.

```
# for file in /etc/passwd /etc/shadow /etc/group ; do /bin/cp -p $file
$file.orig ; \done
```

7.4.2. remove the accounts – Use the following command.

```
# for user in games news uucp ; do /usr/sbin/userdel $user ; done
```

7.4.3. remove the groups – This command removes the extra groups.

```
# for group in games uucp ; do /usr/sbin/groupdel $group ; done
```

7.4.4. Reboot and run through the application checklist – At this point you can recover fairly easily by copying the *.orig files created above back to the original files.

7.4.5. Check files for integrity – Next if all is well, run “pwck” and “grpck” to check the passwd and group files are functioning correctly.

7.4.6. Assign orphaned files – Run the following commands to assign any orphaned files to the root user and/or root group.

7.4.6.1. # /usr/bin/find / -nouser –exec /bin/chown root {} \;

7.4.6.2. # /usr/bin/find / -nogroup -exec /bin/chgrp root {} \;

7.4.7. Remove *.orig files – Remove the files created as a backup above but only remove them after testing all applications.

7.5. Lock remaining service accounts – Use the following command to change the user environment for the listed service accounts to a shell of /dev/null. A shell of /dev/null is preferred so that a shell can’t be spawned by replacing /bin/false. This prevents attackers from using service accounts which can’t be removed to connect to the server and spawn an interactive shell.

```
# for user in bin daemon ftp lp mail named nobody ; do usermod -L -s /dev/null $user ; done
#
```

7.6. Set limits on system resources – Users should be limited in the number of resources they can use so that they can’t cause a denial of service either purposefully or by accident. To do this edit the /etc/security/limits.conf file as shown below.

```
#<domain>      <type>  <item>         <value>
#

*              hard    core           0
*              hard    fsize          102400
*              hard    nproc           150
```

These stop the creation of core files, limit file sizes to 100MB each, and concurrent processes for any user to 150.

8. **Xwindows – GUI protections** – Although X-windows is not loading by default on the server, this could be changed easily by a frustrated administrator and it is available to load manually by changing run levels or typing “startx” at the console prompt. Therefore, the following extra safeguards should be implemented:

8.1.1.1. Disable XDMCP – Remote machines should not be able to get an X terminal login window. Edit the following lines in /etc/X11/xdm/Xaccess to prepend them with a "!" as shown.

```
!*                              #NO host can get a login window
!*          CHOOSER BROADCAST     #NO indirect host can get a chooser
```

8.1.1.2. Disable listening on port 6000 – This prevents the X system from listening for X events from remote machines. Local X access at the console is not affected. In our configuration we need to edit the KDE config file /etc/X11/xdm/Xservers as shown below adding the "-nolisten tcp" switch to this line.

```
:0 local /usr/X11R6/bin/X :0 vt07 -nolisten tcp
```

9. **Restrict cron and at** – Cron and at daemons run processes on the system as root so access to them as well as the crontab command and files so that malicious code can't be "scheduled." The binaries are also world executeable and SUID to root so they can be dangerous. We will restrict access to them with the following steps.

9.1. Create cron.allow and at.allow files – These files will restrict access to cron to only the users listed in the files. All others will be denied. The only user in the list should be root. These files don't exist by default so you can create them with the echo command as follows. Delete any deny files. (/var/spool/cron/deny)

```
# echo root > /etc/cron.allow
# echo root > /etc/at.allow
```

9.2. Modify permissions on cron/at related files – Since all cron and at files are read and written to by processes that are SUID root, normal users on the system will not ever need to have direct access to the files so they should be secured to prevent tampering.

```
# chown –R root:root /etc/cron* /var/spool/cron
# chmod –R go-rwx /etc/cron* /var/spool/cron
```

10. **Securing the File System** – This section deals with specific changes that should be made to the overall file system mount methods listed in fstab. This is more of a global file system security view instead of specific files and permissions as set above. The goal here is to prevent Trojans or attackers from introducing new binaries to the system or changing/deleting existing binaries. We also want to prevent unauthorized SUID/SGID binaries from being loaded from removeable media.

We partitioned the disk purposely with /usr on its own partition so that we could protect it in such a manner, however not all binaries are in the /usr file system. Many binaries are in /bin and /sbin and even other locations on the file system. These can't always be mounted on their own partitions either so it is hard to protect them. Below are the changes you should make to /etc/fstab in order to protect the server in the manner described.

```
/dev/hda1          /                   ext3      defaults             1 1
/dev/hda2          /usr                ext3      ro,nodev             1 2
/dev/hda6          /var                ext3      rw,nosuid,nodev      1 2
/dev/hda5          swap                swap      pri=42               0 0
devpts             /dev/pts            devpts    mode=0620,gid=5      0 0
proc               /proc               proc      defaults             0 0
usbdevfs           /proc/bus/usb       usbdevfs  noauto               0 0
/dev/cdrom     /media/cdrom        auto       ro,noauto,user,exec,nosuid,nodev   0 0
/dev/fd0       /media/floppy       auto       noauto,user,sync,nosuid,nodev      0 0
```

10.1. /usr – is mounted read only and disables the use of device files
10.2. /var – is must be mounted read-write but can be protected from SUID binaries and disabling of device files.
10.3. /cdrom, /fd0 – both are mounted such that they will not honor SUID bits or allow device files to operate.
10.4. /home – Many guides suggest adding the nosuid and nodev options on the /home file system. However, as stated in the planning section above, this server will not be hosting any home directories and will not have more than a handful of admin users created on it so the decision was made to just keep /home with the rest of the root filesystem. We may want to explore disabling the creation of home directories as new admin users are created on the system.
10.5. File permissions – The SuSEconfig system has a method of securing file permissions on the system. This is a very useful function because it secures the system but allows you to do so without breaking most applications. It is also configurable so that you can customize the settings for your own system.

10.5.1. Permissions config files - The SuSEconfig script makes use of 5 default permissions files, permissions, permissions.easy, permissions.secure, permissions.paranoid, and permissions.local. The permissions file is the base level of permissions that should be set on the file system by default. Permissions.easy is slightly more secure but allows for all easy access. Permissions.secure is meant for most multi-user systems running on the network and will greatly increase security but shouldn't break most things. Permissions.paranoid will certainly break some things in multi-user environments and is recommended for single user systems not running many if any network services. Permissions.local is the editable local file for adding in any settings that were not already defined in the other permissions files or that you want to override. The default permissions files may be changed by updates but the permissions.local should not be. Therefore this is the best place for custom additions and changes you might want to make. It is applied after the selected easy, secure, or paranoid file. We will use the permissions.secure settings. A portion of the contents of this file are listed below with some limited description of the file. The file itself is more than 10 pages long.

```
# /etc/permissions.secure
#
# Copyright (c) 2001 SuSE GmbH Nuernberg, Germany.  All rights reserved.
#
# Author: Roman Drahtmueller <draht@suse.de>, 2001
#
#
# See /etc/permissions for general hints on how to use this file.
#
# /etc/permissions.secure is designed for the use in a multi-user and
# networked installation. Most privileged file modes are disabled here.
# Many programs that still have their suid- or sgid-modes have had their
# security problems in the past already.
# The primary target of this configuration is to make the basic things
# such as changing passwords, the basic networking programs as well as
# some of the all-day work programs properly function for the unprivileged
# user. The dial-out packages are executable for users belonging to the
# "dialout" group - therefore, these users are to be treated "privileged".
~~~~SNIP~~~~~
/etc/crontab                  root.root       600
/etc/exports                  root.root       644
/etc/fstab                    root.root       644
/etc/ftpaccess                root.root       644
/etc/ftpconversions              root.root       644
/etc/ftpusers                 root.root       640
/etc/HOSTNAME                    root.root       644
/etc/hosts                    root.root       644
```

10.5.2. Test Run – The behavior of the SuSEconfig script with regard to file permissions is governed by the file /etc/sysconfig/security. This file has a parameter called "check_permissions" which can be set to warn so you can see which settings will be changed before changing them. The lower section describes which permissions.* configurations will be applied and in which order. I have changed the file to reflect the testing behavior and the "secure" settings. Below is the contents of the file and then the output when SuSEconfig is run. Note that just the permissions module is run on the command line.

```
# SuSEconfig can call chkstat to check permissions and ownerships for
# files and directories (using /etc/permissions).
# Setting to "set" will correct it, "warn" produces warnings, if
# something strange is found. Disable this feature with "no".
#
CHECK_PERMISSIONS=warn

#
# SuSE Linux contains two different configurations for
# chkstat. The differences can be found in /etc/permissions.secure
# and /etc/permissions.easy. If you create your own configuration
# (e.g. permissions.foo), you can enter the extension here as well.
#
# (easy/secure local foo whateveryouwant).
#
PERMISSION_SECURITY="secure local"
```

Now run the SuSEconfig script as below to see what would be changed after the settings we’ve applied so far.

```
# /sbin/SuSEconfig --module permissions

Starting SuSEconfig, the SuSE Configuration Tool...
Running module permissions only
Reading /etc/sysconfig and updating the system...
Executing /sbin/conf.d/SuSEconfig.permissions...
Checking permissions and ownerships - using /etc/permissions...
/usr should be root.root 755.
Checking permissions and ownerships - using /etc/permissions.secure...
/etc/crontab should be root.root 600.
/etc/ftpusers should be root.root 640.
/etc/hosts should be root.root 644.
/etc/ssh/sshd_config should be root.root 640.
/etc/syslog.conf should be root.root 600.
/usr/bin/at should be root.trusted 4750.
/usr/bin/crontab should be root.trusted 4750.
/usr/bin/gpasswd should be root.trusted 4750.
```

```
/bin/eject should be root.audio 4750.
/usr/src/packages/SOURCES should be root.root 755.
/usr/src/packages/BUILD should be root.root 755.
/usr/src/packages/RPMS should be root.root 755.
/usr/src/packages/RPMS/athlon should be root.root 755.
/usr/src/packages/RPMS/i386 should be root.root 755.
/usr/src/packages/RPMS/i486 should be root.root 755.
/usr/src/packages/RPMS/i586 should be root.root 755.
/usr/src/packages/RPMS/i686 should be root.root 755.
/usr/src/packages/RPMS/noarch should be root.root 755.
/usr/src/packages/SPECS should be root.root 755.
/usr/src/packages/SRPMS should be root.root 755.
/usr/X11R6/bin/dga should be root.root 0755.
/usr/bin/wall should be root.tty 0755.
/usr/bin/write should be root.tty 0755.
/usr/bin/ssh should be root.root 0755.
/opt/kde3/bin/artswrapper should be root.root 0755.
/opt/kde3/bin/kpac_dhcp_helper should be root.root 0755.
Checking permissions and ownerships - using /etc/permissions.local...
/home/ayerges should be ayerges.ayerges 700.
Finished.
```

The list is fairly short, but there are a few important entries. Ftp, cron, at, and ssh components should all be further protected as should RPMS and sources that might be on the machine. The /usr file system is mounted read only so much of this is already protected, but another layer of protection is certainly welcome.

10.5.3. Edit Local settings – After looking at what will be done, some additional lines have been added to the permissions.local file as shown below. Most of these are select lines from the permissions.paranoid configuration which take away world/group readable bits or remove the SUID bit from specific binaries to protect from exploitation if buffer overflows are found and protect them from other malicious use.

```
# example:
#/usr/local/bin/mtr        root.root      4755
/home/ayerges              ayerges.ayerges 700
/root                  root.root      700
/etc/ftpusers              root.root      600
/etc/grub.conf             root.root      600
/etc/cron.*                root.root      700
/bin/mount                 root.root       0755
/bin/umount                root.root       0755
/usr/bin/fdmount           root.root       0755
/usr/bin/ncpmount          root.trusted    0755
/usr/bin/ncpumount         root.trusted    0755
/usr/bin/vmware-ping       root.root       0755
/usr/bin/ntping            root.trusted   0755
/bin/ping                  root.root       0755
/bin/ping6                 root.root       0755
/usr/bin/chfn              root.shadow     0755
/usr/bin/chsh              root.shadow     0755
/usr/bin/chage             root.shadow     0755
```

10.5.4. Apply the settings – In order to actually apply the settings we have to do a few things

10.5.4.1. Remount /usr – Since the /usr file system is mounted read only, we have to mount RW in order to apply these changes to the files. Do this with the following command:

```
# mount –o remount,rw /usr
```

10.5.4.2. Edit /etc/sysconfig/security – Change the "CHECK_PERMISSIONS" parameter to "set" so that the changes will be applied.

10.5.4.3. Run command - "SuSEconfig --module permissions" in order to apply the permissions.

11. Host Based IPTables firewall – To protect the server we will configure the iptables firewall built into the kernel. SuSE comes with its own front end for configuring basic firewall functionality with iptables, but I chose not to use that because we want much more granular and specific behavior. Most of the other scripts and tools I looked at out there are quite good but also limit the functionality I was looking for so I started with a fairly plain but well documented script and modified it (A Sample Firewall Configuration. In *Linux Network Administrators Guide*: http://www.faqs.org/docs/linux_network/x-087-2-firewall.example.html). The goal is to block both inbound and outbound traffic for everything except desired traffic and to differentiate between traffic from the inside of the network and traffic from the Internet. Here is the script to do this. It should be placed in a file called /etc/fwup.sh (found in the Appendices at the end of the paper).

11.1. fwdown – A script to take down the firewall easily for troubleshooting, etc… should be created in /etc/fwdown.sh. The contents should be as follows.

```
#!/bin/bash
###############################################################################
#
# IPTABLES VERSION
# This sample configuration is for disabling the active firewall configuration
#
###############################################################################
# The name and location of the ipchains utility.
IPTABLES=iptables

# Flush the ALL table rules
$IPTABLES -F

# We want to change the default action for INPUT to ACCEPT.
$IPTABLES -P INPUT ACCEPT
#
# end
```

11.2. Make scripts executable – Make both of these scripts executable and add an S0Xfwup link in /etc/init.d/rc3.d. Also make sure the scripts are owned by root and all other permissions are taken away.

```
Make the startup link.  Create at a number order that is after network, but before other
services.
# ln -s /etc/fwup.sh /etc/init.d/rc3.d/S08fwup

Now ownership and executable bits will be set
# chown root:root /etc/fwup.sh /etc/fwdown.sh
# chmod 700 /etc/fwup.sh /etc/fwdown.sh
```

11.3. Test – After installing these scripts carefully test applications and access while watching the logs generated carefully. This is best done in a low traffic environment so that logging is not too extensive to effectively monitor it. You may need to add lines to accept or deny other specific traffic as noted by monitoring the logs (/var/log/warn) and knowing your specific environment.

12. **Logging Environment** – As discussed above, logging should ideally be sent to a central syslog server instead of/in addition to the local syslog daemon. Either way, the local syslog environment should be changed to separate out logs into clearer and more readable log files to make it easier to spot problems. We also need to make sure all types of logging are properly taking

place from the info level and higher. The logging for the Novell provided daemons is defined by the daemon conf files and by default is put in the application specific */log sub-directories within /var/opt/novell. Application logging will be addressed in the application specific security sections later in this paper.

12.1. First we put in a "catchall" to log any info or greater messages not specifically sent to other log files specifying the exceptions in the line.

12.2. Then we establish logging for each facility excepted in the catchall line. Below is the portion of the file which shows this setup.

```
# Log severity level info and above to get more detailed
# logging. This line logs all Info messages EXCEPT those
# going to authpriv, auth mail etc... Those will go to
# their own facility below.
*.info;authpriv,auth,mail,cron,kern,lpr,lo7cal7.none    /var/log/messages

# all email-messages in one file
#
mail.*                         /var/log/maillog

# Messages about logins and authorizations should be sent
# to one log file
authpriv,auth.*                /var/log/secure

# Put cron and at messages in a specific file
cron.*                         /var/log/cron

# Kernel messages in a separate file. Includes IPTABLES
# messages because it is built into the kernel.
kern.*                         /var/log/kernel

# Specify boot messages to go into their own file
local7.*                       /var/log/boot.log

# Send printing log files to their own file as well
lpr.*                          /var/log/spooler
```

12.3. Log Rotation – Log rotation needs to be set up so that logs do not fill the drive but remain long enough for proper analysis (this is discussed in the "Ongoing Maintenance" section below.)

12.3.1. Default settings – First we edit the default settings for all files in the /var/log directory. We want to keep logs longer by default but compress them to conserve space. The following /etc/syslog.conf file shows these changes.

```
# rotate log files monthly
monthly

# keep 12 months worth of backlogs
rotate 12

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
compress
```

12.3.1.1. Next we edit the /etc/logrotate.d/syslog file to be sure each new log file is listed in this file and all of the settings are correct. Here is the content that should be in this file

```
/var/log/warn /var/log/messages /var/log/secure /var/log/localmessages
/var/log/cron /var/log/kernel /var/log/boot.log /var/log/spooler {
    compress
    dateext
    maxage 365
    rotate 99
    missingok
    notifempty
    size +4096k
    create 644 root root
    sharedscripts
    postrotate
        /etc/init.d/syslog reload
    endscript
}
```

12.3.1.2. Other files in /etc/logrotate.d/ directory should be reviewed since they specify settings for rotating other logs for other services and daemons. For the most part the default settings are fine for these but you may want to edit the Novell provided scripts for your purposes. Notice that they are symlinks to the actual files in the /etc/opt/novell/httpd/logrotate.d/ and /etc/opt/novell/tomcat4/logrotate.d/ directories where the conf files and other program related files are located for these applications. We will address these files and application security below.

Application Hardening

1. **Apache Security** – Since Apache is one of the main components of the services provided by the server it is certainly important to secure it as well as we can. Many documents have been written with regard to Apache security. The full discussion of this is beyond the scope of this paper. However, we will discuss the specific configuration changes that should be made to help secure this particular server and refer to some excellent sources for reference in making those changes. Any of the changes that will be suggested might be overwritten by upgrading or patching the applications so they should be rechecked carefully after every patch or upgrade is applied. Many of these changes will be made in /etc/opt/novell/httpd/conf/httpd.conf. Note that there is also a directory /etc/opt/novell/httpd/conf.d/ which contains links to other conf files that are automatically processed and used by Apache as it starts up. These represent many of the extra modules that are required for functioning of NNLS. These should be left as is in order for the application to be supportable by Novell.

   1.1. Patch code – Probably the most important step is to make sure that the code is patched and up to date. Since this is part of a packaged product, it is not suggested that the Apache version be updated apart from code that has been tested and certified by Novell to work with the rest of the components. We will ensure that Apache is up to date, tested, and patched using the Red Carpet client functionality that is installed by default with the product. This is covered further in the Ongoing Maintenance section later in the paper.

   1.2. Remove Unnecessary Modules – (Caution: might not be supported by Novell) The following modules should probably be removed since they are not needed by the products we installed. Extra modules slow the server down and offer more avenues for attack so we should take out all that we can. Comment out the "LoadModule" lines for each.

   - **1.2.1. mod_autoindex** – This module sends a directory listing to a query to a directory that has no index.html or similar default page. This gives away too much information to anyone doing reconnaissance on our site.
   - **1.2.2. mod_userdir** – This module allows pages within a user's home directory to be served up and should be disabled so that unauthorized content that might not be secure is not displayed.
   - **1.2.3. Index/dir related** – Removing the directives above will also require that all other directives relating to creating file indexes and icons that go with specific file types will have to be remarked out as well. Executing the reload command for Apache (/etc/init.d/novell-httpd reload) will show errors for the lines that must be remarked if you aren't sure which ones they are.

   1.3. Server Information – (Caution: might not be supported by Novell) We don't want to give away server version information so there are a few other parameters to change:

1.3.1. **ServerSignature** – This directive should be changed to “Off” to keep the server name, version, and modules from being displayed by default.

1.3.2. **ServerTokens** – This directive should be set to “Prod” which sends the least amount of information in the HTTP header that is always sent back to a requesting browser. The default sends back all information about the server version and loaded modules that it can.

1.3.3. **ServerAdmin** – An aliased e-mail address (not a real user id) should be used here. If we put a real id in the field then it could potentially give the attacker an id to try against this or other systems in the environment. Something like Webmaster@domain.com for your domain would be better.

1.4. Logging – The log level should be set to “notice” to increase the level of logging so that we can see if we have a problem. Logs, again, should also be sent to a central syslog server in case of compromise of this host.

1.5. Directory Permissions – (Caution: might not be supported by Novell)We need to tighten the directory permissions that are set in the default configuration. However, these permissions might get reset if an upgrade or patch is applied. If this occurs, the changes must be made again to put it back into effect. There are two sections for this right after the “DocumentRoot” directive.

1.5.1. **Default directory** – Here we need to change it to the following

```
# First, we configure the "default" to be a very restrictive set of
# features.
#
<Directory />
   Options SymLinksifOwnerMatch
   AllowOverride None
   Order allow,deny
   Deny from all
</Directory>
```

This will restrict access to any other path on the server other than the “DocumentRoot” path. If this is not done, someone could place a symlink somewhere on the server and the server could follow it to any location in the filesystem.

1.5.2. **DocumentRoot directory** – Here we need to change the defaults to what is shown below. This tightens the security by removing the defaults of “indexes” and “FollowSymLinks.” SymLinks are allowed by this configuration but only if they are owned by root or the novlwww user which is the web server user preventing anyone else from creating SymLinks that would be followed by the server to lead elsewhere in the filesystem.

```
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "/var/opt/novell/httpd/htdocs">
    Options SymLinksifOwnerMatch
    AllowOverride None
    Order allow,deny
    Allow from all

</Directory>
```

1.5.3. Change SymLink ownership – In order for all of the pages to function, any links not owned by the user novlwww must be changed. There are about 5 in the default install. To find them and change them, execute the following command.

```
# find /var/opt/novell/ -user root -type l -exec chown novlwww:root {} \;
```

1.5.4. Permissions to DocumentRoot – Most guides suggest making sure that no documents are writable by the web server user (novlwww in this case). The default is to have all of the files in DocumentRoot owned by root so they should be protected. This should help prevent most defacement attempts from newfound bugs in the Apache code. The other resource directories all have files owned by novlwww user ostensibly to allow writing files to the file system for the application. These must be left as is for now, but I would suggest Novell look into the real security needs of these directories according to their functions. See further discussion in the Tomcat section below.

2. **Tomcat Servlet Engine** – Next we will discuss the security of Tomcat in this configuration and how best to address the issues raised here. With the goal stated in the planning section above of keeping our configuration “supportable” by Novell we can’t change too many parts of Tomcat. The built in security files or the <Realm> tags, both discussed in the Tomcat documentation (http://jakarta.apache.org/tomcat/tomcat-4.1-doc/), can’t be implemented with this requirement in mind. If changes are made to specific configuration files, they will also probably be replaced when patches or upgrades are applied and the files get overwritten. Despite these warnings I outline a method below for increasing the security of the Tomcat server by changing some of the default file permissions.

2.1. Patching – As with Apache above, Red Carpet will be used to keep the patches up to date in order to protect against new vulnerabilities as they are found and patched. Keeping the package in synch with Novell’s controlled and tested patches is very important for all parts of the applications to work together.

2.2. File Permissions – (Caution: might not be supported by Novell) The default file permissions that are set upon installation of the product give the novlwww user write access to almost every file that is part of one of the web applications that run under Tomcat. The default permissions assigned are 644 so that the files are world and group readable but only writeable by the owner. So, we will change the owner of the files to remedy this.

2.2.1. Tomcat applications – The applications that we are securing under Tomcat are: eGuide, iManager, iFolder, NetStorage, and the tomcat root. A good understanding of how these applications function is critical for troubleshooting if changing the permissions causes one of them to fail.

2.2.2. Permissions Discussion – If the user id that Tomcat uses to run has file system rights on the server, especially in the content directories, then it is much easier for an attacker to introduce compromised code to attack the server.

A balance must be maintained here though because some files must be writable by the Tomcat user in order for the applications to function correctly. A specific example is iFolder. The Tomcat user must be able to write data to the iFolder directories on behalf of the users connecting to it. The data is encrypted by the module and written to the disk in an encrypted format. The module also requires both a valid password and a pass phrase in order to access or modify the data.

2.2.3. My permissions changes – The following command worked for me and still allowed all of the applications to function correctly. This should improve the security stance of the installed application but, again, use this with caution. You may want to tar up the original file structure before attempting this command so that you can go back to the original structure and permissions if necessary.

```
# cd /var/opt/novell
#
# find eGuide/ iManager/ ifolder/ netstorage/ novlwww/ tomcat4/ xtier/
! \
-type d ! -type l -user novlwww -perm +202 ! -iname *.log ! -iname
*.txt \
 ! -iname *.gz ! iname *.out -exec chown root:novlwww {} \;
```

Analyzing this command a little is probably helpful here. We are using the find command here to list all of the files that are owned by the Tomcat user and that this user has write access to. We are also excluding log files, directories, symlinks, and zipped log files. This list is then passed to chown in order to change the user ownership to root while keeping the group ownership the same. This allows configuration files and application files to remain readable to the Tomcat engine but takes away write access. The exceptions in the command (logs, symlinks, and directories) are important since the

Tomcat user must be able to write log files and the symlinks must be owned by the Apache user in order to operate as discussed in the Apache section above.

3. **Protecting the entire web services environment** – To add another layer of protection against typical web attacks I suggest adding an application firewall/IDS. The one I suggest currently is mod_security. This is an open source intrusion detection and prevention engine for Apache web servers. (http://www.modsecurity.org). Since it is part of the web server itself, it offers many advantages. It can operate effectively with SSL because the requests are decrypted before the module can take action on them. It can monitor GET, HEAD, and POST processes. It is part of the HTTP processing engine so it can fully understand the protocol and monitor it in a very finely grained manner. Mod_security also fits the requirement of keeping the system supportable by Novell because it can simply be disabled for testing purposes should it come into question during a support incident.

   3.1. Download and Install – The installation files can be downloaded from http://www.modsecurity.org/download/index.html. The version I installed was the 1.7.6 version. Since the Novell provided Apache server is a stripped down version without the development tools and there are no precompiled binaries for SuSE, we will have to download, compile, and install Apache in a different path and then use that installation to compile the mod_security binary for use on our system.

   3.1.1. Download/Install Apache – The current apache version included with NNLS is 2.0.48. To avoid problems, download this version from the archives section of the apache site. (http://archive.apache.org/dist/httpd/httpd-2.0.48.tar.gz) Check the md5sum against the posted value to be sure the file is intact and has not been tampered with. Then extract the archive and install the software according to the instructions in the INSTALL file. This will not overwrite the current Apache installation because the default path is /usr/local/apache2.
   Note: don't forget to remount the /usr filesystem as Read Write temporarily so that the install will work.

   3.1.2. Download mod_security – The source files for mod_security must also be downloaded and extracted in order to compile the module for use. Download from the site given above and extract the files after again confirming the md5sums as we did with the Apache distribution.

   3.1.3. Compile mod_security – To compile the code with the newly installed Apache version full paths should be used. The command is otherwise identical to the one given in the mod_security documentation from the distribution under the DSO heading.

```
/usr/local/apache2/bin/apsx –cia /<path-to-extracted-mod_security-
```

mod>/apache2/mod_security.c

Once compilation finishes, a mod_security.so file should be found in the path /usr/local/apache2/modules. This is the binary that we need for the Novell supplied Apache server to run mod_security.

3.1.4. Install mod_security – To install the module to the server we are running, copy the mod_security.so file noted above to the path /opt/novell/httpd/modules and add the following line to the httpd.conf file at the bottom of the modules section. The following is an excerpt of this file showing the line you should add. Restart the web server to be sure that the module loads without error.

```
# vi /etc/opt/novell/httpd/conf/httpd.conf

#LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
#LoadModule rewrite_module modules/mod_rewrite.so
# The following module adds IDS functionality to Apache
LoadModule security_module modules/mod_security.so
#
```

3.1.5. Configure mod_security – After it is installed, a base configuration should be added per the documentation. I will not discuss all of the options here. Some good references for this are the www.modsecurity.org website and http://www.onlamp.com/pub/a/apache/2003/11/26/mod_security.html , http://www.hackinthebox.org/print.php?sid=12867 , and http://www.securityfocus.com/infocus/1739 . My basic technique was to use much of the configuration from the security focus article and then add the full converted SNORT rule set from the mod_security distribution taking out the specific rules I knew didn't apply to this server (i.e. IIS rules, .cmd rules and .exe rules). After testing, I had to remove some more of the rules because they conflicted with the operation of the iFolder, iPrint, and Red Carpet. My full mod_security rule set can be found in the appendices.

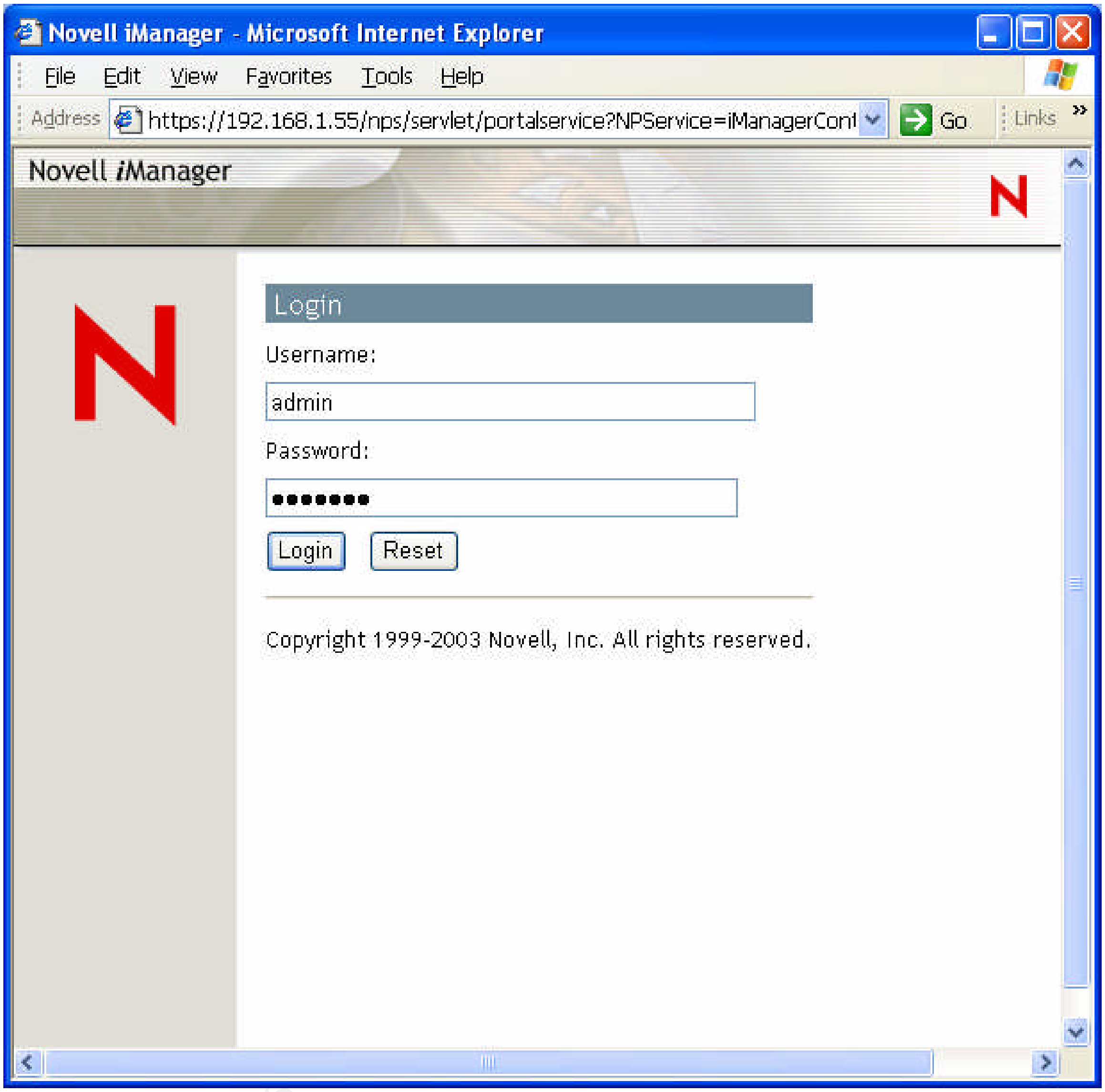
Mod_security also has built in chroot capabilities. This looks like an interesting addition to the configuration but I chose not to implement it because it has some known problems that the author has fixed in the upcoming 1.8 release (http://www.modsecurity.org/download/CHANGES).

4. **Postfix –** By default SendMail is not installed. Postfix is our local mail handler. The default configuration for Postfix is quite secure as long as no external mail server is running on this server. Port 25 (smtp) is only listening on the loopback interface so it is not accessible from the network. Relaying is not enabled and no forwarding address is configured. As initially installed, mail to root is forwarded to the additional user created so that it is not necessary to log in as root or even "su" to root in order to get important

system messages.

5. **LDAP –** LDAP is provided on this server through eDirectory. Therefore control of LDAP is done through iManager and setting specific eDirectory settings. When the LDAP services come up they read their configuration out of the directory and behave accordingly. When we installed NNLS we specified 636 as the LDAP access port for all applications. This is the default TLS/SSL port for conducting encrypted LDAP sessions. A few changes should be made from the defaults in order to ensure that the communications are encrypted and that passwords are required to access the LDAP information.

   5.1. Open iManager and Find LDAP objects – Since this will all be done in iManager, we need to open it up and log in. Use the following URL substituting your IP address as appropriate https://192.168.1.55/nps/iManager.html. Log in with the admin user you created/linked to during the install.


Novell iManager - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Address
https://192.168.1.55/nps/servlet/portalservice?NPService=iManagerConf
Go
Links
Novell iManager
Login
Reset

Login

Username:

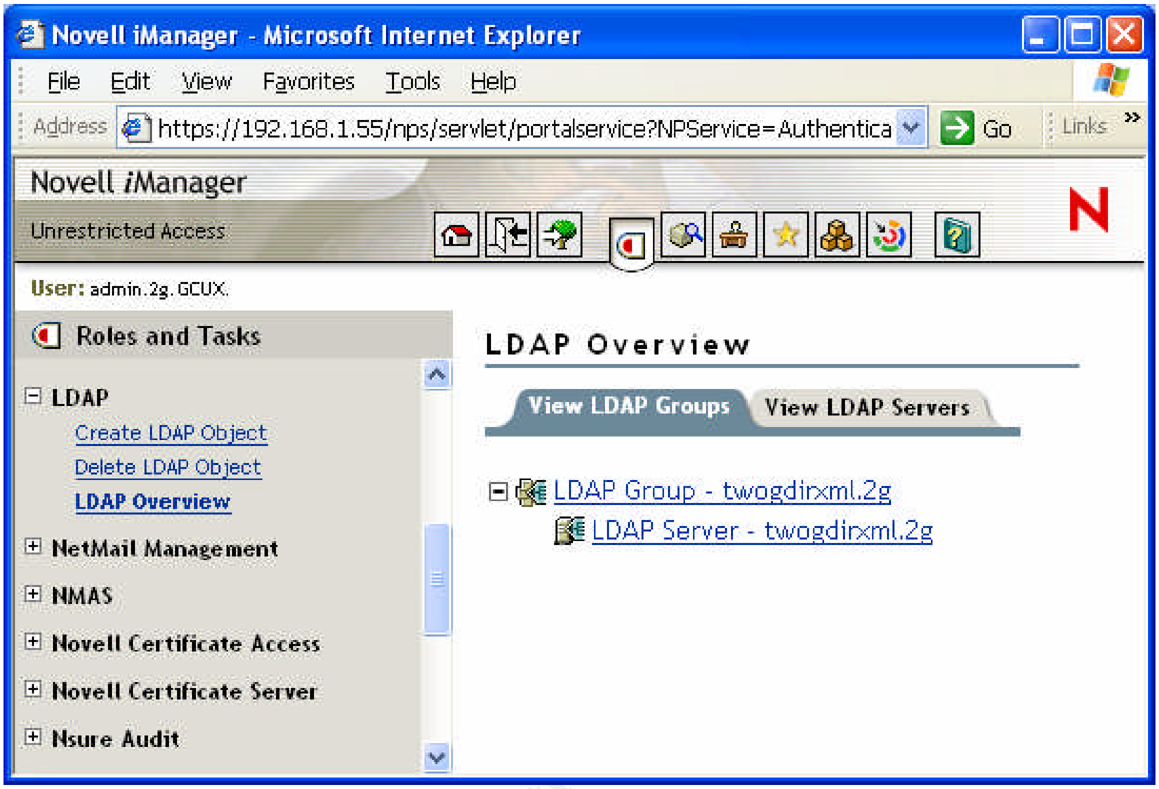
admin

Password:

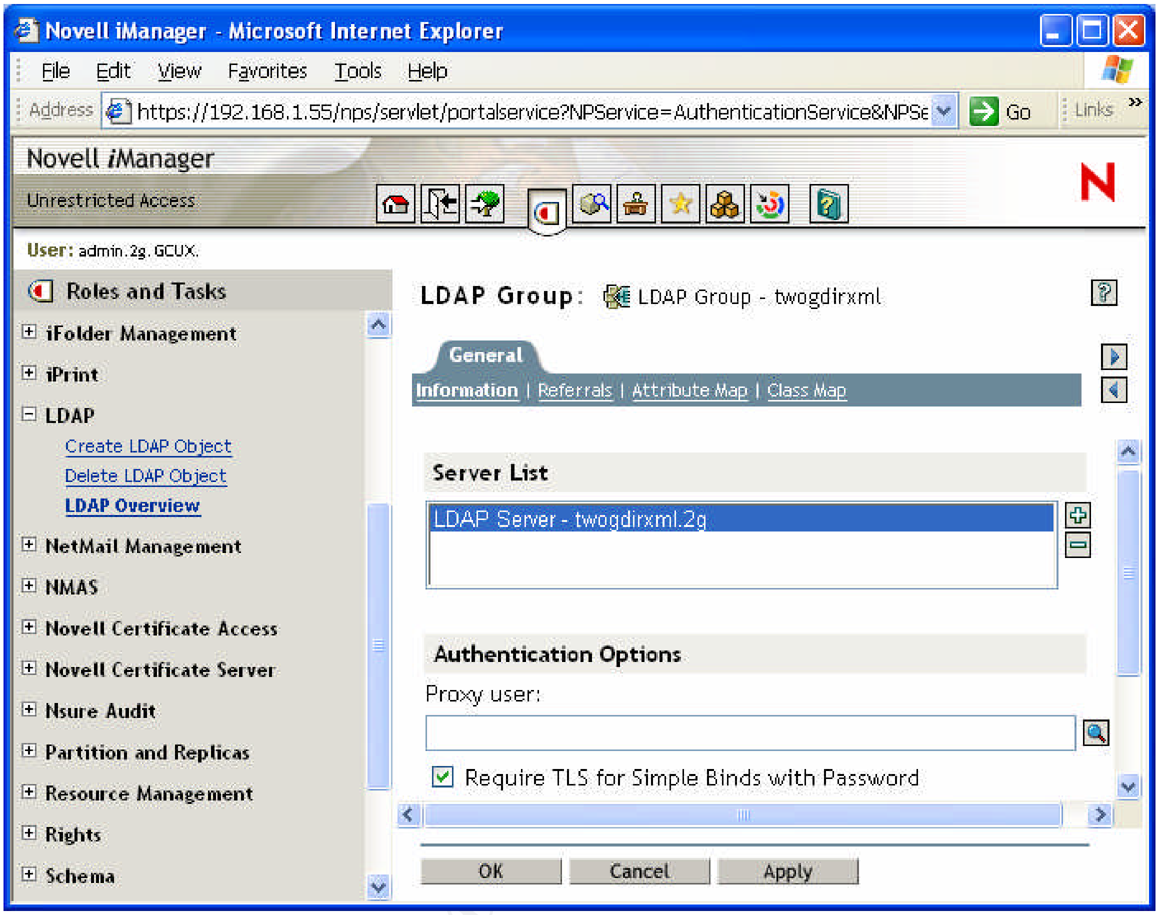
●●●●●●●

Copyright 1999-2003 Novell, Inc. All rights reserved.

Once logged in, expand the LDAP section on the left and select the “LDAP overview” link. On the right expand the LDAP configuration for your server.


Novell iManager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address https://192.168.1.55/nps/servlet/portalservice?NPService=Authentica
Go
Links
Novell iManager
Unrestricted Access
User: admin.2g.GCUX.
Roles and Tasks
LDAP
Create LDAP Object
Delete LDAP Object
LDAP Overview
NetMail Management
NMAS
Novell Certificate Access
Novell Certificate Server
Nsure Audit
LDAP Overview
View LDAP Groups
View LDAP Servers
LDAP Group - twogdirxml.2g
LDAP Server - twogdirxml.2g

5.2. **LDAP Group Object –** The group object and the server object are both going to be edited. First click on the group object link. We are not setting a proxy user for this configuration because we do not require any unauthenticated extended access to directory data. Default access is granted to the "Public" user in eDirectory. This access is sufficient. Ensure that the "Require TLS for simple bind with password" is checked. This setting requires SSL/TLS any time a password is sent for an LDAP request. If an unencrypted password is sent it will not be accepted. The first time it will be sent in the clear but enabling this option discourages plain text use because it will fail. We will prevent any unencrypted connections from being made in the "LDAP Server" section below.

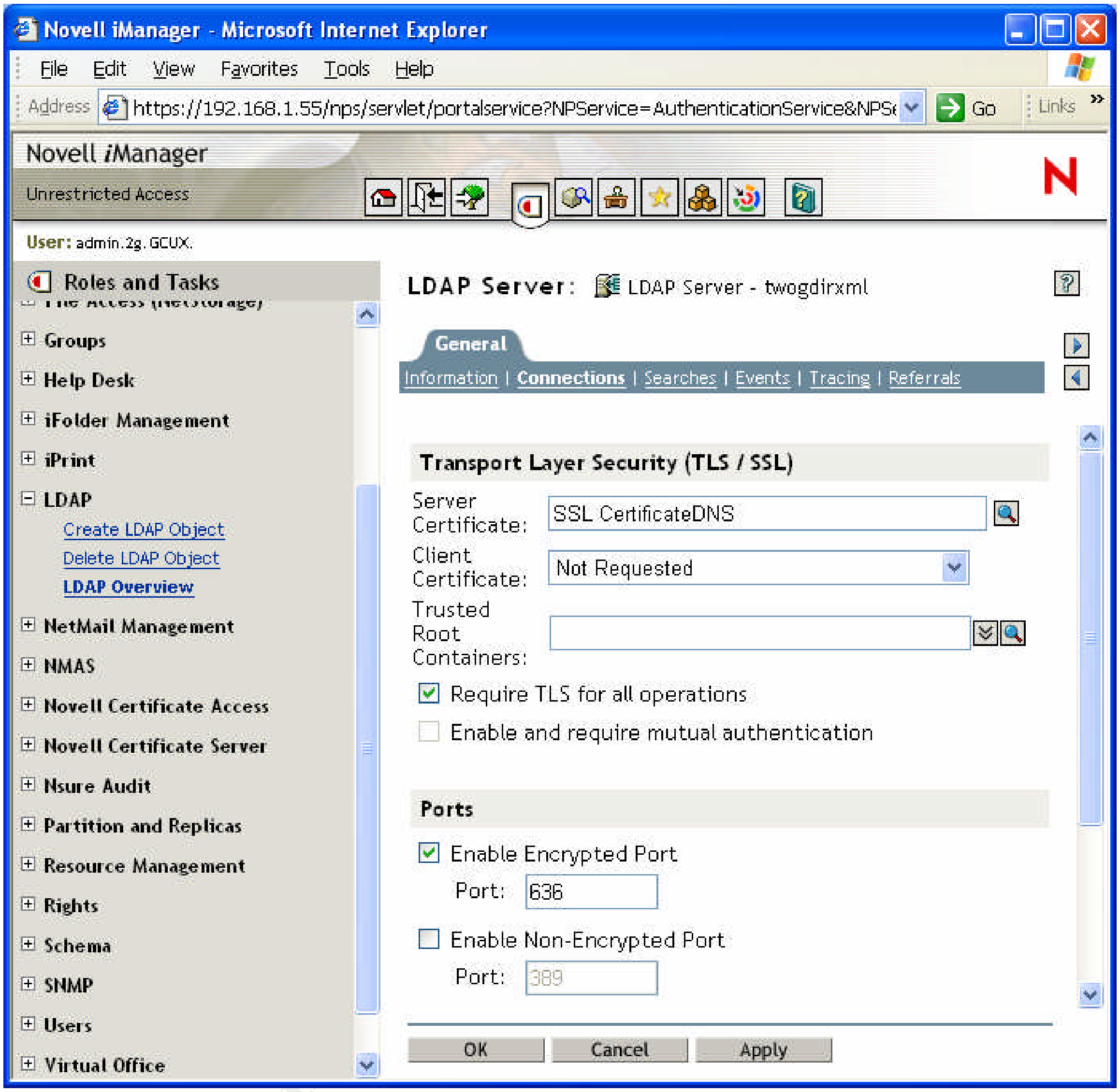

Novell iManager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address https://192.168.1.55/nps/servlet/portalservice?NPService=AuthenticationService&NPSe
Go
Links
Novell iManager
Unrestricted Access
User: admin.2g.GCUX.
Roles and Tasks
iFolder Management
iPrint
LDAP
Create LDAP Object
Delete LDAP Object
LDAP Overview
NetMail Management
NMAS
Novell Certificate Access
Novell Certificate Server
Nsure Audit
Partition and Replicas
Resource Management
Rights
Schema
LDAP Group: LDAP Group - twogdirxml
General
Information | Referrals | Attribute Map | Class Map
Server List
Authentication Options
Proxy user:
OK
Cancel
Apply

Server List: LDAP Server - twogdirxml.2g

Proxy user:

☑ Require TLS for Simple Binds with Password

5.3. LDAP Server Object – Next we will make the changes required on the server object. For this object we will make changes in the “connections” link on the main configuration page. The first thing to look at is the “Server Certificate” field. This is the SSL certificate used by the server to set up secure communications. The certificate is stored in the directory and read each time the server starts up. This certificate must be valid and accessible for TLS communications to take place. Note that we are checking the box “Require TLS for all operations” and unchecking the port 389 box below that. Now the server will only listen on the secure 636 port and require TLS for all communications.



LDAP Server: LDAP Server - twogdirxml

Transport Layer Security (TLS / SSL)

Server Certificate: SSL CertificateDNS

Client Certificate: Not Requested

Trusted Root Containers:

☑ Require TLS for all operations

☐ Enable and require mutual authentication

Ports

☑ Enable Encrypted Port

Port: 636

☐ Enable Non-Encrypted Port

Port: 389

6. **Netware Core Protocol (NCP)/eDirectory –** This is the protocol used by eDirectory clients and servers to talk to each other by default. In the NetWare OS there was an option to add packet signatures to prevent man-in-the-middle or packet insertion attacks. Unfortunately this is not available on the Unix/Linux platforms. This is an option I would like to see Novell offer in the future. Although not all shops will use it, there is certainly some value to this capability.
   The rights assigned to the eDirectory database files are set to 600 which are appropriately secure. The database files themselves are written in an encrypted format so they are not accessible to prying eyes even if someone was able to gain root access.
   Some of the eDirectory tools and utilities should have their default permissions changed to keep them from being world readable and executable. Although I would prefer to set the same rights on all of the library

files found in /usr/lib/ and /usr/lib/nds* directories and subdirectories I cannot recommend this because of the need to keep the server "supportable" by Novell as stated above. The following command will change the permissions to more sane values on the main eDirectory utility files included on the system. As before, make sure your /usr file system is mounted RW in order for this command to work.

```
# chmod 750 /usr/bin/nds*
```

7. **DirXML(Identity Manager) –** Identity Manager uses XML files transferred between applications running on the source and destination servers in order to manage directory information between disparate systems. It is the core of Novell's "Meta Directory" solution and is very customizable and extensible. Because of the direct access into the respective directory stores, protection of these connections, applications, and data is very important. The solution is architected to use SSL certificates for data encryption and authentication. This is well documented and should be configured for every connection made with another directory as well as remote loader and control connections. In conjunction with these certificates, the iptables firewall solution discussed above will block most connections to these ports from unauthorized hosts. I suggest that more specific entries be made so that only servers that need to communicate with this server be allowed access. I also suggest that the script files be protected by modifying the default rights on them to prevent them from being accessed by all users. To do this execute the following command with the /usr file system mounted RW.

   ```
   # chmod 750 /usr/bin/dxml*
   ```

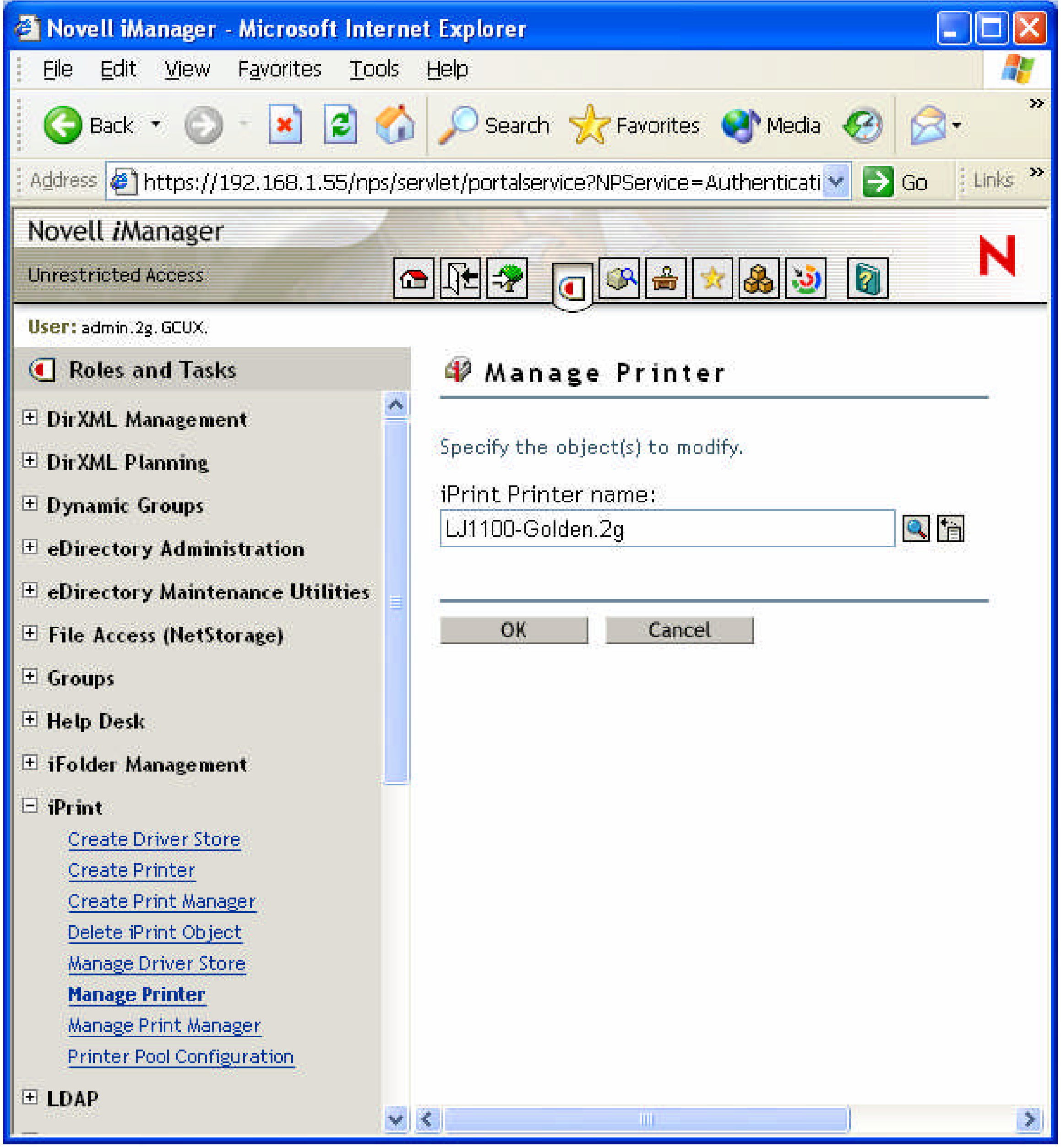
   With a tool of this power it should be emphasized that great care must be taken to prevent loss of data or functionality simply from misconfiguration or poor planning. Make sure you know exactly what you are doing or have contracted with a qualified consulting organization before putting this solution into production. Further discussion of installation and configuration is beyond the scope of this document. Please see the Novell documentation. (http://www.novell.com/documentation/dirxml20/index.html)

8. **iPrint printing –** Printing over IPP was also installed on this server and there are a couple of security configurations to consider with this. IPP support is provided by an Apache module so much of the security has been taken care of by the Apache security configuration above. Printing from the Internet, if allowed, should be restricted to SSL printing which requires authentication and encrypts the job by default.

   General setup of an IPP printer is not covered here. Documentation is available (http://www.novell.com/documentation/nnls/pdfdoc/iprint/iprint.pdf)

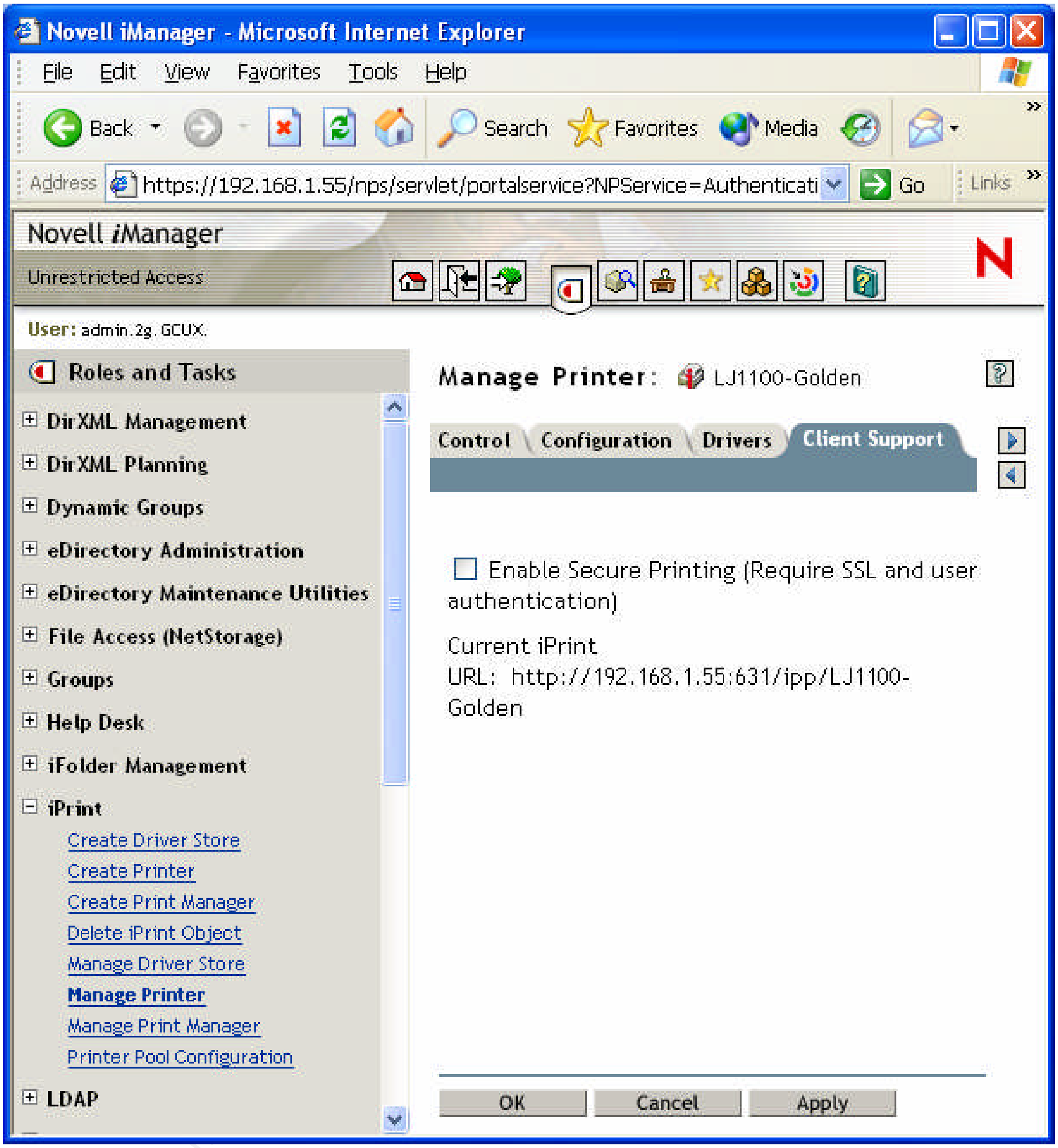
In addition to this basic documentation we are going to enable SSL based IPP printing by making the following change in iManager.

8.1. IPP over SSL – To make the change, open iManager as you did before browse to the "iPrint" section on the left, expand it, select "manage printer", and enter or browse to your previously created printer object and click OK.



iPrint Printer name: LJ1100-Golden.2g

On the screen that appears, select the "client support" tab. Check the box next to "Enable Secure Printing" and click the "apply" button near the bottom.



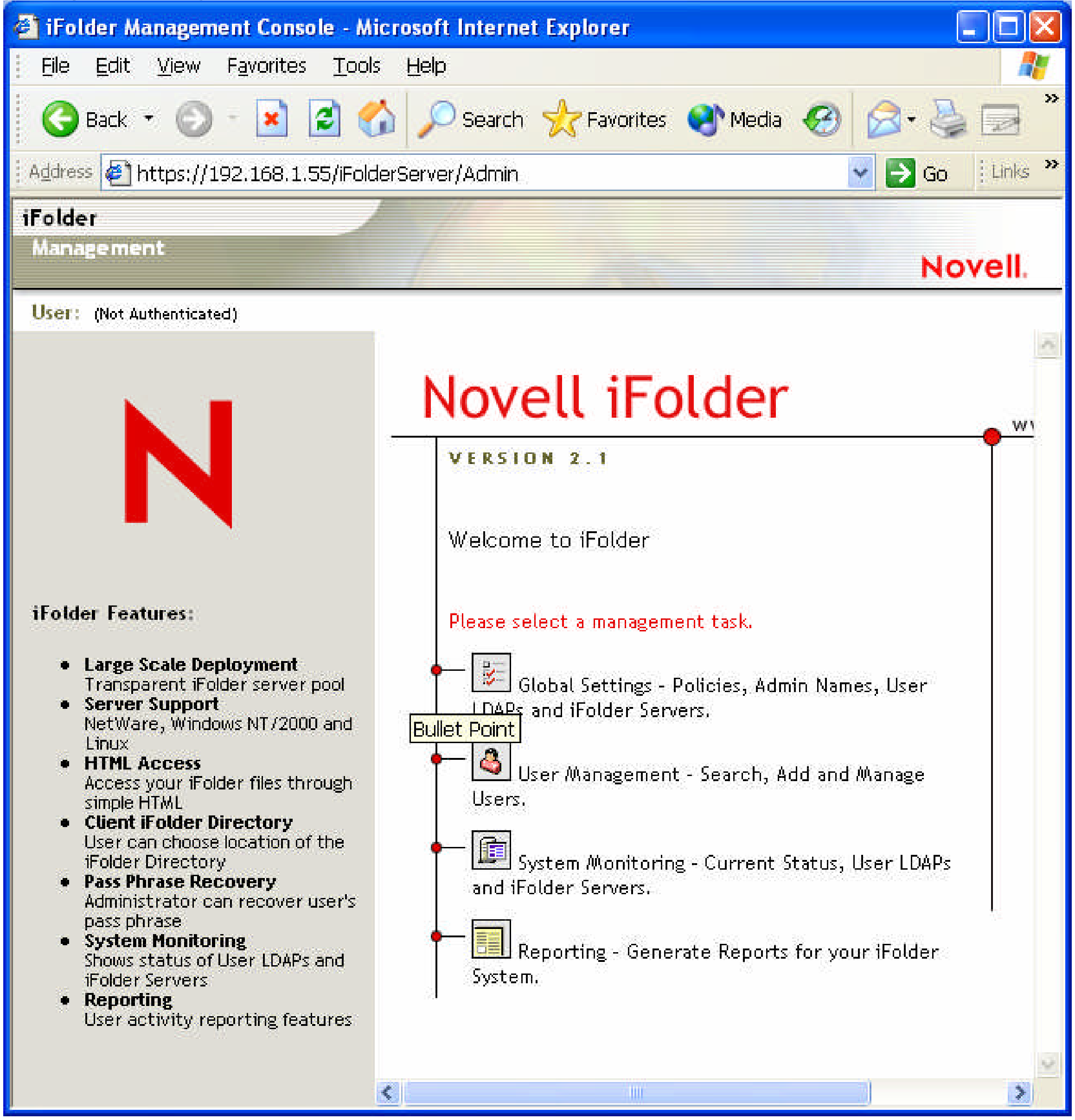
☐ Enable Secure Printing (Require SSL and user authentication)

Current iPrint URL: http://192.168.1.55:631/ipp/LJ1100-Golden

After applying the changes you will see a success message and your printer should now be set up for SSL printing. You should test it by running through the printer install process. You may also have to adjust the “access control” settings to ensure that the proper users are allowed to use the printer. Again the documentation cited above can help with this configuration if you require assistance.

9. **iFolder security –** iFolder is another one of those services that is offered through Apache. It also has some significant security features that should be enabled/edited so I will cover those in this section. Further configuration questions can be addressed with the online documentation (http://www.novell.com/documentation/ifolder21/pdfdoc/admin/admin.pdf).

9.1. iFolder Global Client Policies – First we will discuss and set the global settings that relate to security. To access the iFolder administration page connect to the following URL substituting the correct IP/DNS name for your environment. https://192.168.1.55/iFolderServer/Admin. Once you click on any button (We want the global settings button in this case) you will be prompted to log in.



User: (Not Authenticated)

iFolder Features:

- **Large Scale Deployment**
  Transparent iFolder server pool
- **Server Support**
  NetWare, Windows NT/2000 and Linux
- **HTML Access**
  Access your iFolder files through simple HTML
- **Client iFolder Directory**
  User can choose location of the iFolder Directory
- **Pass Phrase Recovery**
  Administrator can recover user's pass phrase
- **System Monitoring**
  Shows status of User LDAPs and iFolder Servers
- **Reporting**
  User activity reporting features

Novell iFolder

VERSION 2.1

Welcome to iFolder

Please select a management task.

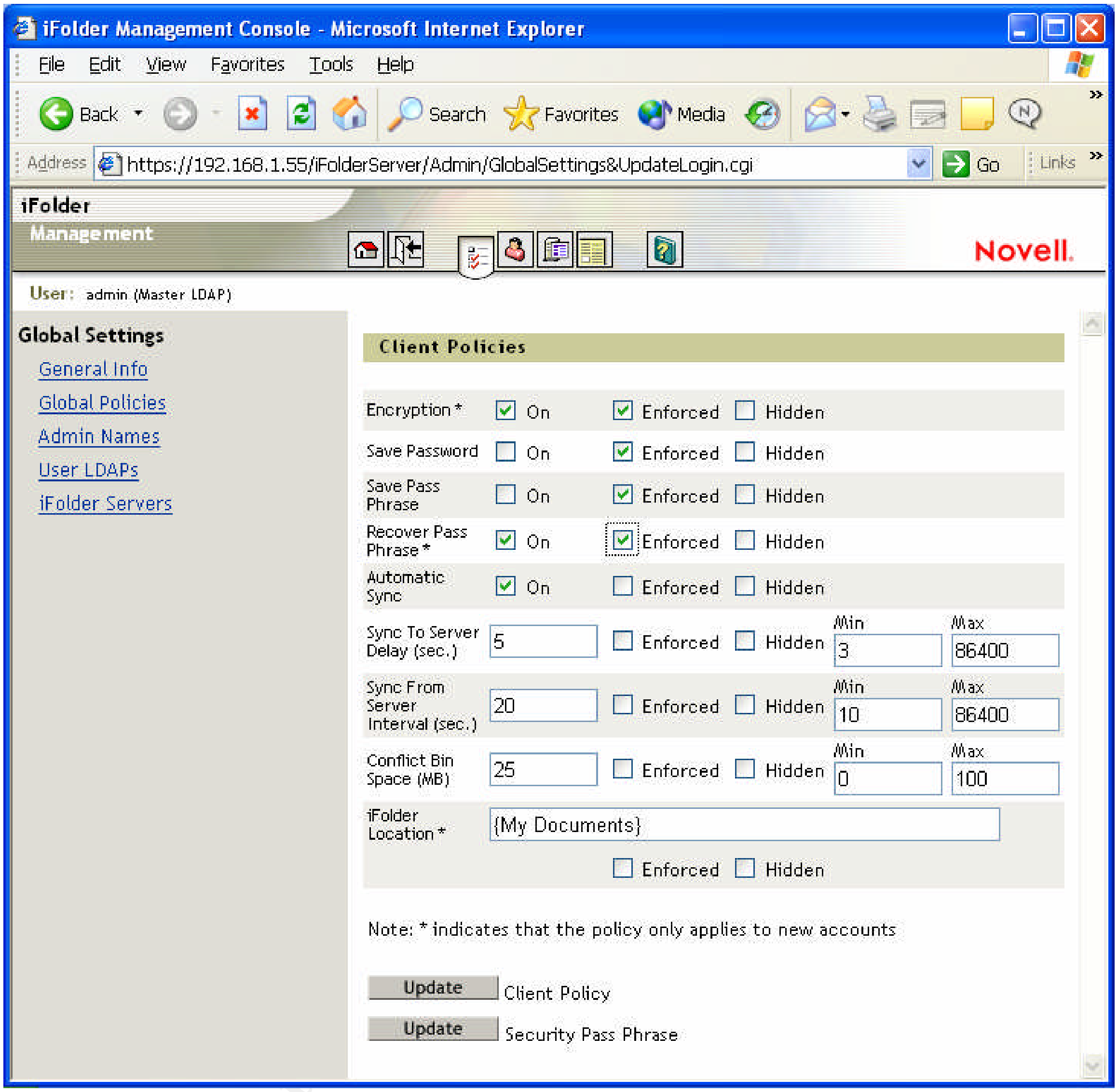
Global Settings - Policies, Admin Names, User LDAPs and iFolder Servers.

User Management - Search, Add and Manage Users.

System Monitoring - Current Status, User LDAPs and iFolder Servers.

Reporting - Generate Reports for your iFolder System.

After logging in, choose the “Global Policies” link on the left and then select the “client policies” button on the right. Edit the policies as shown in the picture below.


iFolder Management Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address https://192.168.1.55/iFolderServer/Admin/GlobalSettings&UpdateLogin.cgi
Go
Links
iFolder Management
Novell.
Global Settings
General Info
Global Policies
Admin Names
User LDAPs
iFolder Servers

User: admin (Master LDAP)

Client Policies

				Min	Max
Encryption *	☑ On	☑ Enforced	□ Hidden		
Save Password	□ On	☑ Enforced	□ Hidden		
Save Pass Phrase	□ On	☑ Enforced	□ Hidden		
Recover Pass Phrase *	☑ On	☑ Enforced	□ Hidden		
Automatic Sync	☑ On	□ Enforced	□ Hidden		
Sync To Server Delay (sec.)	5	□ Enforced	□ Hidden	3	86400
Sync From Server Interval (sec.)	20	□ Enforced	□ Hidden	10	86400
Conflict Bin Space (MB)	25	□ Enforced	□ Hidden	0	100
iFolder Location *	{My Documents}	□ Enforced	□ Hidden		

Note: * indicates that the policy only applies to new accounts

Update Client Policy

Update Security Pass Phrase

For the highest security we want to enforce encryption and disable the saving of passwords. We also want to enable the ability to recover pass phrases.

9.2. iFolder Encryption – A quick discussion of the pass phrase and encryption in iFolder is in order. iFolder runs over port 80, HTTP which is typically a clear text port/protocol. However, the iFolder client and server use RSA encryption to pass the username and password. The data is encrypted by the client and kept on the server with blowfish encryption based on the settings we added above. The data is never encrypted on the user's local hard drive. The user's pass phrase is the "shared key" for this encryption. When the user logs in the first time, they will be prompted to enter a pass phrase for this encryption. Both the user id with a current password and the pass phrase are necessary to access the data on the server.

Being able to recover the pass phrase serves a couple of purposes. Of course it is important to end users who might forget their pass phrase and need access to their data. It is also important to the organization because we don't want the data to be inaccessible to the organization should an employee be terminated or leave for other reasons. Set the global recovery pass phrase by clicking on the "Update" button for the "Security Pass Phrase" and entering your chosen pass phrase for the system. It should be complex and not easily guessed.

9.3. Admin Settings – Now we will take a look at how to set authorized administrators. Click the "Admin Names" link in the left frame and then enter a list of users who should have admin rights over the iFolder system as described on the page. This would be a good opportunity to set up some privilege separation to protect the user data. If you have enough staff it is suggested that the administrators who can administer the iFolder server settings and recover pass phrases not also be able to reset the user's LDAP/eDirectory passwords. Separation of privileges can help make iFolder a very secure but manageable way to store company data

9.4. Other things to consider – iFolder can be very convenient for users and provide an elegant method for backing up data on remote laptops or employee's home PC's. It also encourages the flow of data out of the organization by making it more possible to keep this data safe. Any company data outside of the firewall puts that data at risk. A laptop can be stolen or lost. To help protect against this problem encrypting file systems can be used on laptops but this solution can result in a user created "Denial of Service" condition as many systems Administrators know all too well. Files can also be downloaded to uncontrolled computers with just a web browser. Non-savvy users can easily leave traces or full copies of this sensitive data on these uncontrolled computers. Even with these caveats in mind, I feel that iFolder is still a good security benefit to the Enterprise. Much of this data is already flowing in and out of the organization, usually via insecure methods such as e-mail or FTP servers. Putting a solution in place that is relatively secure and automatic certainly adds value.

10. **Non-secure default access –** Some of the programs available on the server do not automatically redirect users to use HTTPS. This is unfortunate because by default user id's and passwords would then be in danger of being captured by an attacker since they are transmitted in the clear over the Internet. We definitely want to change this default behavior so we will make a few more configuration changes below.

10.1. NetStorage SSL redirect – NetStorage is the first application we will address here. In order to automatically redirect users to the SSL version of the application, we will edit the conf file that is used to add it to the main Apache configuration /etc/opt/novell/httpd/conf.d/netstorage.conf. The lines in bold below are what should be added. This configuration employs the Apache “rewrite” directive.

```
Alias /NetStorage "/opt/novell/netstorage/webapp"
<Directory "/opt/novell/netstorage/webapp">
   Options +MultiViews
   AllowOverride None
      Order deny,allow
   Allow from all
</Directory>
# The following lines redirect NetStorage to using https if it is not started that way
<IfModule !mod_rewite.c>
    LoadModule rewrite_module modules/mod_rewrite.so
</IfModule>
<IfModule mod_rewrite.c>
    RewriteEngine on
    RewriteCond %{SERVER_PORT} !^443$
    RewriteRule ^/NetStorage(.*)$ https://%{HTTP_HOST}/NetStorage [L]
</IfModule>
```

10.2. eGuide SSL redirect – eGuide is another application that must be adjusted to make this redirect happen. The same basic configuration is employed as shown below in the bold portion of this excerpt of the /etc/opt/novell/httpd/conf.d/eGuide-apache.conf file.

```
Alias /eGuide "/var/opt/novell/tomcat4/webapps/eGuide"
<Directory "/var/opt/novell/tomcat4/webapps/eGuide">
   Options FollowSymLinks
   Order allow,deny
   Allow from all
   AllowOverride All
</Directory>

# The following lines redirect eGuide to using https if it is not started that way
<IfModule !mod_rewite.c>
    LoadModule rewrite_module modules/mod_rewrite.so
</IfModule>
<IfModule mod_rewrite.c>
    RewriteEngine on
    RewriteCond %{SERVER_PORT} !^443$
    RewriteRule ^/eGuide(.*)$ https://%{HTTP_HOST}/eGuide [L]
</IfModule>
```

Ongoing Maintenance Procedures/Policies

Now that the server is relatively secure it is very important to maintain it in such a way that it will remain secure. This section details procedures and policies that will help to ensure that this server remains serviceable and at the height of security for a long time to come. Components that will be addressed include backups, log monitoring and analysis, patching, and monitoring file/system integrity.

1. **Backups –** Backups of the data and the system are important for several reasons. Of course the user data saved within the applications is important, but an initial full system level backup is also quite important.
   1.1. Initial "Gold" backup – It is important to get a "snapshot" of the system before it is put into production. This can help for a couple of situations. If the system is ever compromised you have a base image to work from to help identify the method of intrustion and exact damage done. It can also be useful if you need to "clone" the system and put several others like it into production. This can significantly cut down the work load. The "gold" image should be put on tape and then locked away in a secure place and the tape should not be used again. (Caution: tape media that isn't "exercised" on a periodic basis will lose it's data or become unreliable within 12 months or shortly thereafter.) The commands shown below are one way of doing this and show the preferred method of backing up to a directly attached device instead of relying on some sort of remote access method across the network. A remote access method would introduce another path/application that must be secured.

```
#  mt –f /dev/st0 rew
# tar –cpMf /dev/st0n / --exclude=/proc
```

   The "mt" command line is used to rewind the tape device we will use. The tar command is creating (c) and archive, keeping permissions entact (p), and preparing to span multiple tape Media devices if necessary (M). We are also using the "no rewind" tape device "st0n" and excluding the proc file system since it does not contain real files anyway.

   1.2. Incremental "Gold" backups – Whenever major changes or patches are applied to the system the full system backup should be done again to document the steps along the way and have a more realistic baseline to analyze with and step back to if necessary.

   1.3. Data backups – The user data and the functionality of the server are some of the most important things we are protecting throughout this process so it would be foolish not to provide a method of backing up the user data. In the system we have built there are a few specific areas where user data is kept for each of the applications. These areas should be backed up using an incremental rotation method. The directories are listed below along with a resource for setting up automatic tar backups.
      1.3.1. iFolder - /var/opt/novell/ifolderdata

1.3.2. NetStorage - /var/opt/novell/netstorage

1.3.3. Virtual Office - /var/opt/novell/iManager/nps/WEB-INF/communityStore.

1.3.4. Backup Scheme – A good resource for using tar to do incremental backups that includes a good example scheme is "Securing and Optimizing Linux: RedHat Edition -A Hands on Guide" (http://pierre.mit.edu/compfac/linux/Securing-Optimizing-Linux-RH-Edition-v1.3/chap29sec305.html). This should be followed closely but in a manner integrated with your current enterprise backup strategy.

1.4. What about the other data? – Beside the user data on this server we also have other data that will change daily. I am specifically choosing not to back up this data because the data is replicated and backed up elsewhere. Logging data is sent to a central Syslog server and that server has it's own backup strategy that is implemented to protect it. The eDirectory database files do not need to be backed up on this server because that data is replicated with at least 2 other servers in the eDirectory tree. One of those servers is specifically responsible for backups of the eDirectory tree and is optimized for this task.

2. **Logging –** Effectively monitoring and processing log files is very important if we are ever to know about an intrusion attempt that has taken place or been successful. We covered log rotation above and sending logs to a central syslog server, but how do we effectively monitor that mountain of data? Two of the best solutions out there are "Logsurfer" (http://www.cert.org/security-improvement/implementations/i042.02.html) or "LogWatch" (http://www2.logwatch.org:81/) Setting up this central syslog server is beyond the scope of this document, but essentially what these programs do is offer semi-IDS functionality for analyzing log files from different systems. Logsurfer is especially adept at this in more of a realtime manner. Other "pay" systems from Symantec, ISS, and others are also available which can do the same basic thing. Whichever product is used, it is very important that ongoing, effective log file monitoring is planned for and executed.

3. **Tripwire and system Integrity –** Tripwire is a tool that can check the integrity of key files and directories on the system for evidence of tampering or changes. It is available in a free version as well as a more richly featured "pay for" version. We will set up and configure the free version for our purposes here. First we create the configuration file.
   3.1. tw.config – I got most of this from a config file created by Jay Beale and posted at http://networking.earthweb.com/netsecur/article.php/10952_624581_2. I modified it to fit this environment and saved it to /etc/tw.config. Below is what I came up with for reference.

```
# This file is a sample configuration file that you can use for a
# stock system, using Tripwire 1.3.1.

# REFERENCE CHART follows - each letter/number corresponds to a
# different quality of the file to watch.
#
#   p    - Permission and file type (mode)
#   i    - inode number (inode=entry in the filesystem)
#   n    - link count (number of hard links to the file)
#   u    - UID (owner of the file)
#   g    - GID (group owner of the file)
#   s    - size of file
#   a    - access timestamp (last time the file was accessed (RARELY USED!)
#   m    - modification timestamp (last time the file was modified)
#   c    - inode Change timestamp (last time the inode was modified)
#   0-9  - "signature" algorithm to use: 1=md5, 2=snefru, 7=SHA-1
#   E    - Ignore everything

#
# First, define a number of monitoring levels.
#

# Essential system binaries should be monitored on all attributes, with a
# high level of certainty.  We keep only md5 and SHA-1 for now.

@@define BIN E+pinugsmc17

# System logs should be allowed to change, and even to switch inode numbers.
# The inode modification is because of automatic log cycling.

@@define LOG E+pnug

# Device files should simply maintain ownership, permissions and such.
# It doesn't make sense to monitor contents.  We also ignore inode
# mod (c) because this changes every reboot.

@@define DEV E+pnug

# Essential system config files (/etc/fstab, /etc/hosts.allow) should
# be watched very closely.

@@define CONF E+pinugsmc17

# Most directories need to allow for new files to be added, so we
# won't watch size, mod time, changes to the inode, or compute sigs.

@@define DIR  E+pinug

#
# Main configuration starts here...
#

# Monitor the root directory itself, but don't recurse into it.

=/      @@DIR

# Monitor essential system binaries: libraries and programs.
```

```
/bin    @@BIN
/lib    @@BIN
/sbin   @@BIN
/usr    @@BIN
/opt    @@BIN
# Monitor the /boot directory, where the kernel et al. is stored.
# System.map changes inode and mod time on every reboot, so ignore
# these.
/boot                   @@BIN
/boot/System.map*        @@BIN-mc
# Monitor /dev, the devices directory.
/dev    @@DEV
#
# Granularly, watch the system's config files...
/etc    @@CONF

# mtab holds current mounted volume information.  Usually, we should
# treat this as a log, since it must change.
/etc/mtab       @@LOG

# passwd and shadow will change on any system with many users, since
# you'll  be adding users regularly, changing your passwords...  If
# you're on a system with few users, watch /etc/passwd more closely.
# We are watching these closely because there are very few local users.
/etc/passwd     @@CONF
/etc/shadow     @@CONF

# /home should change often.  We can simply watch the directory itself,
# but we have to allow for new directories to be created within.
=/home          @@DIR

# lost+found can be watched as a directory with no monitoring of contents
=/lost+found    @@DIR

# mnt and media contain the system mount points for CD-ROM, floppy,...  Barely
# watch these...
=/mnt           @@DIR
=/media          @@DIR

# The proc filesystem is special and changes a great deal.
=/proc          @@DIR

# /root is root's home directory.  We don't generally watch the contents,
# but you can choose to do this if you're careful enough when logging in
# as root.
=/root          @@DIR

# /tmp should change often and greatly.
=/tmp           @@DIR

# /var is difficult, as it contains logs, mail queues, and mailboxes, to
# name a few type of files.
=/var                   @@DIR
/var/log                 @@LOG
=/var/spool             @@LOG
/var/spool/cron         @@LOG
/var/spool/clientmqueue @@LOG
/var/spool/mail         @@LOG
!/var/lock
```

3.2. Create the Tripwire database – The first time you run tripwire, you should use the –initialize switch to create the database and establish your baseline file signatures. Once it is finished it leaves you with a message to copy the database file and the configuration file to safe media (i.e. writable CDROM) to protect them from tampering. This way you know you have clean copies when you need them.

3.3. Run a Tripwire check – After the database has been created and copied to "safe" media, the way that you would use the tool to check the system would be by specifying the path to the configuration and database files off of the CD with a command similar to the following

```
# tripwire –d /media/cdrom/tw.db_<hostname> -c /media/cdrom/tw.config
```

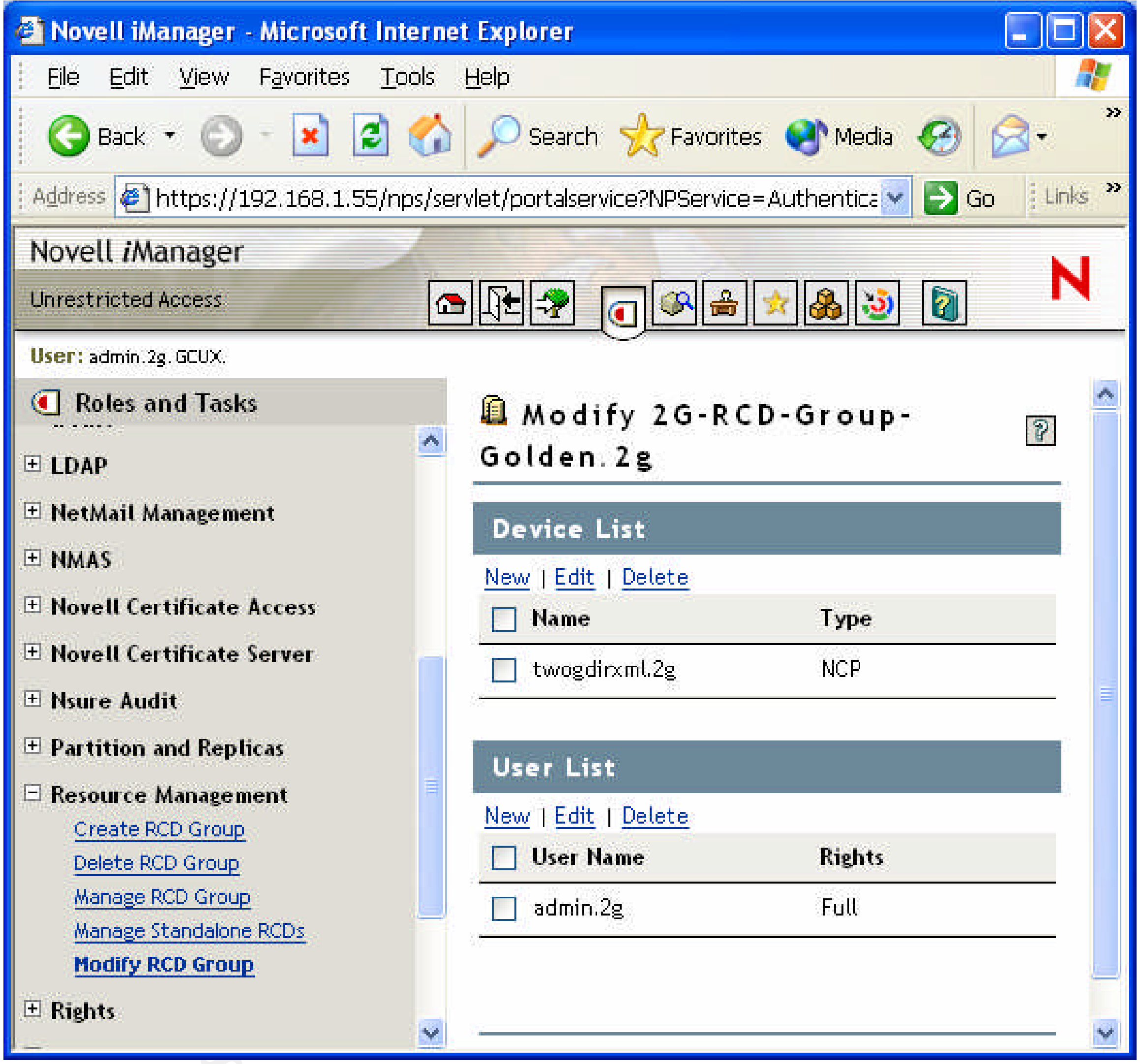
3.4. Run Tripwire automatically – It is good practice to run tripwire on a daily or weekly basis. This is harder to accomplish with the free version but still can be done. The basic technique would be to redirect the output from the command above to a file in root's home directory with a cron job and then have another cron job pick up that file and e-mail it to you. Logsurfer might also be used to monitor the output and notify you only if something changes.

4. **Patching –** Keeping the system patched is a very important part of ongoing maintenance and keeping the system secure. For this system we have two main avenues for patching the system. This could be reduced to one method (Ximian Red Carpet Enterprise, http://www.ximian.com/) which might be a good idea especially if you have many NNLS and/or Linux servers to patch.

4.1. YOU (Yast Online Update) – This is the automatic update method included with SLES8. It identifies and patches only the components of the OS and supporting files included with SLES8. It does not address patching of the NNLS components. However if you don't want to pay for a Red Carpet license this is still a very useful way of maintaining your patch level on your server. This process was covered as part of the OS installation above and will not be addressed again here.

4.2. Red Carpet Patch Management – Patch management for NNLS is built into the NNLS install and points automatically to a Red Carpet server located at Novell for patches to the NNLS code. This patch management does not cover the OS files so you would have to set up or subscribe to a service with full Red Carpet patch management for both if you want to cover everything in on solution. Red Carpet updating can be performed from the command line, but iManager includes the ability to group and manage multiple servers from one management interface. This functionality is fully documented on Novell's website (http://www.novell.com/documentation/nnls/index.html?page=/documentation/nnls/install/data/bnougfv.html#bnougfv) but I am including a few specific configuration options that are especially important/significant.

Once you have the RCD Group(s) defined you can manage the whole group of servers simultaneously. Below is a screen print showing the membership of an RCD group.

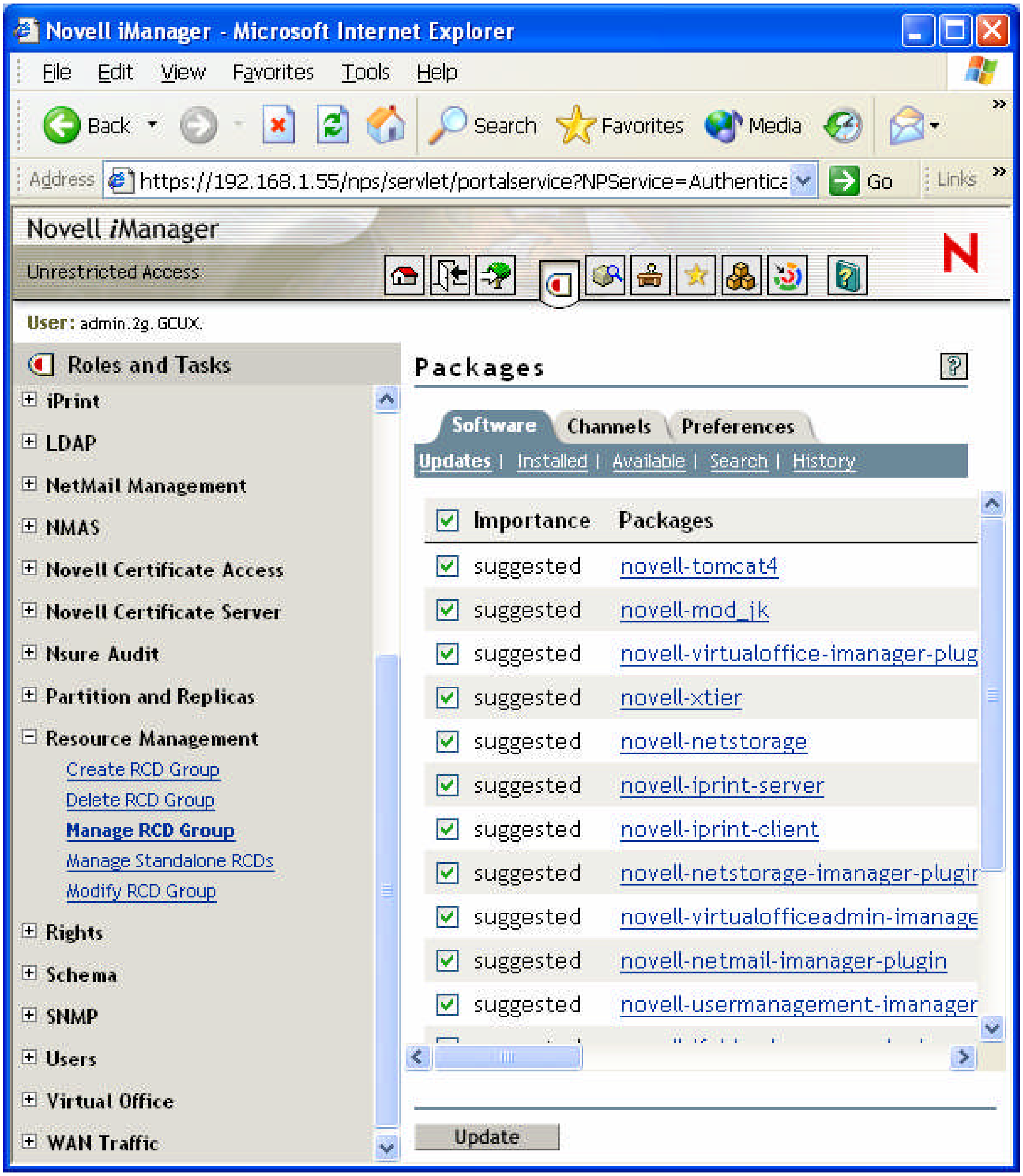


Name	Type
twogdirxml.2g	NCP

User Name	Rights
admin.2g	Full

As you can see in this example you can add multiple devices to any RCD group as well as add users who can administer this RCD group. Groups should be organized around similar installations so that the whole group can be subscribed to specific update channels. Below is a screen print showing the default channel subscribed to upon installation of NNLS. This is where you could add other channels if the Red Carpet Daemon was connected to a different Red Carpet Enterprise server with additional channels available.

In the “Software” tab you can look at available software and also at the updates that are available for installed software. This is where you would select the updates (usually all of them) and actually perform the update. You should come back here on a scheduled basis (at least weekly) to look for and apply updates. Keep in mind that each update should be researched before applied in a production environment. For now we will select all updates and apply them.

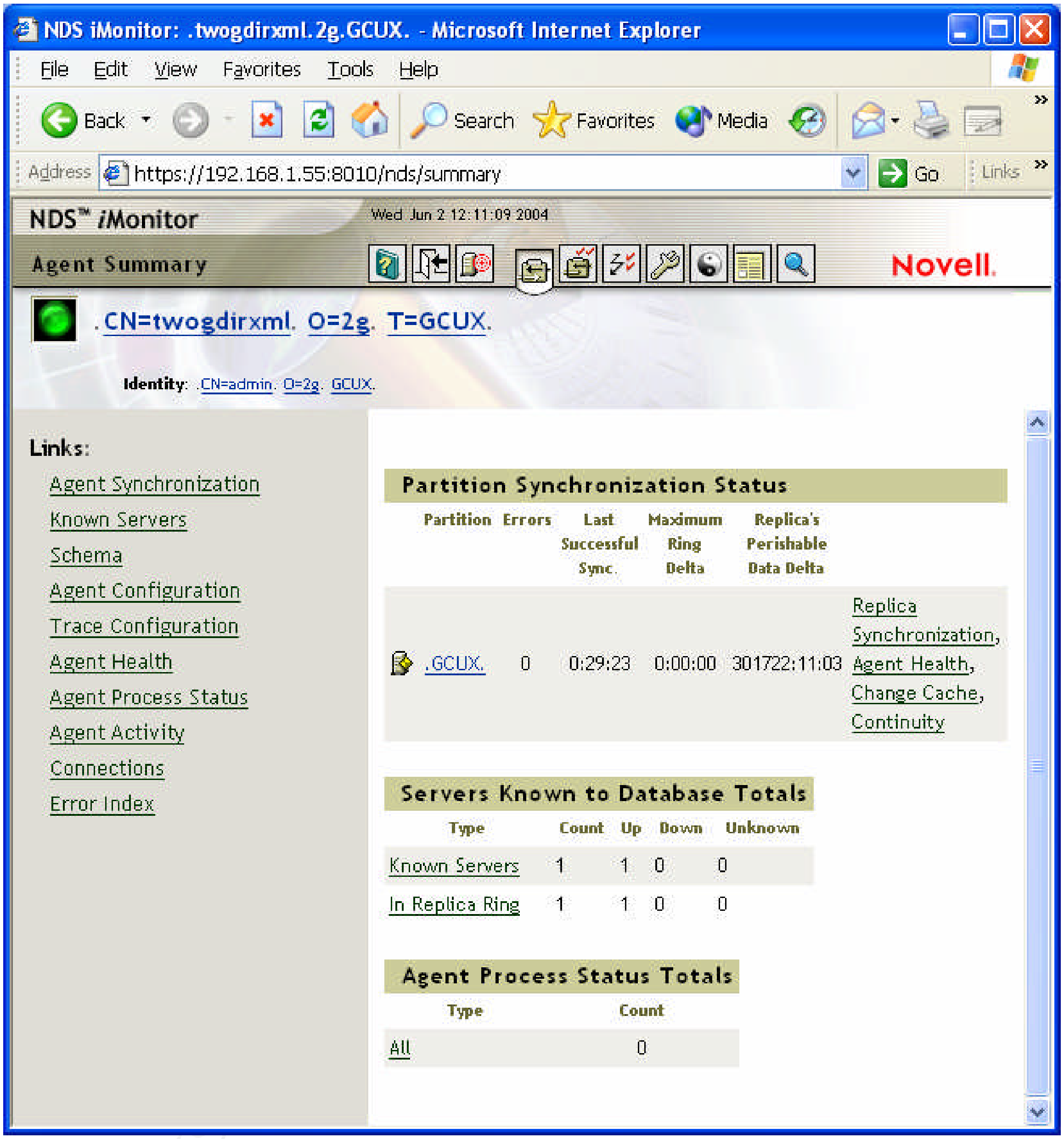

Novell iManager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address https://192.168.1.55/nps/servlet/portalservice?NPService=Authentica Go Links
Novell iManager
Unrestricted Access
User: admin.2g.GCUX.
Roles and Tasks
iPrint
LDAP
NetMail Management
NMAS
Novell Certificate Access
Novell Certificate Server
Nsure Audit
Partition and Replicas
Resource Management
Create RCD Group
Delete RCD Group
Manage RCD Group
Manage Standalone RCDs
Modify RCD Group
Rights
Schema
SNMP
Users
Virtual Office
WAN Traffic
Packages
Software Channels Preferences
Updates | Installed | Available | Search | History
Update

☑	Importance	Packages
☑	suggested	novell-tomcat4
☑	suggested	novell-mod_jk
☑	suggested	novell-virtualoffice-imanager-plug
☑	suggested	novell-xtier
☑	suggested	novell-netstorage
☑	suggested	novell-iprint-server
☑	suggested	novell-iprint-client
☑	suggested	novell-netstorage-imanager-plugir
☑	suggested	novell-virtualofficeadmin-imanage
☑	suggested	novell-netmail-imanager-plugin
☑	suggested	novell-usermanagement-imanager

4.3. Automated Patch Management – The “you” or “rug” commands could be added to cron jobs to automate installation of available patches but I do not recommend that method. Patches installed blindly can result in more security problems or availability problems than not patching at all in some cases. I feel that the approach of putting all of the patching into a management interface that can control multiple servers such as the iManager interface we have explored is the best method for managing patching because it allows a seasoned administrator to make appropriate decisions and do necessary research and testing before applying patches in a production environment.

4.4. Patch Testing – Before putting any patches into production it is critical that testing be done to be sure the patch performs as expected but also to be sure that it didn't change the permissions or configuration settings you have made that secure the server. Appropriate testing strategies are discussed in more detail in the "Testing" section of this paper below.

5. **eDirectory Maintenance –** Since the directory is very much at the core of the functioning of the NNLS products we have installed, it is also important to put in place some ongoing maintenance procedures for eDirectory. We have already discussed doing regular backups of the database so I won't address that again now.
5.1. iMonitor – The main interface for doing all of the maintenance we will discuss is the web based iMonitor interface. Below is a screen print of the default page seen after connecting to https://192.168.1.55:8010/_LOGIN_SERVER_ and logging in. Notice the green status indicator on the upper left. This is a quick view that indicates whether any errors are currently being generated and whether or not other health parameters are being met.



Wed Jun 2 12:11:09 2004

.CN=twogdirxml. O=2g. T=GCUX.

Identity: .CN=admin. O=2g. GCUX.

Partition Synchronization Status

Partition	Errors	Last Successful Sync.	Maximum Ring Delta	Replica's Perishable Data Delta	
.GCUX.	0	0:29:23	0:00:00	301722:11:03	Replica Synchronization, Agent Health, Change Cache, Continuity

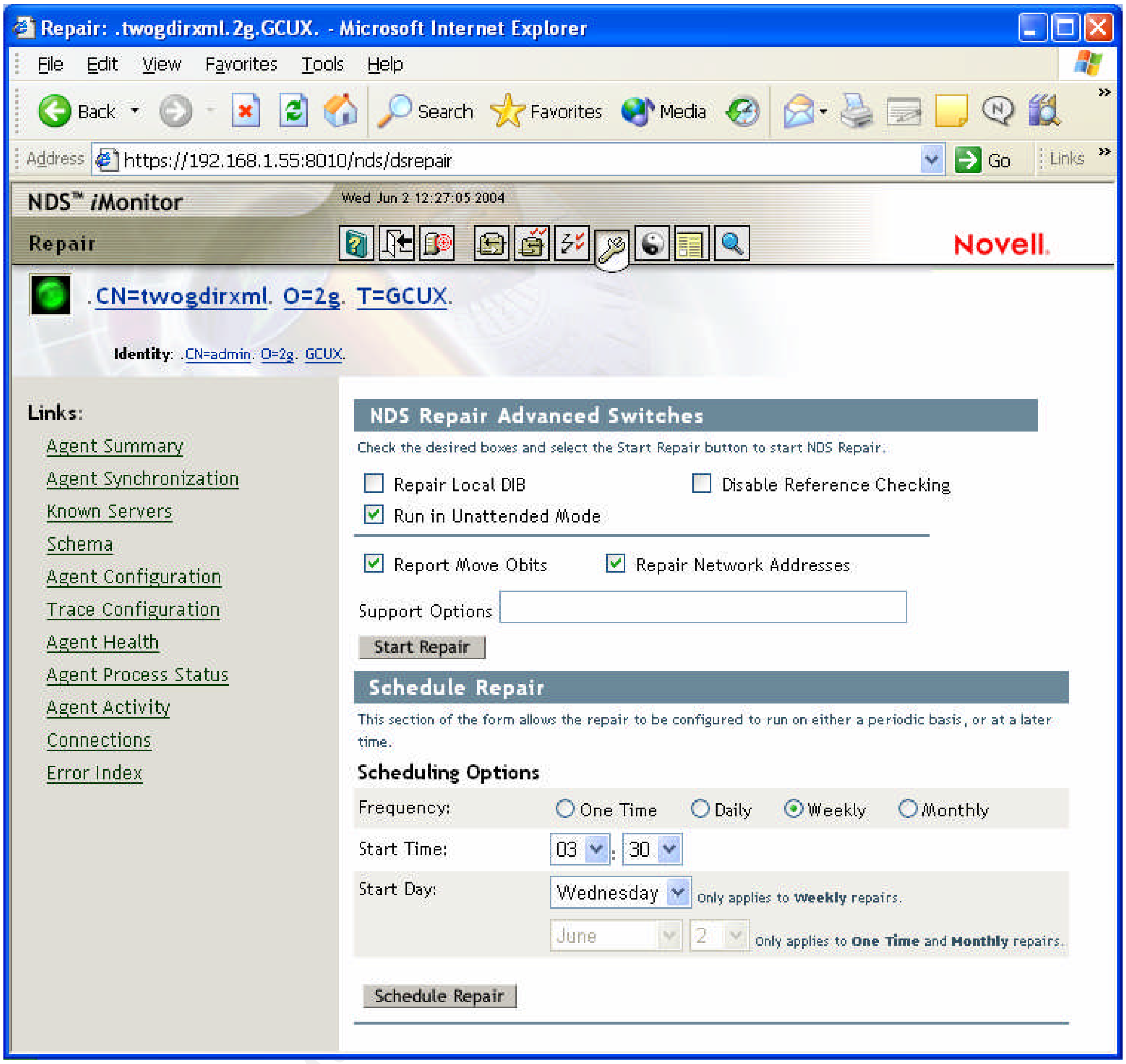
Servers Known to Database Totals

Type	Count	Up	Down	Unknown
Known Servers	1	1	0	0
In Replica Ring	1	1	0	0

Agent Process Status Totals

Type	Count
All	0

5.2. Repairing the database – Now we will use iMonitor to run and schedule periodic database repairs. This will help the server and the whole eDirectory database to continue to perform up to expectations and needs. To access the repair screens, click the image of the wrench on the top navigation bar. Then click the “Advanced” button to see the screen shown below.



NDS Repair Advanced Switches

Check the desired boxes and select the Start Repair button to start NDS Repair.

☐ Repair Local DIB ☐ Disable Reference Checking
☑ Run in Unattended Mode

☑ Report Move Obits ☑ Repair Network Addresses

Support Options

Schedule Repair

This section of the form allows the repair to be configured to run on either a periodic basis, or at a later time.

Scheduling Options

Frequency:	○ One Time ○ Daily ● Weekly ○ Monthly
Start Time:	03 : 30
Start Day:	Wednesday Only applies to **Weekly** repairs. June 2 Only applies to **One Time** and **Monthly** repairs.

The selections made in this screen should work well for most environments. We are scheduling the repair for once a week at 3:30 a.m., running in unattended mode, checking obituaries, and repairing network addresses. To schedule this repair click on the “Schedule Repair” button. Repair results will be found in the reports screen (second link from the right in the top navigation bar). This should be monitored weekly after the repair has been run for errors or problems found. Other automated tools are also available for eDirectory maintenance and monitoring from companies such as NetPro (www.netpro.com).

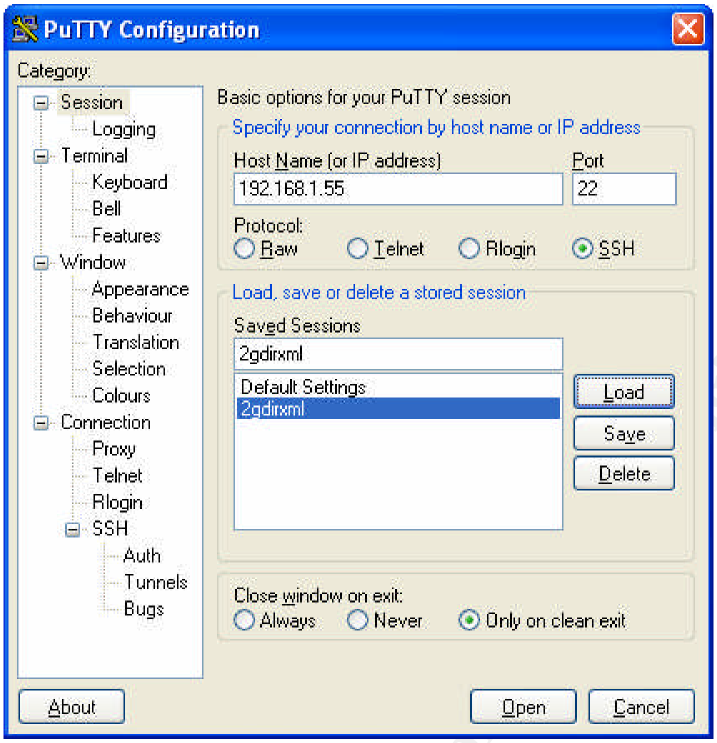
6. **Auditing the network/server –** Another ongoing maintenance procedure that should be performed is conducting periodic security audits to confirm that new vulnerabilities or problems have not been introduced. This should typically be done by an external entity which wouldn’t have a vested interest in showing that the system was secure. The methods used by an external entity are also less likely to be clouded by previous knowledge. Some of the typical steps for

this audit are discussed below in the “Testing and Verifying” section.

Testing and Verifying

Before putting the system into production it is important to test it to see that it performs as expected from a security standpoint. As instructed above, we have been testing the functionality of the system after all configuration sections and probably fixed some things along the way. Now we need to test to be sure that the server is as secure as we expect it to be and if we find problems, we will have to analyze those and loop back to see how those problems can be fixed or whether or not we will have to live with the risks we have identified.
To do this testing we will perform some manual checks and employ a vulnerability scanning tool. Here are the security testing steps I suggest based on the risks identified in the “Risk Mitigation Plan” section above. I do not show all of the procedures and output for every step but instead have included links to documentation to help you conduct those steps.

1. **Section A – Network Access Checks –** In this section we will verify that the network access is both limited and allowed in the manner that we specified.
   1.1. Login allowed with SSH, banners present – We specified that login with SSH should be allowed from the internal network but not for root. To test this we will use PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/)to attempt login and verify banners are displayed.
   1.1.1. PuTTY config screen – included here for reference is the main screen of the PuTTY configuration showing the connection attempt using SSH.



Host Name (or IP address): 192.168.1.55
Port: 22

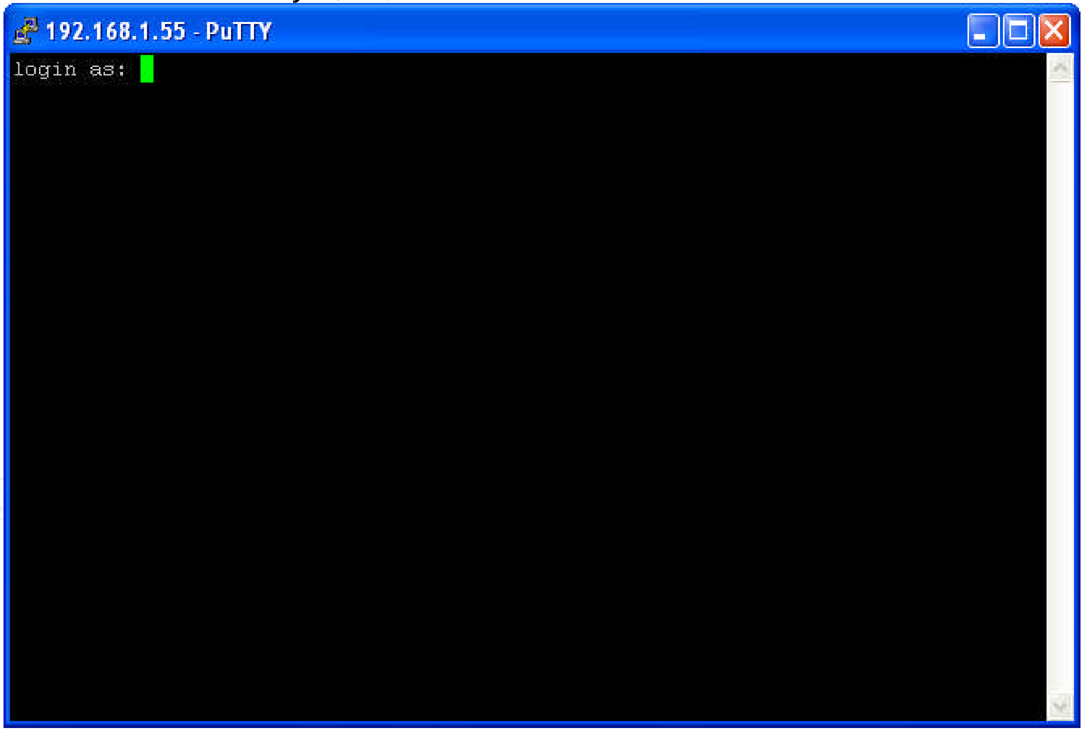
Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Saved Sessions: 2gdirxml

Default Settings
2gdirxml

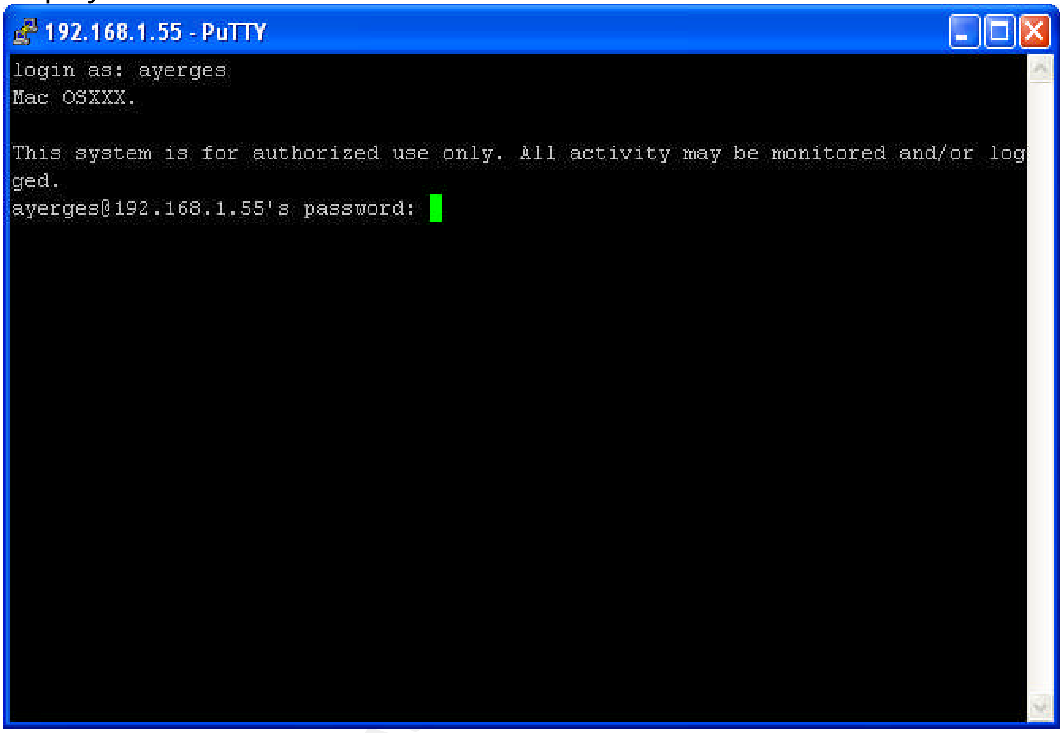
Close window on exit: ○ Always ○ Never ● Only on clean exit

1.1.2. Initial connection with PuTTY – Here is the initial connection screen. No banners yet, but that is normal.



```
login as: 
```

1.1.3. After putting in a userid – Now we see that the OS identification has been properly obfuscated and we see the simple warning displayed as it should be.

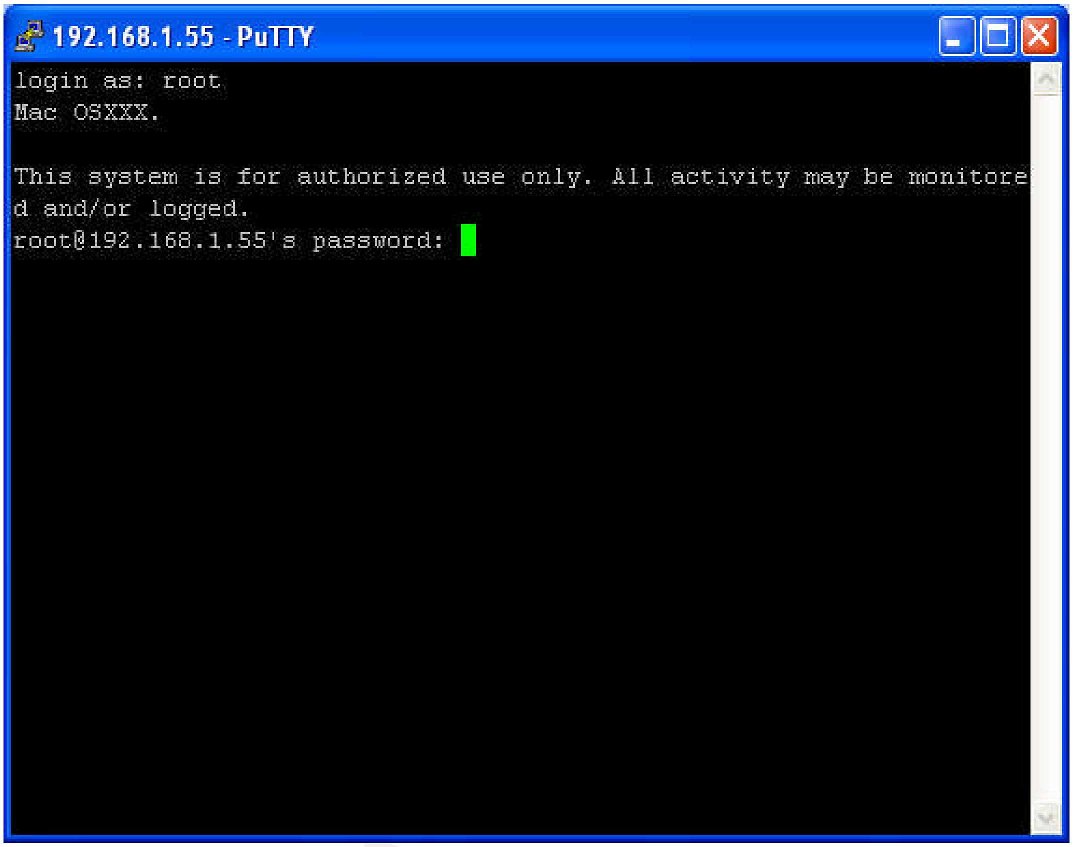


```
login as: ayerges
Mac OSXXX.

This system is for authorized use only. All activity may be monitored and/or log
ged.
ayerges@192.168.1.55's password:
```

1.2. Root Login denied even via SSH – We indicated that root should not be able to log in to the terminal even through SSH directly. This improves audit trails and security of the system. Now we need to test it. Again we will connect with PuTTY.

1.2.1. Putty connected "root" id entered – The root id is allowed? What is the problem, I thought we disabled this? Actually the program must accept the id being entered to know who is logging in.

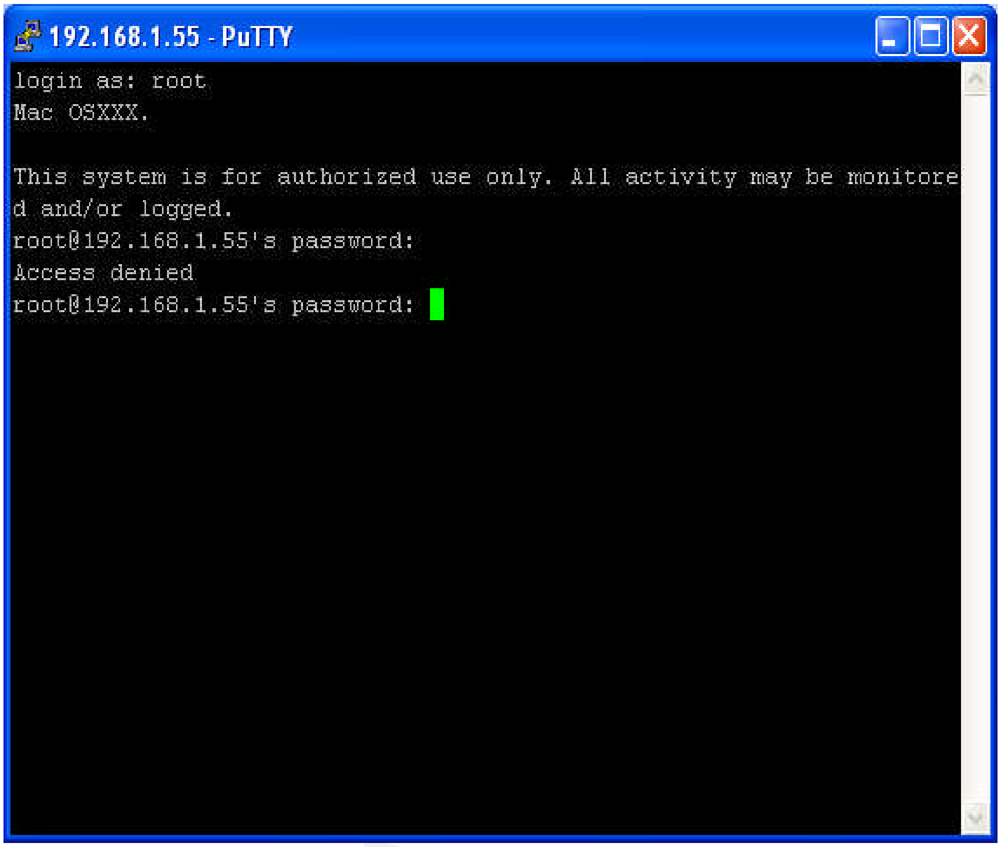


```
login as: root
Mac OSXXX.

This system is for authorized use only. All activity may be monitore
d and/or logged.
root@192.168.1.55's password:
```

1.2.2. Root denied – Now that is better. Root indeed is denied access to the server even over SSH. The first entry is from the /var/log/secure log file which shows that root access is indeed denied. The print below shows the message on the screen.

```
Jun  2 14:31:04 twogdirxml sshd[3319]: ROOT LOGIN REFUSED
FROM ::ffff:192.168.1.100
```



```
login as: root
Mac OSXXX.

This system is for authorized use only. All activity may be monitore
d and/or logged.
root@192.168.1.55's password:
Access denied
root@192.168.1.55's password:
```

1.3. Login blocked from external networks – We also specified that login should not be allowed over telnet or from external networks. I had a colleague check this with me from the outside using standard tools such as PuTTY again or just the command line telnet. The sessions just hung as expected and this is what I saw in the /var/log/kernel log file which is where the iptables firewall logs are being sent.

```
Jun  2 14:43:14 twogdirxml kernel: IN=eth0 OUT=
MAC=00:0f:1f:47:0f:ce:00:04:5a:fd:a1:f9:08:00 SRC=class.c.addr.200
DST=192.168.1.55 LEN=60 TOS=0x00 PREC=0x00 TTL=47 ID=13817 DF
PROTO=TCP SPT=40584 DPT=23 WINDOW=5840 RES=0x00 SYN URGP=0
Jun  2 14:43:14 twogdirxml kernel: IN=eth0 OUT= MAC=00:0f:1f:47:0f:ce:00:04:5a:f
d:a1:f9:08:00 SRC=class.c.addr.200 DST=192.168.1.55 LEN=60 TOS=0x00
PREC=0x00 TTL=47 ID=13817 DF PROTO=TCP SPT=40584 DPT=23
WINDOW=5840 RES=0x00 SYN URGP=0
Jun  2 14:43:52 twogdirxml kernel: IN=eth0 OUT=
MAC=00:0f:1f:47:0f:ce:00:04:5a:fd:a1:f9:08:00 SRC=class.c.addr.200
DST=192.168.1.55 LEN=60 TOS=0x00 PREC=0x00 TTL=47 ID=35230 DF
PROTO=TCP SPT=41362 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Jun  2 14:43:52 twogdirxml kernel: IN=eth0 OUT= MAC=00:0f:1f:47:0f:ce:00:04:5a:f
d:a1:f9:08:00 SRC=class.c.addr.200 DST=192.168.1.55 LEN=60 TOS=0x00
PREC=0x00 TTL
=47 ID=35230 DF PROTO=TCP SPT=41362 DPT=22 WINDOW=5840 RES=0x00
SYN URGP=0
```

2. **File System checks –** We implemented several very specific file system protections. We now need to check these to verify that they are working properly.

   2.1. Mount options – We changed fstab to indicate some specific mounting options. Now we need to check to see if they are being honored and working as we expect.

   2.1.1. First we execute the "mount" command – This should show us the current state of the mounted file systems and show us if the options we specified are being honored. First I will mount a floppy disk and a CDROM so that those will show up as well.

```
# mount
/dev/hda1 on / type ext3 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda2 on /usr type ext3 (ro,nodev)
/dev/hda6 on /var type ext3 (rw,nosuid,nodev)
shmfs on /dev/shm type shm (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
/dev/fd0 on /media/floppy type vfat (rw,noexec,nosuid,nodev,sync)
/dev/hdc on /media/cdrom type iso9660 (ro,nosuid,nodev)
```

The options are shown in parenthesis at the end of each line and, yes, they do correspond to our configuration plan. Now we need to further test this.

2.1.2. Test that floppy mount honors noexec flag – In order to test this, we will copy an executeable script to the floppy drive and then try to execute it. Here are the commands and the results.

```
# cp /etc/init.d/novell-httpd /media/floppy
# ls -l /media/floppy
total 45
drwxr--r--    2 root     root         7168 Jun  2 15:31 .
drwxr-xr-x    4 root     root         4096 Oct 21  2002 ..
-rwxr--r--    1 root     root        13440 May 13 05:48 301849089.nfk
-rwxr--r--    1 root     root        14547 May 13 05:47 301849089.nlf
-rwxr--r--    1 root     root         3248 Jun  2 15:31 novell-httpd
# /media/floppy/novell-httpd force-reload
-bash: /media/floppy/novell-httpd: /bin/sh: bad interpreter: Permission denied
```

2.1.3. Test deletion of a binary on /usr file system – With the /usr file system mounted as read only (ro) even root should not be able to delete a file or copy a file to the system. To test this, I will attempt to delete a file. Here are the commands I used and the results. As expected, I could not delete it.

```
# rm -f /usr/local/apache2/lib/libapr-0.so.0.9.5
rm: cannot remove `/usr/local/apache2/lib/libapr-0.so.0.9.5': Read-only file
system
#
#
```

2.1.4. Test nodev functionality on /var – To test this functionality, we will create a “null” device on the /var file system and then try to send data to it. We should be denied access when we try to copy data to this “device.” Below are the commands and the results. This is indeed denied.

```
# mknod -m 666 /var/opt/null c 1 3
#
# ls -l /var/opt
total 12
drwxr-xr-x    3 root     root         4096 Jun  2 16:11 .
drwxr-xr-x   18 root     root         4096 May 17 11:56 ..
drwxr-xr-x   14 root     root         4096 May 30 01:01 novell
crw-rw-rw-    1 root     root       1,   3 Jun  2 16:11 null
# cat 888888 > /var/opt/null
-bash: /var/opt/null: Permission denied
# cp /var/opt/novell /var/opt/null
cp: omitting directory `novell'
```

2.1.5. Test nosuid functionality – To test this functionality we will copy a functioning suid executeable to /var/opt and then try to use it. The /bin/su binary is one of the few left on the system after hardening it so we will use that. Below are the commands I used and the results of those commands.

```
# cp /bin/su /var/opt
# ls -l /var/opt/su
-rws------   1 root     root        28157 Jun  2 16:41 su

Note that since we had set the default umask to 077 we have to chmod this to test
it.

# chmod 4755 /var/opt/su
# ls -l su
-rwsr-xr-x    1 root     root        28157 Jun  2 16:41 su

Now have a non privileged user test it.
:~> cd /var/opt
:/var/opt> ./su -
Password:
./su: incorrect password
```

The message that you see isn't obviously an SUID problem, but I tested it several times and the password was indeed correct. If we look at the /etc/shadow file again, we can see why this would happen. The "su" binary must be SUID to root in order to read the /etc/shadow file. In this case both binaries have that bit set, but the one in /var/opt/ will not work because the file system is mounted "nosuid." For reference, here are the permissions set on the /etc/shadow file.

```
-rw-r-----    1 root     shadow        554 2004-05-18 13:43 shadow
```

3. **Review Running processes –** We need to look carefully at all running processes to be sure that only the open ports we planned to have open are listening. This should be reviewed periodically to ensure that no new processes have been started maliciously or inadvertently. We will first use netstat and then lsof to look at these. The commands and results are below.

   3.1. TCP Ports - First we will look at all listening tcp ports. Then later we will look at any that have not been identified previously to see what processes have the ports open.

```
# netstat -ntl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:8008          0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.55:8008       0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:8010          0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.55:8010       0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3019          0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.55:427        0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:427           0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.55:524        0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:524           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:8018            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:8020            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:505             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:636             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:8005          :::*                    LISTEN
tcp        0      0 :::8009                 :::*                    LISTEN
tcp        0      0 :::80                   :::*                    LISTEN
tcp        0      0 :::8080                 :::*                    LISTEN
tcp        0      0 :::22                   :::*                    LISTEN
tcp        0      0 :::631                  :::*                    LISTEN
tcp        0      0 :::443                  :::*                    LISTEN
```

   Two ports immediately come into question, 3019 and 505.
   3.2. UDP Ports – Now we will see what is open for UDP and then do further investigation.

```
# netstat -nul
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 127.0.0.1:32772         0.0.0.0:*
udp        0      0 0.0.0.0:32773           0.0.0.0:*
udp        0      0 0.0.0.0:32774           0.0.0.0:*
udp        0      0 127.0.0.1:32775         0.0.0.0:*
udp        0      0 192.168.1.55:524        0.0.0.0:*
udp        0      0 255.255.255.255:427     0.0.0.0:*
udp        0      0 192.168.1.55:427        0.0.0.0:*
udp        0      0 239.255.255.253:427     0.0.0.0:*
udp        0      0 192.168.1.55:123        0.0.0.0:*
udp        0      0 127.0.0.1:123           0.0.0.0:*
udp        0      0 0.0.0.0:123             0.0.0.0:*
```

The high ports 32772 through 32775 are interesting but all the rest are already identified and should be listening. We will investigate those high ports further as well as the open TCP ports with lsof now.

3.3. LSOF identification of open ports – Now we will use lsof to find out what those unidentified process are. Below are the commands used and the results. Following the results, is a short discussion of the findings and any actions that might be necessary.

```
# lsof -i TCP:3019
COMMAND  PID   USER   FD   TYPE DEVICE SIZE NODE NAME
idsd    1069 iprint    6u  IPv4   6131       TCP localhost.localdomain:resource_mgr
(LISTEN)

# lsof -i TCP:505
COMMAND PID USER   FD   TYPE DEVICE SIZE NODE NAME
rcd     255 root    4u  IPv4   1085       TCP *:mailbox-lm (LISTEN)

# lsof -i UDP:32772
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
ndsd     974 root   20u  IPv4   5911       UDP localhost.localdomain:32772
---- many processes, truncated here

# lsof -i UDP:32773 | more
COMMAND  PID    USER   FD   TYPE DEVICE SIZE NODE NAME
httpd   1066 novlwww   24u  IPv4  12329       UDP *:32773
---- many processes, truncated here

# lsof -i UDP:32774 | more
COMMAND  PID    USER   FD   TYPE DEVICE SIZE NODE NAME
httpd   1066 novlwww   22u  IPv4  10471       UDP *:32774
---- many processes, truncated here

# lsof -i UDP:32775
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
ndsd     974 root   60u  IPv4  12385       UDP localhost.localdomain:32775
---- many processes, truncated here
```

The first tcp port is easily identified. It is a management port for the iPrint product and is used to manage printers set up on the server. It may have to be opened in the firewall configuration in order to effectively manage the printers but should only be opened to internal hosts.

The second port of tcp 505 is from the Red Carpet Client. This is what allows the remote management of the client and client commands to be sent to the Red Carpet daemon running on the server. This will be needed for remote management and should be added to only the internal firewall configuration.

The UDP ports 32772 seem most likely to be RPC like ports that eDirectory (ndsd) and Apache (httpd) listen on to offer services. These should definitely remain blocked, but probably can't be shut off.

4. **Run a Vulnerability Scan –** The last test that we will conduct before putting this machine into production is a Nessus scan (www.nessus.org) This scan will be conducted from inside the network and will be a pretty aggressive scan in order to stress the system and find any vulnerabilities that might still exist. Nessus was installed on a separate host and the signatures updated before starting. Results of this scan will guide any last hardening steps that need to be taken. I will not include all of the steps or results found but simply summarize what was found and any actions necessary to fix the vulnerability found.
4.1. Two High risk conditions found – Two high risk conditions were found that should be corrected. These are listed below with a brief discussion.
    - **4.1.1. Old open-ssh version** – The Open SSH version installed is a vulnerable one and it should be upgraded to 3.7.1 or later to patch this hole.
        - **4.1.1.1. Firewall hole** – The firewall configuration allows UDP packets to traverse it as long as the source port is 53. This is a common firewall problem that can allow an attacker to communicate to otherwise blocked ports on the server. The firewall should be changed to block this traffic especially from the outside. The host based firewall probably can't be changed since DNS communications are necessary to internal DNS servers.

References

A Sample Firewall Configuration. In *Linux Network Administrators Guide*. Retrieved June 3, 2004, from http://www.faqs.org/docs/linux_network/x-087-2-firewall.example.html

Barnett, R. (n.d.). *Securing Apache Step-by-Step*. Retrieved June 3, 2004, from http://www.cgisecurity.com/lib/ryan_barnett_gcux_practical.html#references

Bates, M. (2004). *Using Apache's RewriteEngine to redirect requests to other URLS and to https://*. Retrieved June 3, 2004, from http://www.whoopis.com/howtos/apache-rewrite.html

Glossary. *UMASK*. Retrieved June 3, 2004, from http://www.linuxquestions.org/questions/glossary.php?s=&glossaryid=34&long=1

Jagjit. (2003) *GNU Linux Security*. Retrieved June 3, 2004, from http://www.freeos.com/articles/4628/

Koconis, D., Murray, J., Purvis, J., Wassom, D. (2003). *Securing Linux: A Survival Guide for Linux Security* (Version 1.0). Bethesda: SANS Press.

Lechnyr, D. (2002). *Network Security with /proc/sys/net/ipv4*. Retrieved June 3, 2004, from http://search.linuxsecurity.com/articles/network_security_article-4528.html

Loza, B. (n.d.). *Build your own IDS with Logsurfer*. Retrieved June 3, 2004, from http://tegosystemonline.com/papers/Logsurfer.pdf

Novell Technical Support Knowledgebase. (2003). *How to configure Dirxml 1.1a with Remote Loader to use SSL*. Retrieved June 3, 2004, from http://support.novell.com/cgi-bin/search/searchtid.cgi?/10083691.htm

O'Neil, P. (2003). *Build your own firewall using SuSE Linux: A mechanics guide* (Version 2.5b). Retrieved June 3, 2004, from http://www.sans.org/rr/papers/index.php?id=1112

Pomeranz, H. (2003). *Securing Unix*. Bethesda: SANS Press.

Project: OWASP Source Code Center: File List. Retrieved June 3, 2004, from http://sourceforge.net/project/showfiles.php?group_id=64424

Ristic, I. (2003). OnLamp.com, *Introduction mod_security*. Retrieved June 3, 2004, from http://www.onlamp.com/pub/a/apache/2003/11/26/mod_security.html

Sharma, K. (n.d.). Linux Security Tips. *Linux Gazette*. Retrieved June 3, 2004, from http://www.linuxgazette.com/issue58/sharma.html

Smith, E. (2001). *Securely Implementing LDAP*. Retrieved June 3, 2004, from http://www.blacksheepnetworks.com/security/resources/securely-implementing-ldap.html

Tomcat Security. Retrieved June 3, 2004, from http://www.jspolympus.com/JSPTOMCAT/TomcatSecurity.jsp

Unofficial SuSE FAQ. *Understanding how SuSEconfig works*. Retrieved June 3, 2004, from http://susefaq.sourceforge.net/faq/suseconfig.html

Walden, C. (2003). Windows-to-Linux roadmap: Part 8. Backup and recovery, *A quick guide to Linux backup and recovery*. Retrieved June 3, 2004, from http://www-106.ibm.com/developerworks/linux/library/l-roadmap8/#resources

Winston, K. (n.d.). Unofficial SuSE FAQ. *SuSE 7.3 Bash Initialization*. Retrieved June 3, 2004, from http://susefaq.sourceforge.net/articles/bash.html

Appendix A – Novell Supplied packages and versions

NOVLpkia-2.7.0-6
NOVLpkit-2.7.0-6
NOVLpkis-2.7.0-6
NOVLsas-8.7.3-34
NOVLsnmp-8.7.3-34
NOVLxis-8.7.3-34
NOVLsubag-8.7.3-34
NOVLnmas-2.3.0-20031205
NOVLlmgnt-8.7.3-34
NOVLstlog-8.7.3-34
NOVLice-8.7.3-34
NOVLembox-8.7.3-16
NOVLjvml-2.0.1-4
novell-base-0.1.1-4
novell-db4-4.1.25.NC-2
novell-virtualoffice-imanager-plugin-1.0-20040129_1710
novell-libldap_c-1.0-6
novell-openssl-0.9.6k-4
novell-httpd-2.0.48-9
novell-httpd-manual-2.0.48-9
novell-j2sdk-1.4.2.01-14
novell-mod_jk-4.1.25-2
novell-mdb-1.0-4
novell-webadmin-4.0.0-30
novell-wa-rcd-1.0.0-30
novell-imanager-2.0.2-17
novell-iprint-management-5.0.20040123-1
novell-ifolder-imanager-plugin-1.0-20040129_1710
novell-netmail-imanager-plugin-1.0-20040129_1710
novell-DXMLplgs-1.1.10-5
novell-plugin-backup-restore-2.0.2-21
novell-plugin-ice-2.0.2-12
novell-plugin-indexmanager-2.0.2-14
novell-plugin-ldap-2.0.2-12
novell-plugin-merge-2.0.2-12
novell-plugin-nmas-2.0.2-14
novell-plugin-pki-2.0.2-13
novell-plugin-repair-2.0.2-12
novell-plugin-rwiz-2.0.2-12
novell-plugin-snmp-2.0.2-12
novell-plugin-service-manager-2.0.2-14
novell-plugin-wanman-2.0.2-14
novell-AUDTauditplugin-1.0.1-20
novell-imgr-rcd-1.0.0-19

```
novell-netstorage-imanager-plugin-3.0.0-21
novell-virtualofficeadmin-imanager-plugin-1.0-20040129_1710
novell-iprint-server-5.0.20040422-8
novell-usermanagement-imanager-plugin-1.0-20040129_1710
novell-webadmin-netmail-plugin-3.5-20040127.1546
novell-nrm-rcd-link-1.0.0-18
novell-xtier-3.0.1-2
```

Appendix B /etc/sysconfig/sysctl.conf

```
# Do you want the "dynamic IP patch" to be enabled at bootup? (yes/no)
#
IP_DYNIP="yes"
#
# Enable syn flood protection (see
/usr/src/linux/Documentation/Configure.help)
# (yes/no)
#
IP_TCP_SYNCOOKIES="yes"
# Runtime-configurable parameter: forward IP packets.
# Is this host a router? (yes/no)
#
IP_FORWARD="no"
#
# Enable Magic SysRq Keys?
#
ENABLE_SYSRQ="no"
#
# DISABLE_ECN
#
DISABLE_ECN="yes"
# Load IPV6???
LOAD_IPV6="no"
# Runtime-configurable parameter: forward IPv6 packets.
#
IPV6_FORWARD="no"
IPV6_PRIVACY="no"
############################################
# The following were added by Al for additional protection.
# IPV4 Startup security options.  Some are probably also in SuSE
firewall but
# I'm not running that so I put them in here with short explanations
and
# edited the boot.ipconfig script to read the options below in this
file.

# Accept source routing on ALL interfaces? "no" disables source
routing.
ACCEPT_SOURCE_ROUTE="yes"

# Syn flood protection enabled. Default number for backlog is set to
4096
# in boot.ipconfig.
SYN_FLOOD_BACKLOG="yes"

# IP Spoofing protection enabled? This setting may drop valid packets.
watch it
RP_FILTER="no"

# Disable ICMP redirects being sent by this machine. If needed change
to "no"
DISABLE_SEND_REDIRECTS="no"
```

```
# Disable receiving ICMP redirects on ALL currently defined interfaces
- yes disables
DISABLE_ACCEPT_REDIRECTS_ALL="no"

# Disable receiving ICMP redirects on newly defined interfaces - yes
disables
DISABLE_ACCEPT_REDIRECTS_DEFAULT="no"
```

Appendix C - /etc/init.d/boot.ipconfig

```
#! /bin/sh
#
# Copyright (c) 2001-2002 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
#
# /etc/init.d/boot.ipconfig
#
### BEGIN INIT INFO
# Provides:          boot.ipconfig
# Required-Start:
# X-UnitedLinux-Should-Start: setserial boot.isapnp boot.sysctl
# Required-Stop:
# Default-Start:     B
# Default-Stop:
# Description:       run ip configuration hooks
### END INIT INFO

. /etc/rc.status
. /etc/sysconfig/sysctl

rc_reset

case "$1" in
  start)
   #
   # Enable "dynamic IP patch"
   #
   if test -n "$IP_DYNIP" -a "$IP_DYNIP" != no -a \
    -e /proc/sys/net/ipv4/ip_dynaddr ; then
    echo -n "Enabling dynamic IP patch"
    case "$IP_DYNIP" in
      yes)    echo 7         ; ECHO_RETURN=$rc_done ;;
      [1-9])  echo $IP_DYNIP ; ECHO_RETURN=$rc_done ;;
      *)      ECHO_RETURN=" invalid IP_DYNIP=$IP_DYNIP $rc_skipped" ;;
    esac > /proc/sys/net/ipv4/ip_dynaddr || ECHO_RETURN=$rc_failed
    echo -e "$ECHO_RETURN"
   fi

   #
   # Enable syn flood protection
   #
   if test -n "$IP_TCP_SYNCOOKIES" -a "$IP_TCP_SYNCOOKIES" != no -a \
    -e /proc/sys/net/ipv4/tcp_syncookies ; then
    echo -n "Enabling syn flood protection"
    case "$IP_TCP_SYNCOOKIES" in
        yes)    echo 1         ; ECHO_RETURN=$rc_done ;;
        *)      ECHO_RETURN=" invalid
IP_TCP_SYNCOOKIES=$IP_TCP_SYNCOOKIES $rc_skipped" ;;
    esac > /proc/sys/net/ipv4/tcp_syncookies || ECHO_RETURN=$rc_failed
    echo -e "$ECHO_RETURN"
   fi


   #
   # Accept source routing???
   #
   if test -n "$ACCEPT_SOURCE_ROUTE" -a "$ACCEPT_SOURCE_ROUTE" != yes -
a \
    -e /proc/sys/net/ipv4/conf/all/accept_source_route ; then
    echo -n "Disabling Source Routing"
    case "$ACCEPT_SOURCE_ROUTE" in
        no)    echo 0         ; ECHO_RETURN=$rc_done ;;
        *)      ECHO_RETURN=" invalid
ACCEPT_SOURCE_ROUTE=$ACCEPT_SOURCE_ROUTE $rc_skipped" ;;
    esac > /proc/sys/net/ipv4/conf/all/accept_source_route ||
ECHO_RETURN=$rc_failed
    echo -e "$ECHO_RETURN"
   fi


   #
   # Enable syn flood backlog protection
   #
   if test -n "$SYN_FLOOD_BACKLOG" -a "$SYN_FLOOD_BACKLOG" != no -a \
    -e /proc/sys/net/ipv4/tcp_max_syn_backlog ; then
    echo -n "Enabling syn flood BACKLOG protection"
    case "$SYN_FLOOD_BACKLOG" in
        yes)    echo 4096         ; ECHO_RETURN=$rc_done ;;
        *)      ECHO_RETURN=" invalid
SYN_FLOOD_BACKLOG=$SYN_FLOOD_BACKLOG $rc_skipped" ;;
    esac > /proc/sys/net/ipv4/tcp_max_syn_backlog ||
ECHO_RETURN=$rc_failed
    echo -e "$ECHO_RETURN"
   fi


   #
   # Enable IP spoofing protection
   #
   if test -n "$RP_FILTER" -a "$RP_FILTER" != no -a \
    -e /proc/sys/net/ipv4/conf/all/rp_filter ; then
    echo -n "Enabling IP spoofing protection"
    case "$RP_FILTER" in
        yes)    echo 1         ; ECHO_RETURN=$rc_done ;;
        *)      ECHO_RETURN=" invalid RP_FILTER=$RP_FILTER $rc_skipped"
;;
    esac > /proc/sys/net/ipv4/conf/all/rp_filter ||
ECHO_RETURN=$rc_failed
    echo -e "$ECHO_RETURN"
   fi


   #
   # Disable ICMP Redirects from being sent
   #
   if test -n "$DISABLE_SEND_REDIRECTS" -a "$DISABLE_SEND_REDIRECTS" !=
no -a \
    -e /proc/sys/net/ipv4/conf/all/send_redirects ; then
    echo -n "Disabling sending of Redirects"
    case "$DISABLE_SEND_REDIRECTS" in
        yes)    echo 0         ; ECHO_RETURN=$rc_done ;;
        *)      ECHO_RETURN=" invalid
DISABLE_SEND_REDIRECTS=$DISABLE_SEND_REDIRECTS $rc_skipped" ;;
    esac > /proc/sys/net/ipv4/conf/all/send_redirects ||
ECHO_RETURN=$rc_failed
    echo -e "$ECHO_RETURN"
   fi


   #
   # Disable ICMP Redirects from being accepted by all current
interfaces
   #
   if test -n "$DISABLE_ACCEPT_REDIRECTS_ALL" -a
"$DISABLE_ACCEPT_REDIRECTS_ALL" != no -a \
    -e /proc/sys/net/ipv4/conf/all/accept_redirects ; then
    echo -n "Disabling accepting of Redirects"
    case "$DISABLE_ACCEPT_REDIRECTS_ALL" in
        yes)    echo 0         ; ECHO_RETURN=$rc_done ;;
        *)      ECHO_RETURN=" invalid
DISABLE_ACCEPT_REDIRECTS_ALL=$DISABLE_ACCEPT_REDIRECTS_ALL $rc_skipped"
;;
    esac > /proc/sys/net/ipv4/conf/all/accept_redirects ||
ECHO_RETURN=$rc_failed
    echo -e "$ECHO_RETURN"
   fi



   #
   # Disable ICMP Redirects from being accepted by newly activated
interfaces
   #
   if test -n "$DISABLE_ACCEPT_REDIRECTS_DEFAULT" -a
"$DISABLE_ACCEPT_REDIRECTS_DEFAULT" != no -a \
    -e /proc/sys/net/ipv4/conf/default/accept_redirects ; then
    echo -n "Disabling accepting of Redirects on newly activated
interfaces"
    case "$DISABLE_SEND_REDIRECTS" in
        yes)    echo 0         ; ECHO_RETURN=$rc_done ;;
        *)      ECHO_RETURN=" invalid
DISABLE_ACCEPT_REDIRECTS_DEFAULT=$DISABLE_ACCEPT_REDIRECTS_DEFAULT
$rc_skipped" ;;
    esac > /proc/sys/net/ipv4/conf/default/accept_redirects ||
ECHO_RETURN=$rc_failed
    echo -e "$ECHO_RETURN"
   fi




   #
   # Enable IP forwarding ?
   #
   if test -e /proc/sys/net/ipv4/ip_forward -a -n "$IP_FORWARD" ; then
    case $IP_FORWARD in
      yes)
        echo -n "Enabling IP forwarding"
        echo "1" > /proc/sys/net/ipv4/ip_forward
      ;;
      *)
        echo -n "Disabling IP forwarding"
        echo "0" > /proc/sys/net/ipv4/ip_forward
      ;;
    esac
    rc_status -v -r
   fi

   #
   # Enable IPv6 forwarding ?
   #
   LOAD_IPV6="no"
   case $IPV6_FORWARD in
      yes) LOAD_IPV6="yes" ;;
   esac
   case $IPV6_PRIVACY in
      yes) LOAD_IPV6="yes" ;;
   esac
   test "$LOAD_IPV6" = "yes" && /sbin/modprobe ipv6 >/dev/null 2>&1
   #
   if test -e /proc/sys/net/ipv6/conf/all/forwarding -a -n
"$IPV6_FORWARD" ; then
    case $IPV6_FORWARD in
      yes)
      echo -n "Enabling IPv6 forwarding"
      echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
      ;;
      *)
      echo -n "Disabling IPv6 forwarding"
      echo "0" > /proc/sys/net/ipv6/conf/all/forwarding
      ;;
    esac
    rc_status -v -r
   fi

   #
   # Enable IPv6 privacy?
   #
   if test -e /proc/sys/net/ipv6/conf/all/use_tempaddr -a -n
"$IPV6_PRIVACY"; then
    case $IPV6_PRIVACY in
      yes)
      echo -n "Enabling IPv6 privacy"
      echo "1" > /proc/sys/net/ipv6/conf/all/use_tempaddr
      ;;
      *)
      echo -n "Disabling IPv6 privacy"
      echo "0" > /proc/sys/net/ipv6/conf/all/use_tempaddr
      ;;
    esac
    rc_status -v -r
   fi
   ;;
  stop)
    rc_failed 3
```

```
        rc_status -v
        ;;
      status)
        rc_failed 4
        rc_status -v
        ;;
      *)
        echo "Usage: $0 {start|stop|status}"
        exit 1
        ;;
    esac

    rc_exit
```

Appendix D – fwup.sh script to start iptables firewall

```
#!/bin/bash
#######################################################################
###
# IPTABLES VERSION
# This sample configuration is for a single host firewall configuration
#
#######################################################################
###

# USER CONFIGURABLE SECTION

# The name and location of the ipchains utility.
IPTABLES=iptables

# The path to the ipchains executable. Don't need to add to path
variable
# PATH="/usr/sbin"

# Our internal network address space and its supporting network device.
OURNET="192.168.1.0/24"
OURBCAST="192.168.1.255"
OURDEV="eth0"

# The outside address and the network device that supports it.
ANYADDR="0/0"
ANYDEV="eth0"

# The TCP services we wish to allow to pass - "" empty means all ports
# note: comma separated up to 15 values for each
TCPINTERNAL="https,www,ssh,8008,8010,8018,8020,ldap,ldaps,ncp,505,3019"
TCPOUTINTERNAL="smtp,www,ftp,ftp-
data,https,ldap,ldaps,445,139,137,cifs,601"
TCPEXTERNAL="https,www,8010,8020"
TCPOUTEXTERNAL="www,ftp,ftp-data,ntp,https"

# The UDP services we wish to allow to pass - "" empty means all ports
# note: comma separated
UDPINTERNAL="domain,ssh,ldap,ncp,ldap,ldaps"
UDPOUT="domain,ntp,ldap,ldaps,445,139,137,cifs,syslog,601"
UDPEXTERNAL="ssh,ldaps"

# The ICMP services we wish to allow to pass - "" empty means all types
# ref: /usr/include/netinet/ip_icmp.h for type numbers
# note: comma separated
ICMPIN="0,3,11"
ICMPOUT="8,3,11"

# Logging; uncomment the following line to enable logging of datagrams
# that are blocked by the firewall.
LOGGING=1

# END USER CONFIGURABLE SECTION
#######################################################################
####
```

```
# Flush the ALL table rules
$IPTABLES -F

# We want to deny incoming access by default.
$IPTABLES -P FORWARD DROP
$IPTABLES -P INPUT DROP
# Drop all datagrams destined for this host received from outside.
Can't
# do this since this host will have to have services running on it.
#$IPTABLES -A INPUT -i $ANYDEV -j DROP

# SPOOFING
# We should not accept any datagrams with a source address matching
ours
# from the outside, so we deny them.
#$IPTABLES -A INPUT -s $OURNET -i $ANYDEV -j DROP

# SMURF
# Disallow ICMP to our broadcast address to prevent "Smurf" style
attack.
$IPTABLES -A INPUT -p icmp -i $ANYDEV -d $OURBCAST -j DROP

# We should accept fragments, in iptables we must do this explicitly.
$IPTABLES -A INPUT -f -j ACCEPT

# We should accept any traffic originating from the loopback and going
to the loopback
# Since routing is not on only traffic from this machine can be
received on the loopback.
# Change this section if additional interfaces are added or routing is
enabled.
$IPTABLES -A INPUT -i lo -j ACCEPT

# TCP
# We will accept all TCP datagrams belonging to an existing connection
# for the TCP ports we're allowing through.
# This should catch more than 95 % of all valid TCP packets.
$IPTABLES -A INPUT -m multiport -m conntrack -p tcp -s $OURNET -d
$OURNET --dports $TCPINTERNAL --ctstate ESTABLISHED,NEW,RELATED -j
ACCEPT
$IPTABLES -A INPUT -m multiport -m conntrack -p tcp -s $OURNET -d
$OURNET --sports $TCPINTERNAL --ctstate ESTABLISHED,NEW,RELATED -j
ACCEPT
$IPTABLES -A INPUT -m multiport -m conntrack -p tcp -s $OURNET -d
$OURNET --dports $TCPOUTINTERNAL --ctstate ESTABLISHED,NEW,RELATED -j
ACCEPT
$IPTABLES -A INPUT -m multiport -m conntrack -p tcp -s $OURNET -d
$OURNET --sports $TCPOUTINTERNAL --ctstate ESTABLISHED,NEW,RELATED -j
ACCEPT

$IPTABLES -A INPUT -m multiport -m conntrack -p tcp -d $OURNET --dports
$TCPEXTERNAL --ctstate ESTABLISHED,NEW,RELATED -j ACCEPT
$IPTABLES -A INPUT -m multiport -m conntrack -p tcp -s $OURNET --sports
$TCPEXTERNAL --ctstate ESTABLISHED,NEW,RELATED -j ACCEPT

$IPTABLES -A INPUT -m multiport -m conntrack -p tcp -s $OURNET --dports
$TCPOUTEXTERNAL --ctstate ESTABLISHED,NEW,RELATED -j ACCEPT
```

```
$IPTABLES -A INPUT -m multiport -m conntrack -p tcp -d $OURNET --sports
$TCPOUTEXTERNAL --ctstate ESTABLISHED,NEW,RELATED -j ACCEPT

# TCP - INCOMING CONNECTIONS
# We will accept connection requests from the INside only on the
# allowed TCP ports.
$IPTABLES -A INPUT -m multiport -p tcp -i $ANYDEV -s $OURNET -d $OURNET
--dports $TCPINTERNAL --syn -j ACCEPT

# We will accept connection requests from the OUTside only on the
# allowed TCP ports.
$IPTABLES -A INPUT -m multiport -p tcp -i $ANYDEV -s $ANYADDR -d
$OURNET --dports $TCPEXTERNAL --syn -j ACCEPT


# TCP - OUTGOING CONNECTIONS Internal
# We will accept all outgoing tcp connection requests on the allowed
TCP ports for INTERNAL traffic

$IPTABLES -A INPUT -m multiport -p tcp -i $OURDEV -s $OURNET -d $OURNET
--dports $TCPOUTINTERNAL --syn -j ACCEPT

# TCP - OUTGOING CONNECTIONS External
# We will accept all outgoing tcp connection requests on the allowed
TCP ports for INTERNAL traffic

$IPTABLES -A INPUT -m multiport -p tcp -i $OURDEV -s $OURNET -d
$ANYADDR --dports $TCPOUTEXTERNAL --syn -j ACCEPT

# UDP - INCOMING
# We will allow UDP datagrams in on the allowed ports and back.
$IPTABLES -A INPUT -m multiport -p udp -i $ANYDEV -s $OURNET -d $OURNET
--dports $UDPINTERNAL -j ACCEPT
$IPTABLES -A INPUT -m multiport -p udp -i $ANYDEV -s $OURNET -d $OURNET
--sports $UDPINTERNAL -j ACCEPT

$IPTABLES -A INPUT -m multiport -p udp -i $ANYDEV -s $ANYADDR -d
$OURNET --dports $UDPEXTERNAL -j ACCEPT
$IPTABLES -A INPUT -m multiport -p udp -i $ANYDEV -s $OURNET -d
$ANYADDR --sports $UDPEXTERNAL -j ACCEPT

# UDP - OUTGOING
# We will allow UDP datagrams out to the allowed ports and back.
$IPTABLES -A INPUT -m multiport -p udp -i $OURDEV -s $OURNET -d
$ANYADDR --dports $UDPOUT -j ACCEPT

$IPTABLES -A INPUT -m multiport -p udp -i $OURDEV -s $ANYADDR -d
$OURNET --sports $UDPOUT -j ACCEPT

# ICMP - INCOMING
# We will allow ICMP datagrams in of the allowed types.
$IPTABLES -A INPUT -p icmp -i $ANYDEV -d $OURNET --icmp-type 0 -j
ACCEPT
$IPTABLES -A INPUT -p icmp -i $ANYDEV -d $OURNET --icmp-type 3 -j
ACCEPT
$IPTABLES -A INPUT -p icmp -i $ANYDEV -d $OURNET --icmp-type 11 -j
ACCEPT
```

```
# ICMP - OUTGOING
# We will allow ICMP datagrams out of the allowed types.
$IPTABLES -A INPUT -p icmp -i $OURDEV -d $ANYADDR --icmp-type 8 -j
ACCEPT
$IPTABLES -A INPUT -p icmp -i $OURDEV -d $ANYADDR --icmp-type 3 -j
ACCEPT
$IPTABLES -A INPUT -p icmp -i $OURDEV -d $ANYADDR --icmp-type 11 -j
ACCEPT



# DEFAULT and LOGGING
# All remaining datagrams fall through to the default
# rule and are dropped. They will be logged if you've
# configured the LOGGING variable above.
#
if [ "$LOGGING" ]
then
        # Log barred TCP
        $IPTABLES -A INPUT -m tcp -p tcp -j LOG
        # Log barred UDP
        $IPTABLES -A INPUT -m udp -p udp -j LOG
        # Log barred ICMP
        $IPTABLES -A INPUT -m icmp -p icmp -j LOG
fi
#
# end.
```

Appendix E – mod_security configuration

```
# mod_security Configuration Section. If Mod_security is not used
comment all
# of these lines out.


# Yes, we want to use mod_security
SecFilterEngine On

# Scan request body
SecFilterScanPOST On

# Scan response body
SecFilterScanOutput On

# Check URL encoding
SecFilterCheckURLEncoding On

# This setting should be set to On only if the Web site is
# using the Unicode encoding. Otherwise it may interfere with
# the normal Web site operation.
SecFilterCheckUnicodeEncoding Off

# Only allow certain byte values to be a part of the request.
# This is pretty relaxed, most applications where only English
# is used will happily work with a range 32 - 126.
SecFilterForceByteRange 1 255

# Audit log logs complete requests. Configured as below it
# will only log invalid requests for further analysis.
SecAuditEngine RelevantOnly
SecAuditLog logs/audit_log

# You may need this later but we don't log anything
# here for now. Excessive debug logging may slow down
# the server.
SecFilterDebugLevel 0
SecFilterDebugLog logs/modsec_debug_log

# By default, deny requests with status 500
SecFilterDefaultAction "deny,log,status:500"


# Below this section are the more restrictive additional configs.

# Command execution attacks
SecFilter /etc/password
SecFilter /bin/ls
# Directory traversal attacks
SecFilter "\.\./"
# XSS attacks ---Note: disabled first line because it breaks iFolder
access
# in NetStorage
#SecFilter "<(.|\n)+>"
SecFilter "<[[:space:]]*script"
```

```
# Detect responses that might indicate and intrusion
SecFilterSelective OUTPUT "Volume Serial Number"

SecFilterSelective OUTPUT "Command completed"

SecFilterSelective OUTPUT "Bad command or filename"

SecFilterSelective OUTPUT "file(s) copied"

SecFilterSelective OUTPUT "Index of /cgi-bin/"

SecFilterSelective OUTPUT ".*uid\=\("

# SNORT rules section. These are SNORT rules that have been converted
to
# mod_security syntax. Rules that don't apply have been removed.

# WEB-ATTACKS ps command attempt
SecFilterSelective THE_REQUEST "/bin/ps"

# WEB-ATTACKS /bin/ps command attempt
#SecFilterSelective THE_REQUEST "ps\x20"

# WEB-ATTACKS wget command attempt
SecFilter "wget\x20"

# WEB-ATTACKS uname -a command attempt
SecFilter "uname\x20-a"

# WEB-ATTACKS /usr/bin/id command attempt
SecFilter "/usr/bin/id"

# WEB-ATTACKS id command attempt
SecFilter "\;id"

# WEB-ATTACKS echo command attempt
SecFilter "/bin/echo"

# WEB-ATTACKS kill command attempt
SecFilter "/bin/kill"

# WEB-ATTACKS chmod command attempt
SecFilter "/bin/chmod"

# WEB-ATTACKS chgrp command attempt
SecFilter "/chgrp"

# WEB-ATTACKS chown command attempt
SecFilter "/chown"

# WEB-ATTACKS chsh command attempt
SecFilter "/usr/bin/chsh"

# WEB-ATTACKS tftp command attempt
SecFilter "tftp\x20"
```

```
# WEB-ATTACKS /usr/bin/gcc command attempt
SecFilter "/usr/bin/gcc"

# WEB-ATTACKS gcc command attempt
SecFilter "gcc\x20-o"

# WEB-ATTACKS /usr/bin/cc command attempt
SecFilter "/usr/bin/cc"

# WEB-ATTACKS cc command attempt
SecFilter "cc\x20"

# WEB-ATTACKS /usr/bin/cpp command attempt
SecFilter "/usr/bin/cpp"

# WEB-ATTACKS cpp command attempt
SecFilter "cpp\x20"

# WEB-ATTACKS /usr/bin/g++ command attempt
SecFilter "/usr/bin/g\+\+"

# WEB-ATTACKS g++ command attempt
SecFilter "g\+\+\x20"

# WEB-ATTACKS bin/python access attempt
SecFilter "bin/python"

# WEB-ATTACKS python access attempt
SecFilter "python\x20"

# WEB-ATTACKS bin/tclsh execution attempt
SecFilter "bin/tclsh"

# WEB-ATTACKS tclsh execution attempt
SecFilter "tclsh8\x20"

# WEB-ATTACKS bin/nasm command attempt
SecFilter "bin/nasm"

# WEB-ATTACKS nasm command attempt
SecFilter "nasm\x20"

# WEB-ATTACKS /usr/bin/perl execution attempt
SecFilter "/usr/bin/perl"

# WEB-ATTACKS perl execution attempt
SecFilter "perl\x20"

# WEB-ATTACKS traceroute command attempt
SecFilter "traceroute\x20"

# WEB-ATTACKS ping command attempt
SecFilter "/bin/ping"

# WEB-ATTACKS netcat command attempt
SecFilter "nc\x20"
```

```
# WEB-ATTACKS nmap command attempt
SecFilter "nmap\x20"

# WEB-ATTACKS xterm command attempt
SecFilter "/usr/X11R6/bin/xterm"

# WEB-ATTACKS X application to remote host attempt
SecFilter "\x20-display\x20"

# WEB-ATTACKS lsof command attempt
SecFilter "lsof\x20"

# WEB-ATTACKS rm command attempt
SecFilter "rm\x20"

# WEB-ATTACKS mail command attempt
SecFilter "/bin/mail"

# WEB-ATTACKS mail command attempt
SecFilter "mail\x20"

# WEB-ATTACKS /bin/ls command attempt
SecFilterSelective THE_REQUEST "/bin/ls"

# WEB-ATTACKS /etc/inetd.conf access
SecFilter "/etc/inetd\.conf" log,pass

# WEB-ATTACKS /etc/motd access
SecFilter "/etc/motd" log,pass

# WEB-ATTACKS /etc/shadow access
SecFilter "/etc/shadow" log,pass

# WEB-ATTACKS conf/httpd.conf attempt
SecFilter "conf/httpd\.conf" log,pass

# WEB-ATTACKS .htgroup access
SecFilterSelective THE_REQUEST "\.htgroup" log,pass

# WEB-CGI HyperSeek hsx.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/hsx\.cgi" chain
SecFilter "\x00"

# WEB-CGI HyperSeek hsx.cgi access
SecFilterSelective THE_REQUEST "/hsx\.cgi" log,pass

# WEB-CGI SWSoft ASPSeek Overflow attempt
SecFilterSelective THE_REQUEST "/s\.cgi" chain
SecFilter "tmpl="

# WEB-CGI webspeed access
SecFilterSelective THE_REQUEST "/wsisa\.dll/WService=" chain
SecFilter "WSMadmin"

# WEB-CGI yabb.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/YaBB\.pl" chain
SecFilter "\.\./"
```

```
# WEB-CGI yabb.cgi access
SecFilterSelective THE_REQUEST "/YaBB\.pl"

# WEB-CGI whois_raw.cgi access
SecFilterSelective THE_REQUEST "/whois_raw\.cgi"

# WEB-CGI glimpse access
SecFilterSelective THE_REQUEST "/glimpse"

# WEB-CGI htmlscript attempt
SecFilterSelective THE_REQUEST "/htmlscript\?\.\./\.\."

# WEB-CGI htmlscript access
SecFilterSelective THE_REQUEST "/htmlscript"

# WEB-CGI info2www access
SecFilterSelective THE_REQUEST "/info2www"

# WEB-CGI maillist.pl access
SecFilterSelective THE_REQUEST "/maillist\.pl"

# WEB-CGI nph-test-cgi access
SecFilterSelective THE_REQUEST "/nph-test-cgi"

# WEB-CGI NPH-publish access
SecFilterSelective THE_REQUEST "/nph-maillist\.pl"

# WEB-CGI NPH-publish access
SecFilterSelective THE_REQUEST "/nph-publish"

# WEB-CGI rguest.exe access
SecFilterSelective THE_REQUEST "/rguest\.exe"

# WEB-CGI rwwwshell.pl access
SecFilterSelective THE_REQUEST "/rwwwshell\.pl"

# WEB-CGI test-cgi attempt
SecFilterSelective THE_REQUEST "/test-cgi/*\?*"

# WEB-CGI test-cgi access
SecFilterSelective THE_REQUEST "/test-cgi"

# WEB-CGI testcgi access
SecFilterSelective THE_REQUEST "/testcgi" log,pass

# WEB-CGI test.cgi access
SecFilterSelective THE_REQUEST "/test\.cgi" log,pass

# WEB-CGI textcounter.pl access
SecFilterSelective THE_REQUEST "/textcounter\.pl"

# WEB-CGI uploader.exe access
SecFilterSelective THE_REQUEST "/uploader\.exe"

# WEB-CGI webgais access
SecFilterSelective THE_REQUEST "/webgais"
```

```
# WEB-CGI finger access
SecFilterSelective THE_REQUEST "/finger"

# WEB-CGI perlshop.cgi access
SecFilterSelective THE_REQUEST "/perlshop\.cgi"

# WEB-CGI pfdisplay.cgi access
SecFilterSelective THE_REQUEST "/pfdisplay\.cgi"

# WEB-CGI aglimpse access
SecFilterSelective THE_REQUEST "/aglimpse"

# WEB-CGI anform2 access
SecFilterSelective THE_REQUEST "/AnForm2"

# WEB-CGI AT-admin.cgi access
SecFilterSelective THE_REQUEST "/AT-admin\.cgi"

# WEB-CGI AT-generated.cgi access
SecFilterSelective THE_REQUEST "/AT-generated\.cgi"

# WEB-CGI bnbform.cgi access
SecFilterSelective THE_REQUEST "/bnbform\.cgi"

# WEB-CGI campas access
SecFilterSelective THE_REQUEST "/campas"

# WEB-CGI view-source directory traversal
SecFilterSelective THE_REQUEST "/view-source" chain
SecFilter "\.\./"

# WEB-CGI view-source access
SecFilterSelective THE_REQUEST "/view-source"

# WEB-CGI wais.pl access
SecFilterSelective THE_REQUEST "/wais\.pl"

# WEB-CGI wwwwais access
SecFilterSelective THE_REQUEST "/wwwwais"

# WEB-CGI files.pl access
SecFilterSelective THE_REQUEST "/files\.pl"

# WEB-CGI wrap access
SecFilterSelective THE_REQUEST "/wrap"

# WEB-CGI classifieds.cgi access
SecFilterSelective THE_REQUEST "/classifieds\.cgi"

# WEB-CGI environ.cgi access
SecFilterSelective THE_REQUEST "/environ\.cgi"

# WEB-CGI faxsurvey attempt (full path)
SecFilterSelective THE_REQUEST "/faxsurvey\?/"

# WEB-CGI faxsurvey arbitrary file read attempt
```

```
SecFilterSelective THE_REQUEST "/faxsurvey\?cat\x20"

# WEB-CGI faxsurvey access
SecFilterSelective THE_REQUEST "/faxsurvey" log,pass

# WEB-CGI filemail access
SecFilterSelective THE_REQUEST "/filemail\.pl"

# WEB-CGI man.sh access
SecFilterSelective THE_REQUEST "/man\.sh"

# WEB-CGI day5datacopier.cgi access
SecFilterSelective THE_REQUEST "/day5datacopier\.cgi"

# WEB-CGI day5datanotifier.cgi access
SecFilterSelective THE_REQUEST "/day5datanotifier\.cgi"

# WEB-CGI post-query access
SecFilterSelective THE_REQUEST "/post-query"

# WEB-CGI dumpenv.pl access
SecFilterSelective THE_REQUEST "/dumpenv\.pl"

# WEB-CGI calendar_admin.pl access
SecFilterSelective THE_REQUEST "/calendar_admin\.pl" log,pass

# WEB-CGI calendar-admin.pl access
SecFilterSelective THE_REQUEST "/calendar-admin\.pl" log,pass

# WEB-CGI calender.pl access
SecFilterSelective THE_REQUEST "/calender\.pl"

# WEB-CGI calendar access
SecFilterSelective THE_REQUEST "/calendar"

# WEB-CGI user_update_admin.pl access
SecFilterSelective THE_REQUEST "/user_update_admin\.pl"

# WEB-CGI user_update_passwd.pl access
SecFilterSelective THE_REQUEST "/user_update_passwd\.pl"

# WEB-CGI survey.cgi access
SecFilterSelective THE_REQUEST "/survey\.cgi"

# WEB-CGI scriptalias access
SecFilterSelective THE_REQUEST "///"

# WEB-CGI win-c-sample.exe access
SecFilterSelective THE_REQUEST "/win-c-sample\.exe"

# WEB-CGI w3tvars.pm access
SecFilterSelective THE_REQUEST "/w3tvars\.pm"

# WEB-CGI admin.pl access
SecFilterSelective THE_REQUEST "/admin\.pl"

# WEB-CGI LWGate access
```

```
SecFilterSelective THE_REQUEST "/LWGate"

# WEB-CGI archie access
SecFilterSelective THE_REQUEST "/archie"

# WEB-CGI flexform access
SecFilterSelective THE_REQUEST "/flexform"

# WEB-CGI phf arbitrary command execution attempt
SecFilterSelective THE_REQUEST "/phf" chain
SecFilter "\x0a/"

# WEB-CGI phf access
SecFilterSelective THE_REQUEST "/phf" log,pass

# WEB-CGI www-sql access
SecFilterSelective THE_REQUEST "/www-sql"

# WEB-CGI wwwadmin.pl access
SecFilterSelective THE_REQUEST "/wwwadmin\.pl"

# WEB-CGI sendform.cgi access
SecFilterSelective THE_REQUEST "/sendform\.cgi"

# WEB-CGI upload.pl access
SecFilterSelective THE_REQUEST "/upload\.pl"

# WEB-CGI AnyForm2 access
SecFilterSelective THE_REQUEST "/AnyForm2"

# WEB-CGI MachineInfo access
SecFilterSelective THE_REQUEST "/MachineInfo"

# WEB-CGI bb-hist.sh attempt
SecFilterSelective THE_REQUEST "/bb-hist\.sh\?HISTFILE=\.\./\.\."

# WEB-CGI bb-hist.sh access
SecFilterSelective THE_REQUEST "/bb-hist\.sh"

# WEB-CGI bb-histlog.sh access
SecFilterSelective THE_REQUEST "/bb-histlog\.sh"

# WEB-CGI bb-histsvc.sh access
SecFilterSelective THE_REQUEST "/bb-histsvc\.sh"

# WEB-CGI bb-hostscv.sh attempt
SecFilterSelective THE_REQUEST "/bb-hostsvc\.sh\?HOSTSVC\?\.\./\.\."

# WEB-CGI bb-hostscv.sh access
SecFilterSelective THE_REQUEST "/bb-hostsvc\.sh" log,pass

# WEB-CGI bb-rep.sh access
SecFilterSelective THE_REQUEST "/bb-rep\.sh"

# WEB-CGI bb-replog.sh access
SecFilterSelective THE_REQUEST "/bb-replog\.sh"
```

```
# WEB-CGI redirect access
SecFilterSelective THE_REQUEST "/redirect"

# WEB-CGI wayboard attempt
SecFilterSelective THE_REQUEST "/way-board/way-board\.cgi" chain
SecFilter "\.\./\.\."

# WEB-CGI way-board access
SecFilterSelective THE_REQUEST "/way-board" log,pass

# WEB-CGI pals-cgi arbitrary file access attempt
SecFilterSelective THE_REQUEST "/pals-cgi" chain
SecFilter "documentName="

# WEB-CGI pals-cgi access
SecFilterSelective THE_REQUEST "/pals-cgi"

# WEB-CGI commerce.cgi arbitrary file access attempt
SecFilterSelective THE_REQUEST "/commerce\.cgi" chain
SecFilter "/\.\./"

# WEB-CGI commerce.cgi access
SecFilterSelective THE_REQUEST "/commerce\.cgi"

# WEB-CGI Amaya templates sendtemp.pl directory traversal attempt
SecFilterSelective THE_REQUEST "/sendtemp\.pl" chain
SecFilter "templ="

# WEB-CGI Amaya templates sendtemp.pl access
SecFilterSelective THE_REQUEST "/sendtemp\.pl" log,pass

# WEB-CGI webspirs.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/webspirs\.cgi" chain
SecFilter "\.\./\.\./"

# WEB-CGI webspirs.cgi access
SecFilterSelective THE_REQUEST "/webspirs\.cgi"

# WEB-CGI tstisapi.dll access
SecFilterSelective THE_REQUEST "tstisapi\.dll"

# WEB-CGI sendmessage.cgi access
SecFilterSelective THE_REQUEST "/sendmessage\.cgi"

# WEB-CGI lastlines.cgi access
SecFilterSelective THE_REQUEST "/lastlines\.cgi"

# WEB-CGI zml.cgi attempt
SecFilterSelective THE_REQUEST "/zml\.cgi" chain
SecFilter "file=\.\./" log,pass

# WEB-CGI zml.cgi access
SecFilterSelective THE_REQUEST "/zml\.cgi" log,pass

# WEB-CGI AHG search.cgi access
SecFilterSelective THE_REQUEST "/publisher/search\.cgi" chain
SecFilter "template=" log,pass
```

```
# WEB-CGI agora.cgi attempt
SecFilterSelective THE_REQUEST "/store/agora\.cgi\?cart_id=<SCRIPT>"

# WEB-CGI agora.cgi access
SecFilterSelective THE_REQUEST "/store/agora\.cgi" log,pass

# WEB-CGI rksh access
SecFilterSelective THE_REQUEST "/rksh"

# WEB-CGI bash access
SecFilterSelective THE_REQUEST "/bash" log,pass

# WEB-CGI perl.exe command attempt
SecFilterSelective THE_REQUEST "/perl\.exe\?"

# WEB-CGI perl.exe access
SecFilterSelective THE_REQUEST "/perl\.exe"

# WEB-CGI perl command attempt
SecFilterSelective THE_REQUEST "/perl\?"

# WEB-CGI zsh access
SecFilterSelective THE_REQUEST "/zsh"

# WEB-CGI csh access
SecFilterSelective THE_REQUEST "/csh"

# WEB-CGI tcsh access
SecFilterSelective THE_REQUEST "/tcsh"

# WEB-CGI rsh access
SecFilterSelective THE_REQUEST "/rsh"

# WEB-CGI ksh access
SecFilterSelective THE_REQUEST "/ksh"

# WEB-CGI auktion.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/auktion\.cgi" chain
SecFilter "menue=\.\./\.\./"

# WEB-CGI auktion.cgi access
SecFilterSelective THE_REQUEST "/auktion\.cgi" log,pass

# WEB-CGI cgiforum.pl attempt
SecFilterSelective THE_REQUEST "/cgiforum\.pl\?thesection=\.\./\.\."

# WEB-CGI cgiforum.pl access
SecFilterSelective THE_REQUEST "/cgiforum\.pl" log,pass

# WEB-CGI directorypro.cgi attempt
SecFilterSelective THE_REQUEST "/directorypro\.cgi" chain
SecFilter "\.\./\.\."

# WEB-CGI directorypro.cgi access
SecFilterSelective THE_REQUEST "/directorypro\.cgi" log,pass
```

```
# WEB-CGI Web Shopper shopper.cgi attempt
SecFilterSelective THE_REQUEST "/shopper\.cgi" chain
SecFilter "newpage=\.\./"

# WEB-CGI Web Shopper shopper.cgi access
SecFilterSelective THE_REQUEST "/shopper\.cgi"

# WEB-CGI listrec.pl access
SecFilterSelective THE_REQUEST "/listrec\.pl"

# WEB-CGI mailnews.cgi access
SecFilterSelective THE_REQUEST "/mailnews\.cgi"

# WEB-CGI book.cgi access
SecFilterSelective THE_REQUEST "/book\.cgi" log,pass

# WEB-CGI newsdesk.cgi access
SecFilterSelective THE_REQUEST "/newsdesk\.cgi"

# WEB-CGI cal_make.pl directory traversal attempt
SecFilterSelective THE_REQUEST "/cal_make\.pl" chain
SecFilter "p0=\.\./\.\./"

# WEB-CGI cal_make.pl access
SecFilterSelective THE_REQUEST "/cal_make\.pl" log,pass

# WEB-CGI mailit.pl access
SecFilterSelective THE_REQUEST "/mailit\.pl"

# WEB-CGI sdbsearch.cgi access
SecFilterSelective THE_REQUEST "/sdbsearch\.cgi"

# WEB-CGI swc access
SecFilterSelective THE_REQUEST "/swc"

# WEB-CGI ttawebtop.cgi arbitrary file attempt
SecFilterSelective THE_REQUEST "/ttawebtop\.cgi" chain
SecFilter "pg=\.\./"

# WEB-CGI ttawebtop.cgi access
SecFilterSelective THE_REQUEST "/ttawebtop\.cgi"

# WEB-CGI upload.cgi access
SecFilterSelective THE_REQUEST "/upload\.cgi"

# WEB-CGI view_source access
SecFilterSelective THE_REQUEST "/view_source"

# WEB-CGI ustorekeeper.pl directory traversal attempt
SecFilterSelective THE_REQUEST "/ustorekeeper\.pl" chain
SecFilter "file=\.\./\.\./"

# WEB-CGI ustorekeeper.pl access
SecFilterSelective THE_REQUEST "/ustorekeeper\.pl" log,pass

# WEB-CGI icat access
SecFilterSelective THE_REQUEST "/icat" log,pass
```

```
# WEB-CGI Bugzilla doeditvotes.cgi access
SecFilterSelective THE_REQUEST "/doeditvotes\.cgi" log,pass

# WEB-CGI htsearch arbitrary configuration file attempt
SecFilterSelective THE_REQUEST "/htsearch\?-c"

# WEB-CGI htsearch arbitrary file read attempt
SecFilterSelective THE_REQUEST "/htsearch\?exclude=`"

# WEB-CGI htsearch access
SecFilterSelective THE_REQUEST "/htsearch" log,pass

# WEB-CGI a1stats a1disp3.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/a1disp3\.cgi\?/\.\./\.\./"

# WEB-CGI a1stats a1disp3.cgi access
SecFilterSelective THE_REQUEST "/a1disp3\.cgi" log,pass

# WEB-CGI a1stats access
SecFilterSelective THE_REQUEST "/a1stats/" log,pass

# WEB-CGI admentor admin.asp access
SecFilterSelective THE_REQUEST "/admentor/admin/admin\.asp" log,pass

# WEB-CGI alchemy http server PRN arbitrary command execution attempt
SecFilterSelective THE_REQUEST "/PRN/\.\./\.\./" log,pass

# WEB-CGI alchemy http server NUL arbitrary command execution attempt
SecFilterSelective THE_REQUEST "/NUL/\.\./\.\./" log,pass

# WEB-CGI alibaba.pl access
SecFilterSelective THE_REQUEST "/alibaba\.pl" log,pass

# WEB-CGI AltaVista Intranet Search directory traversal attempt
SecFilterSelective THE_REQUEST "/query\?mss=\.\."

# WEB-CGI /cgi-bin/ls access
SecFilterSelective THE_REQUEST "/cgi-bin/ls" log,pass

# WEB-CGI cgimail access
SecFilterSelective THE_REQUEST "/cgimail" log,pass

# WEB-CGI cgiwrap access
SecFilterSelective THE_REQUEST "/cgiwrap" log,pass

# WEB-CGI csSearch.cgi arbitrary command execution attempt
SecFilterSelective THE_REQUEST "/csSearch\.cgi" chain
SecFilter "`"

# WEB-CGI csSearch.cgi access
SecFilterSelective THE_REQUEST "/csSearch\.cgi" log,pass

# WEB-CGI /cart/cart.cgi access
SecFilterSelective THE_REQUEST "/cart/cart\.cgi" log,pass

# WEB-CGI dbman db.cgi access
```

```
SecFilterSelective THE_REQUEST "/dbman/db\.cgi" log,pass

# WEB-CGI DCShop access
SecFilterSelective THE_REQUEST "/dcshop" log,pass

# WEB-CGI DCShop orders.txt access
SecFilterSelective THE_REQUEST "/orders/orders\.txt" log,pass

# WEB-CGI DCShop auth_user_file.txt access
SecFilterSelective THE_REQUEST "/auth_data/auth_user_file\.txt"
log,pass

# WEB-CGI eshop.pl arbitrary commane execution attempt
SecFilterSelective THE_REQUEST "/eshop\.pl\?seite=\;"

# WEB-CGI eshop.pl access
SecFilterSelective THE_REQUEST "/eshop\.pl" log,pass

# WEB-CGI loadpage.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/loadpage\.cgi" chain
SecFilter "file=\.\./"

# WEB-CGI loadpage.cgi access
SecFilterSelective THE_REQUEST "/loadpage\.cgi" log,pass

# WEB-CGI faqmanager.cgi arbitrary file access attempt
SecFilterSelective THE_REQUEST "\x00"

# WEB-CGI faqmanager.cgi access
SecFilterSelective THE_REQUEST "/faqmanager\.cgi" log,pass

# WEB-CGI /fcgi-bin/echo.exe access
SecFilterSelective THE_REQUEST "/fcgi-bin/echo\.exe" log,pass

# WEB-CGI FormHandler.cgi directory traversal attempt attempt
SecFilterSelective THE_REQUEST "/FormHandler\.cgi" chain
SecFilter "/\.\./"

# WEB-CGI FormHandler.cgi external site redirection attempt
SecFilterSelective THE_REQUEST "/FormHandler\.cgi" chain
SecFilter "redirect=http"

# WEB-CGI FormHandler.cgi access
SecFilterSelective THE_REQUEST "/FormHandler\.cgi" log,pass

# WEB-CGI guestbook.cgi access
SecFilterSelective THE_REQUEST "/guestbook\.cgi" log,pass

# WEB-CGI Home Free search.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/search\.cgi" chain
SecFilter "letter=\.\./\.\."

# WEB-CGI search.cgi access
SecFilterSelective THE_REQUEST "/search\.cgi" log,pass

# WEB-CGI enivorn.pl access
SecFilterSelective THE_REQUEST "/enivron\.pl" log,pass
```

```
# WEB-CGI campus attempt
SecFilterSelective THE_REQUEST "/campus\?\x0a"

# WEB-CGI campus access
SecFilterSelective THE_REQUEST "/campus" log,pass

# WEB-CGI pfdispaly.cgi arbitrary command execution attempt
SecFilterSelective THE_REQUEST "/pfdispaly\.cgi\?'"

# WEB-CGI pfdispaly.cgi access
SecFilterSelective THE_REQUEST "/pfdispaly\.cgi" log,pass

# WEB-CGI pagelog.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/pagelog\.cgi" chain
SecFilter "name=\.\./" log,pass

# WEB-CGI pagelog.cgi access
SecFilterSelective THE_REQUEST "/pagelog\.cgi" log,pass

# WEB-CGI ad.cgi access
SecFilterSelective THE_REQUEST "/ad\.cgi" log,pass

# WEB-CGI bbs_forum.cgi access
SecFilterSelective THE_REQUEST "/bbs_forum\.cgi" log,pass

# WEB-CGI bsguest.cgi access
SecFilterSelective THE_REQUEST "/bsguest\.cgi" log,pass

# WEB-CGI bslist.cgi access
SecFilterSelective THE_REQUEST "/bslist\.cgi" log,pass

# WEB-CGI cgforum.cgi access
SecFilterSelective THE_REQUEST "/cgforum\.cgi" log,pass

# WEB-CGI newdesk access
SecFilterSelective THE_REQUEST "/newdesk" log,pass

# WEB-CGI register.cgi access
SecFilterSelective THE_REQUEST "/register\.cgi" log,pass

# WEB-CGI gbook.cgi access
SecFilterSelective THE_REQUEST "/gbook\.cgi" log,pass

# WEB-CGI simplestguest.cgi access
SecFilterSelective THE_REQUEST "/simplestguest\.cgi" log,pass

# WEB-CGI statusconfig.pl access
SecFilterSelective THE_REQUEST "/statusconfig\.pl" log,pass

# WEB-CGI talkback.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/talkbalk\.cgi" chain
SecFilter "article=\.\./\.\./"

# WEB-CGI talkback.cgi access
SecFilterSelective THE_REQUEST "/talkbalk\.cgi" log,pass
```

```
# WEB-CGI adcycle access
SecFilterSelective THE_REQUEST "/adcycle" log,pass

# WEB-CGI MachineInfo access
SecFilterSelective THE_REQUEST "/MachineInfo" log,pass

# WEB-CGI emumail.cgi NULL attempt
SecFilterSelective THE_REQUEST "/emumail\.cgi" chain
SecFilter "\x00" log,pass

# WEB-CGI emumail.cgi access
SecFilterSelective THE_REQUEST "/emumail\.cgi" log,pass

# WEB-CGI document.d2w access
SecFilterSelective THE_REQUEST "/document\.d2w" log,pass

# WEB-CGI db2www access
SecFilterSelective THE_REQUEST "/db2www" log,pass

# WEB-CGI /cgi-bin/ access
SecFilterSelective THE_REQUEST "/cgi-bin/" chain
SecFilter "/cgi-bin/ HTTP"

# WEB-CGI /cgi-dos/ access
SecFilterSelective THE_REQUEST "/cgi-dos/" chain
SecFilter "/cgi-dos/ HTTP"

# WEB-CGI technote main.cgi file directory traversal attempt
SecFilterSelective THE_REQUEST "/technote/main\.cgi" chain
SecFilter "\.\./\.\./"

# WEB-CGI technote print.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/technote/print\.cgi" chain
SecFilter "\x00"

# WEB-CGI eXtropia webstore directory traversal
SecFilterSelective THE_REQUEST "/web_store\.cgi" chain
SecFilter "page=\.\./"

# WEB-CGI eXtropia webstore access
SecFilterSelective THE_REQUEST "/web_store\.cgi" log,pass

# WEB-CGI shopping cart directory traversal
SecFilterSelective THE_REQUEST "/shop\.cgi" chain
SecFilter "page=\.\./"

# WEB-CGI count.cgi access
SecFilterSelective THE_REQUEST "/count\.cgi" log,pass

# WEB-CGI webdist.cgi arbitrary command attempt
SecFilterSelective THE_REQUEST "/webdist\.cgi" chain
SecFilter "distloc=\;"

# WEB-CGI webdist.cgi access
SecFilterSelective THE_REQUEST "/webdist\.cgi" log,pass

# WEB-CGI bigconf.cgi access
```

```
SecFilterSelective THE_REQUEST "/bigconf\.cgi" log,pass

# WEB-CGI /cgi-bin/jj access
SecFilterSelective THE_REQUEST "/cgi-bin/jj" log,pass

# WEB-CGI bizdbsearch attempt
SecFilterSelective THE_REQUEST "/bizdb1-search\.cgi" chain
SecFilter "mail"

# WEB-CGI bizdbsearch access
SecFilterSelective THE_REQUEST "/bizdb1-search\.cgi" log,pass

# WEB-CGI sojourn.cgi File attempt
SecFilterSelective THE_REQUEST "/sojourn\.cgi\?cat=" chain
SecFilter "\x00"

# WEB-CGI sojourn.cgi access
SecFilterSelective THE_REQUEST "/sojourn\.cgi" log,pass

# WEB-CGI SGI InfoSearch fname attempt
SecFilterSelective THE_REQUEST "/infosrch\.cgi\?" chain
SecFilter "fname="

# WEB-CGI SGI InfoSearch fname access
SecFilterSelective THE_REQUEST "/infosrch\.cgi" log,pass

# WEB-CGI ax-admin.cgi access
SecFilterSelective THE_REQUEST "/ax-admin\.cgi" log,pass

# WEB-CGI axs.cgi access
SecFilterSelective THE_REQUEST "/axs\.cgi" log,pass

# WEB-CGI cachemgr.cgi access
SecFilterSelective THE_REQUEST "/cachemgr\.cgi" log,pass

# WEB-CGI responder.cgi access
SecFilterSelective THE_REQUEST "/responder\.cgi" log,pass

# WEB-CGI web-map.cgi access
SecFilterSelective THE_REQUEST "/web-map\.cgi" log,pass

# WEB-CGI ministats admin access
SecFilterSelective THE_REQUEST "/ministats/admin\.cgi" log,pass

# WEB-CGI dfire.cgi access
SecFilterSelective THE_REQUEST "/dfire\.cgi" log,pass

# WEB-CGI txt2html.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/txt2html\.cgi" chain
SecFilter "/\.\./\.\./\.\./\.\./"

# WEB-CGI txt2html.cgi access
SecFilterSelective THE_REQUEST "/txt2html\.cgi" log,pass

# WEB-CGI store.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/store\.cgi" chain
SecFilter "\.\./"
```

```
# WEB-CGI store.cgi access
SecFilterSelective THE_REQUEST "/store\.cgi" log,pass

# WEB-CGI SIX webboard generate.cgi attempt
SecFilterSelective THE_REQUEST "/generate\.cgi" chain
SecFilter "content=\.\./"

# WEB-CGI SIX webboard generate.cgi access
SecFilterSelective THE_REQUEST "/generate\.cgi" log,pass

# WEB-CGI spin_client.cgi access
SecFilterSelective THE_REQUEST "/spin_client\.cgi" log,pass

# WEB-CGI csPassword.cgi access
SecFilterSelective THE_REQUEST "/csPassword\.cgi" log,pass

# WEB-CGI csPassword password.cgi.tmp access
SecFilterSelective THE_REQUEST "/password\.cgi\.tmp" log,pass

# WEB-CGI Nortel Contivity cgiproc DOS attempt
SecFilterSelective THE_REQUEST "/cgiproc\?Nocfile="

# WEB-CGI Nortel Contivity cgiproc DOS attempt
SecFilterSelective THE_REQUEST "/cgiproc\?\$"

# WEB-CGI Nortel Contivity cgiproc access
SecFilterSelective THE_REQUEST "/cgiproc" log,pass

# WEB-CGI Oracle reports CGI access
SecFilterSelective THE_REQUEST "/rwcgi60" chain
SecFilter "setauth=" log,pass

# WEB-CGI alienform.cgi access
SecFilterSelective THE_REQUEST "/alienform\.cgi" log,pass

# WEB-CGI AlienForm af.cgi access
SecFilterSelective THE_REQUEST "/af\.cgi" log,pass

# WEB-CGI story.pl arbitrary file read attempt
SecFilterSelective THE_REQUEST "/story\.pl" chain
SecFilter "next=\.\./"

# WEB-CGI story.pl access
SecFilterSelective THE_REQUEST "/story\.pl"

# WEB-CGI siteUserMod.cgi access
SecFilterSelective THE_REQUEST "/\.cobalt/siteUserMod/siteUserMod\.cgi"
log,pass

# WEB-CGI cgicso access
SecFilterSelective THE_REQUEST "/cgicso" log,pass

# WEB-CGI nph-publish.cgi access
SecFilterSelective THE_REQUEST "/nph-publish\.cgi" log,pass

# WEB-CGI printenv access
```

```
SecFilterSelective THE_REQUEST "/printenv" log,pass

# WEB-CGI sdbsearch.cgi access
SecFilterSelective THE_REQUEST "/sdbsearch\.cgi" log,pass

# WEB-CGI rpc-nlog.pl access
SecFilterSelective THE_REQUEST "/rpc-nlog\.pl" log,pass

# WEB-CGI rpc-smb.pl access
SecFilterSelective THE_REQUEST "/rpc-smb\.pl" log,pass

# WEB-CGI cart.cgi access
SecFilterSelective THE_REQUEST "/cart\.cgi" log,pass

# WEB-CGI vpasswd.cgi access
SecFilterSelective THE_REQUEST "/vpasswd\.cgi" log,pass

# WEB-CGI alya.cgi access
SecFilterSelective THE_REQUEST "/alya\.cgi" log,pass

# WEB-CGI viralator.cgi access
SecFilterSelective THE_REQUEST "/viralator\.cgi" log,pass

# WEB-CGI smartsearch.cgi access
SecFilterSelective THE_REQUEST "/smartsearch\.cgi" log,pass

# WEB-CGI mrtg.cgi directory traversal attempt
SecFilterSelective THE_REQUEST "/mrtg\.cgi" chain
SecFilter "cfg=/\.\./"

# WEB-CGI overflow.cgi access
SecFilterSelective THE_REQUEST "/overflow\.cgi" log,pass

# WEB-CGI way-board.cgi access
SecFilterSelective THE_REQUEST "/way-board\.cgi" log,pass

# WEB-CGI process_bug.cgi access
SecFilterSelective THE_REQUEST "/process_bug\.cgi" log,pass

# WEB-CGI enter_bug.cgi arbitrary command attempt
SecFilterSelective THE_REQUEST "/enter_bug\.cgi" chain
SecFilter "\;"

# WEB-CGI enter_bug.cgi access
SecFilterSelective THE_REQUEST "/enter_bug\.cgi" log,pass

# WEB-CGI parse_xml.cgi access
SecFilterSelective THE_REQUEST "/parse_xml\.cgi" log,pass

# WEB-CGI streaming server parse_xml.cgi access
SecFilter "/parse_xml\.cgi" log,pass

# WEB-CGI album.pl access
SecFilter "/album\.pl" log,pass

# WEB-CGI chipcfg.cgi access
SecFilterSelective THE_REQUEST "/chipcfg\.cgi" log,pass
```

```
# WEB-CGI ikonboard.cgi access
SecFilterSelective THE_REQUEST "/ikonboard\.cgi" log,pass

# WEB-CGI swsrv.cgi access
SecFilterSelective THE_REQUEST "/srsrv\.cgi" log,pass

# WEB-CLIENT Outlook EML access
SecFilterSelective THE_REQUEST "\.eml"

# WEB-CLIENT XMLHttpRequest attempt
SecFilter "file\://"

# WEB-CLIENT readme.eml download attempt
SecFilterSelective THE_REQUEST "/readme\.eml"

# WEB-CLIENT readme.eml autoload attempt
SecFilter "window\.open\(\"readme\.eml\""

# WEB-CLIENT Javascript document.domain attempt
SecFilter "document\.domain\("

# WEB-CLIENT Javascript URL host spoofing attempt
SecFilter "javascript\://"

# WEB-MISC cross site scripting attempt
SecFilter "<SCRIPT>"

# WEB-MISC cross site scripting \(img src=javascript\) attempt
SecFilter "img src=javascript"

# WEB-MISC Cisco IOS HTTP configuration attempt
SecFilterSelective THE_REQUEST "/exec/"

# WEB-MISC Netscape Enterprise DOS
SecFilter "REVLOG / "

# WEB-MISC Netscape Enterprise directory listing attempt
SecFilter "INDEX "

# WEB-MISC iPlanet GETPROPERTIES attempt
SecFilter "GETPROPERTIES"

# WEB-MISC weblogic view source attempt
SecFilterSelective THE_REQUEST "\.js\x70"

# WEB-MISC Tomcat directory traversal attempt
SecFilterSelective THE_REQUEST "\x00\.jsp"

# WEB-MISC Tomcat view source attempt
SecFilterSelective THE_REQUEST "\x252ejsp"

# WEB-MISC xp_enumdsn attempt
SecFilter "xp_enumdsn"

# WEB-MISC xp_filelist attempt
SecFilter "xp_filelist"
```

```
# WEB-MISC xp_availablemedia attempt
SecFilter "xp_availablemedia"

# WEB-MISC xp_cmdshell attempt
SecFilter "xp_cmdshell"

# WEB-MISC xp_regread attempt
SecFilter "xp_regread" log,pass

# WEB-MISC xp_regwrite attempt
SecFilter "xp_regwrite" log,pass

# WEB-MISC xp_regdeletekey attempt
SecFilter "xp_regdeletekey" log,pass

# WEB-MISC WebDAV search access
SecFilter "SEARCH " log,pass

# WEB-MISC .htpasswd access
SecFilter "\.htpasswd"

# WEB-MISC queryhit.htm access
SecFilterSelective THE_REQUEST "/samples/search/queryhit\.htm" log,pass

# WEB-MISC WebDAV propfind access
SecFilter "xmlns\:a=\"DAV\">" log,pass

# WEB-MISC unify eWave ServletExec upload
SecFilterSelective THE_REQUEST
"/servlet/com\.unify\.servletexec\.UploadServlet"

# WEB-MISC Netscape Servers suite DOS
SecFilterSelective THE_REQUEST "/dsgw/bin/search\?context="

# WEB-MISC amazon 1-click cookie theft
SecFilter "ref\x3Cscript\x20language\x3D\x22Javascript"

# WEB-MISC unify eWave ServletExec DOS
SecFilterSelective THE_REQUEST "/servlet/ServletExec" log,pass

# WEB-MISC Allaire JRUN DOS attempt
SecFilterSelective THE_REQUEST "servlet/\.\.\.\.\.\.\."

# WEB-MISC ICQ Webfront HTTP DOS
SecFilterSelective THE_REQUEST "\?\?\?\?\?\?\?\?\?\?"

# WEB-MISC Nessus 404 probe
SecFilterSelective THE_REQUEST "/nessus_is_probing_you_"

# WEB-MISC Netscape admin passwd
SecFilterSelective THE_REQUEST "/admin-serv/config/admpw"

# WEB-MISC BigBrother access
SecFilterSelective THE_REQUEST "/bb-hostsvc\.sh\?HOSTSVC"

# WEB-MISC ftp.pl attempt
```

```
SecFilterSelective THE_REQUEST "/ftp\.pl\?dir=\.\./\.\."

# WEB-MISC ftp.pl access
SecFilterSelective THE_REQUEST "/ftp\.pl" log,pass

# WEB-MISC Tomcat server snoop access
SecFilterSelective THE_REQUEST "\.snp"

# WEB-MISC apache source.asp file access
SecFilterSelective THE_REQUEST "/site/eg/source\.asp"

# WEB-MISC Tomcat server exploit access
SecFilterSelective THE_REQUEST "/contextAdmin/contextAdmin\.html"

# WEB-MISC http directory traversal
SecFilter "\.\.\\"

# WEB-MISC ICQ webserver DOS
SecFilterSelective THE_REQUEST "\.html/\.\.\.\.\."

# WEB-MISC ls%20-l
SecFilter "ls\x20-l"

# WEB-MISC mlog.phtml access
SecFilterSelective THE_REQUEST "/mlog\.phtml"

# WEB-MISC mylog.phtml access
SecFilterSelective THE_REQUEST "/mylog\.phtml"

# WEB-MISC /etc/passwd
SecFilter "/etc/passwd"

# WEB-MISC ?PageServices access
SecFilterSelective THE_REQUEST "\?PageServices"

# WEB-MISC Ecommerce check.txt access
SecFilterSelective THE_REQUEST "/config/check\.txt"

# WEB-MISC webcart access
SecFilterSelective THE_REQUEST "/webcart/"

# WEB-MISC AuthChangeUrl access
SecFilterSelective THE_REQUEST "_AuthChangeUrl\?"

# WEB-MISC convert.bas access
SecFilterSelective THE_REQUEST "/scripts/convert\.bas"

# WEB-MISC cpshost.dll access
SecFilterSelective THE_REQUEST "/scripts/cpshost\.dll"

# WEB-MISC .htaccess access
SecFilter "\.htaccess"

# WEB-MISC .wwwacl access
SecFilterSelective THE_REQUEST "\.wwwacl"

# WEB-MISC .wwwacl access
```

```
SecFilterSelective THE_REQUEST "\.www_acl"

# WEB-MISC cd..
SecFilter "cd\.\."

# WEB-MISC guestbook.pl access
SecFilterSelective THE_REQUEST "/guestbook\.pl"

# WEB-MISC handler access
SecFilterSelective THE_REQUEST "/handler" log,pass

# WEB-MISC /.... access
SecFilter "/\.\.\.\."

# WEB-MISC ///cgi-bin access
SecFilterSelective THE_REQUEST "///cgi-bin"

# WEB-MISC /cgi-bin/// access
SecFilterSelective THE_REQUEST "/cgi-bin///"

# WEB-MISC /~root access
SecFilterSelective THE_REQUEST "/~root"

# WEB-MISC /~ftp access
SecFilterSelective THE_REQUEST "/~ftp"

# WEB-MISC Ecommerce import.txt access
SecFilterSelective THE_REQUEST "/config/import\.txt"

# WEB-MISC cat%20 access
SecFilter "cat\x20"

# WEB-MISC Ecommerce import.txt access
SecFilterSelective THE_REQUEST "/orders/import\.txt"

# WEB-MISC Ecommerce checks.txt access
SecFilterSelective THE_REQUEST "/orders/checks\.txt"

# WEB-MISC Netscape PublishingXpert access
SecFilterSelective THE_REQUEST "/PSUser/PSCOErrPage\.htm" log,pass

# WEB-MISC webplus access
SecFilterSelective THE_REQUEST "/webplus\?script"

# WEB-MISC Netscape dir index wp
SecFilterSelective THE_REQUEST "\?wp-"

# WEB-MISC shopping cart access
SecFilterSelective THE_REQUEST "/quikstore\.cfg"

# WEB-MISC Novell Groupwise gwweb.exe attempt
SecFilterSelective THE_REQUEST "/GWWEB\.EXE\?HELP="

# WEB-MISC Novell Groupwise gwweb.exe access
SecFilter "/GWWEB\.EXE"

# WEB-MISC ws_ftp.ini access
```

```
SecFilterSelective THE_REQUEST "/ws_ftp\.ini"

# WEB-MISC rpm_query access
SecFilterSelective THE_REQUEST "/rpm_query"

# WEB-MISC mall log order access
SecFilterSelective THE_REQUEST "/mall_log_files/order\.log"

# WEB-MISC architext_query.pl access
SecFilterSelective THE_REQUEST "/ews/architext_query\.pl"

# WEB-MISC wwwboard.pl access
SecFilterSelective THE_REQUEST "/wwwboard\.pl"

# WEB-MISC order.log access
SecFilterSelective THE_REQUEST "/admin_files/order\.log"

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-verify-link"

# WEB-MISC Annex Terminal DOS attempt
SecFilterSelective THE_REQUEST "/ping\?query="

# WEB-MISC cgitest.exe access
SecFilterSelective THE_REQUEST "/cgitest\.exe" log,pass

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-cs-dump"

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-ver-info"

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-ver-diff"

# WEB-MISC SalesLogix Eviewer web command attempt
SecFilterSelective THE_REQUEST "/slxweb\.dll/admin\?command="

# WEB-MISC SalesLogix Eviewer access
SecFilterSelective THE_REQUEST "/slxweb\.dll" log,pass

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-start-ver"

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-stop-ver"

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-uncheckout"

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-html-rend"

# WEB-MISC Trend Micro OfficeScan attempt
SecFilterSelective THE_REQUEST "event="

# WEB-MISC Trend Micro OfficeScan access
```

```
SecFilterSelective THE_REQUEST "/officescan/cgi/jdkRqNotify\.exe"

# WEB-MISC oracle web arbitrary command execution attempt
SecFilterSelective THE_REQUEST "\?&"

# WEB-MISC oracle web application server access
SecFilterSelective THE_REQUEST "/ows-bin/" log,pass

# WEB-MISC Netscape Enterprise Server directory view
SecFilterSelective THE_REQUEST "\?wp-usr-prop"

# WEB-MISC search.vts access
SecFilterSelective THE_REQUEST "/search\.vts"

# WEB-MISC htgrep attempt
SecFilterSelective THE_REQUEST "/htgrep" chain
SecFilter "hdr=/"

# WEB-MISC htgrep access
SecFilterSelective THE_REQUEST "/htgrep" log,pass

# WEB-MISC .nsconfig access
SecFilterSelective THE_REQUEST "/\.nsconfig"

# WEB-MISC Admin_files access
SecFilterSelective THE_REQUEST "/admin_files"

# WEB-MISC backup access
SecFilterSelective THE_REQUEST "/backup"

# WEB-MISC intranet access
SecFilterSelective THE_REQUEST "/intranet/"

# WEB-MISC filemail access
SecFilterSelective THE_REQUEST "/filemail"

# WEB-MISC plusmail access
SecFilterSelective THE_REQUEST "/plusmail"

# WEB-MISC adminlogin access
SecFilterSelective THE_REQUEST "/adminlogin"

# WEB-MISC ultraboard access
SecFilterSelective THE_REQUEST "/ultraboard"

# WEB-MISC musicat empower attempt
SecFilterSelective THE_REQUEST "/empower\?DB="

# WEB-MISC musicat empower access
SecFilterSelective THE_REQUEST "/empower" log,pass

# WEB-MISC ROADS search.pl attempt
SecFilterSelective THE_REQUEST "/ROADS/cgi-bin/search\.pl" chain
SecFilter "form="

# WEB-MISC Tomcat sourecode view
SecFilterSelective THE_REQUEST "\.js\x2570"
```

```
# WEB-MISC Tomcat sourecode view
SecFilterSelective THE_REQUEST "\.j\x2573p"

# WEB-MISC Tomcat sourecode view
SecFilterSelective THE_REQUEST "\.\x256Asp"

# WEB-MISC SWEditServlet directory traversal attempt
SecFilterSelective THE_REQUEST "/SWEditServlet" chain
SecFilter "template=\.\./\.\./\.\./"

# WEB-MISC SWEditServlet access
SecFilterSelective THE_REQUEST "/SWEditServlet"

# WEB-MISC whisker HEAD/./
SecFilter "HEAD/\./"

# WEB-MISC long basic authorization string
SecFilter "Authorization\: Basic "

# WEB-MISC sml3com access
SecFilterSelective THE_REQUEST "/graphics/sml3com" log,pass

# WEB-MISC http directory traversal
SecFilter "\.\./"

# WEB-MISC sadmind worm access
SecFilter "GET x HTTP/1\.0"

# WEB-MISC jrun directory browse attempt
SecFilterSelective THE_REQUEST "/\x3f\.jsp"

# WEB-MISC mod-plsql administration access
SecFilterSelective THE_REQUEST "/admin_/" log,pass

# WEB-MISC Phorecast remote code execution attempt
SecFilter "includedir="

# WEB-MISC viewcode access
SecFilterSelective THE_REQUEST "/viewcode"

# WEB-MISC showcode access
SecFilterSelective THE_REQUEST "/showcode"

# WEB-MISC .history access
SecFilterSelective THE_REQUEST "/\.history"

# WEB-MISC .bash_history access
SecFilterSelective THE_REQUEST "/\.bash_history"

# WEB-MISC /~nobody access
SecFilterSelective THE_REQUEST "/~nobody"

# WEB-MISC RBS ISP /newuser  directory traversal attempt
SecFilterSelective THE_REQUEST "/newuser\?Image=\.\./\.\."

# WEB-MISC RBS ISP /newuser access
```

```
SecFilterSelective THE_REQUEST "/newuser" log,pass

# WEB-MISC *%0a.pl access
SecFilterSelective THE_REQUEST "/*\x0a\.pl"

# WEB-MISC PCCS mysql database admin tool access
SecFilter "pccsmysqladm/incs/dbconnect\.inc"

# WEB-MISC .DS_Store access
SecFilterSelective THE_REQUEST "/\.DS_Store" log,pass

# WEB-MISC .FBCIndex access
SecFilterSelective THE_REQUEST "/\.FBCIndex" log,pass

# WEB-MISC ExAir access
SecFilterSelective THE_REQUEST "/exair/search/" log,pass

# WEB-MISC apache ?M=D directory list attempt
SecFilterSelective THE_REQUEST "/\?M=D" log,pass

# WEB-MISC server-info access
SecFilterSelective THE_REQUEST "/server-info" log,pass

# WEB-MISC server-status access
SecFilterSelective THE_REQUEST "/server-status" log,pass

# WEB-MISC ans.pl attempt
SecFilterSelective THE_REQUEST "/ans\.pl\?p=\.\./\.\./"

# WEB-MISC ans.pl access
SecFilterSelective THE_REQUEST "/ans\.pl" log,pass

# WEB-MISC AxisStorpoint CD attempt
SecFilterSelective THE_REQUEST "/cd/\.\./config/html/cnf_gi\.htm"

# WEB-MISC Axis Storpoint CD access
SecFilterSelective THE_REQUEST "/config/html/cnf_gi\.htm" log,pass

# WEB-MISC basilix sendmail.inc access
SecFilterSelective THE_REQUEST "/inc/sendmail\.inc" log,pass

# WEB-MISC basilix mysql.class access
SecFilterSelective THE_REQUEST "/class/mysql\.class" log,pass

# WEB-MISC BBoard access
SecFilterSelective THE_REQUEST "/servlet/sunexamples\.BBoardServlet"
log,pass

# WEB-MISC Cisco Catalyst command execution attempt
SecFilterSelective THE_REQUEST "/exec/show/config/cr" log,pass

# WEB-MISC Cisco /%% DOS attempt
SecFilterSelective THE_REQUEST "/%%"

# WEB-MISC /CVS/Entries access
SecFilterSelective THE_REQUEST "/CVS/Entries" log,pass
```

```
# WEB-MISC cvsweb version access
SecFilterSelective THE_REQUEST "/cvsweb/version" log,pass

# WEB-MISC /doc/packages access
SecFilterSelective THE_REQUEST "/doc/packages" log,pass

# WEB-MISC /doc/ access
SecFilterSelective THE_REQUEST "/doc/" log,pass

# WEB-MISC ?open access
SecFilterSelective THE_REQUEST "\?open" log,pass

# WEB-MISC login.htm attempt
SecFilterSelective THE_REQUEST "/login\.htm\?password=" log,pass

# WEB-MISC login.htm access
SecFilterSelective THE_REQUEST "/login\.htm" log,pass

# WEB-MISC DELETE attempt
SecFilter "DELETE " log,pass

# WEB-MISC /home/ftp access
SecFilterSelective THE_REQUEST "/home/ftp" log,pass

# WEB-MISC /home/www access
SecFilterSelective THE_REQUEST "/home/www" log,pass

# WEB-MISC global.inc access
SecFilterSelective THE_REQUEST "/global\.inc"

# WEB-MISC SecureSite authentication bypass attempt
SecFilter "secure_site, ok"

# WEB-MISC b2 arbitrary command execution attempt
SecFilterSelective THE_REQUEST "/b2/b2-include/" chain
SecFilter "http\://"

# WEB-MISC b2 access
SecFilterSelective THE_REQUEST "/b2/b2-include/" chain
SecFilter "http\://"

# WEB-MISC PIX firewall manager directory traversal attempt
SecFilterSelective THE_REQUEST "/\.\./\.\./"

# WEB-MISC iChat directory traversal attempt
SecFilterSelective THE_REQUEST "/\.\./\.\./" log,pass

# WEB-MISC Delegate whois overflow attempt
SecFilter "whois\://" log,pass

# WEB-MISC nstelemetry.adp access
SecFilterSelective THE_REQUEST "/nstelemetry\.adp" log,pass

# WEB-MISC Compaq Insight directory traversal
SecFilterSelective THE_REQUEST "\.\./"

# WEB-MISC VirusWall catinfo access
```

```
SecFilterSelective THE_REQUEST "/catinfo"

# WEB-MISC VirusWall catinfo access
SecFilterSelective THE_REQUEST "/catinfo"

# WEB-MISC Apache Chunked-Encoding worm attempt
SecFilter "CCCCCCC\: AAAAAAAAAAAAAAAAAAA"

# WEB-MISC Transfer-Encoding\: chunked
SecFilter "chunked"

# WEB-MISC CISCO VoIP DOS ATTEMPT
SecFilterSelective THE_REQUEST "/StreamingStatistics"

# WEB-MISC IBM Net.Commerce orderdspc.d2w access
SecFilterSelective THE_REQUEST "/ncommerce3/ExecMacro/orderdspc\.d2w"
log,pass

# WEB-MISC WEB-INF access
SecFilterSelective THE_REQUEST "/WEB-INF" log,pass

# WEB-MISC Tomcat servlet mapping cross site scripting attempt
SecFilterSelective THE_REQUEST "/org\.apache\."

# WEB-MISC iPlanet Search directory traversal attempt
SecFilterSelective THE_REQUEST "/search" chain
SecFilter "\.\./\.\./"

# WEB-MISC Tomcat TroubleShooter servlet access
SecFilterSelective THE_REQUEST "/examples/servlet/TroubleShooter"
log,pass

# WEB-MISC Tomcat SnoopServlet servlet access
SecFilterSelective THE_REQUEST "/examples/servlet/SnoopServlet"
log,pass

# WEB-MISC jigsaw dos attempt
SecFilterSelective THE_REQUEST "/servlet/con"

# WEB-MISC Macromedia SiteSpring cross site scripting attempt
SecFilterSelective THE_REQUEST "<script"

# WEB-MISC mailman cross site scripting attempt
SecFilterSelective THE_REQUEST "<script"

# WEB-MISC webalizer access
SecFilterSelective THE_REQUEST "/webalizer/" log,pass

# WEB-MISC webcart-lite access
SecFilterSelective THE_REQUEST "/webcart-lite/" log,pass

# WEB-MISC active.log access
SecFilterSelective THE_REQUEST "/active\.log" log,pass

# WEB-MISC robots.txt access
SecFilterSelective THE_REQUEST "/robots\.txt" log,pass
```

```
# WEB-MISC robot.txt access
SecFilterSelective THE_REQUEST "/robot\.txt" log,pass

# WEB-MISC CISCO PIX Firewall Manager directory traversal attempt
SecFilterSelective THE_REQUEST "/pixfir~1/how_to_login\.html"

# WEB-MISC Sun JavaServer default password login attempt
SecFilterSelective THE_REQUEST "/servlet/admin" chain
SecFilter "ae9f86d6beaa3f9ecb9a5b7e072a4138"

# WEB-MISC Linksys router default password login attempt \(\:admin\)
SecFilter "Authorization\: Basic OmFkbWlu"

# WEB-MISC Linksys router default password login attempt
\(admin\:admin\)
SecFilter "YWRtaW46YWRtaW4"

# WEB-MISC Oracle XSQLConfig.xml access
SecFilterSelective THE_REQUEST "/XSQLConfig\.xml" log,pass

# WEB-MISC Oracle Dynamic Monitoring Services (dms) access
SecFilterSelective THE_REQUEST "/dms0" log,pass

# WEB-MISC globals.jsa access
SecFilterSelective THE_REQUEST "/globals\.jsa" log,pass

# WEB-MISC Oracle Java Process Manager access
SecFilterSelective THE_REQUEST "/oprocmgr-status" log,pass

# WEB-MISC /Carello/add.exe access
SecFilterSelective THE_REQUEST "/Carello/add\.exe" log,pass

# WEB-MISC ion-p access
SecFilterSelective THE_REQUEST "/ion-p" log,pass

# WEB-MISC answerbook2 admin attempt
SecFilterSelective THE_REQUEST "/cgi-bin/admin/admin" log,pass

# WEB-MISC answerbook2 arbitrary command execution attempt
SecFilterSelective THE_REQUEST "/ab2/" chain
SecFilter "\;"

# WEB-MISC perl post attempt
SecFilterSelective THE_REQUEST "/perl/" chain
SecFilter "POST"

# WEB-MISC TRACE attempt
SecFilter "TRACE"

# WEB-MISC DB4Web access
SecFilterSelective THE_REQUEST "/DB4Web/" log,pass

# WEB-MISC iPlanet .perf access
SecFilterSelective THE_REQUEST "/\.perf" log,pass

# WEB-MISC Demarc SQL injection attempt
SecFilterSelective THE_REQUEST "/dm/demarc" chain
```

```
SecFilter "'" log,pass

# WEB-MISC Lotus Notes .csp script source download attempt
#SecFilterSelective THE_REQUEST "\.csp" chain
#SecFilter "\."

# WEB-MISC Lotus Notes .pl script source download attempt
#SecFilterSelective THE_REQUEST "\.pl" chain
#SecFilter "\."

# WEB-MISC BitKeeper arbitrary command attempt
SecFilterSelective THE_REQUEST "/diffs/" chain
SecFilter "'"

# WEB-MISC chip.ini access
SecFilterSelective THE_REQUEST "/chip\.ini" log,pass

# WEB-MISC lyris.pl access
SecFilterSelective THE_REQUEST "/lyris\.pl" log,pass

# WEB-MISC globals.pl access
SecFilterSelective THE_REQUEST "/globals\.pl" log,pass

# WEB-MISC philboard.mdb access
SecFilterSelective THE_REQUEST "/philboard\.mdb" log,pass

# WEB-MISC philboard_admin.asp authentication bypass attempt
SecFilterSelective THE_REQUEST "/philboard_admin\.asp" chain
SecFilter "philboard_admin=True"

# WEB-MISC philboard_admin.asp access
SecFilterSelective THE_REQUEST "/philboard_admin\.asp" log,pass

# WEB-MISC logicworks.ini access
SecFilterSelective THE_REQUEST "/logicworks\.ini" log,pass

# WEB-MISC /*.shtml access
SecFilterSelective THE_REQUEST "/*\.shtml" log,pass

# WEB-MISC mod_gzip_status access
SecFilterSelective THE_REQUEST "/mod_gzip_status" log,pass

# WEB-PHP bb_smilies.php access
SecFilterSelective THE_REQUEST "/bb_smilies\.php" log,pass

# WEB-PHP squirrel mail spell-check arbitrary command attempt
SecFilterSelective THE_REQUEST
"/squirrelspell/modules/check_me\.mod\.php" chain
SecFilter "SQSPELL_APP\["

# WEB-PHP squirrel mail theme arbitrary command attempt
SecFilterSelective THE_REQUEST "/left_main\.php" chain
SecFilter "cmdd="

# WEB-PHP DNSTools administrator authentication bypass attempt
SecFilterSelective THE_REQUEST "/dnstools\.php" chain
SecFilter "user_dnstools_administrator=true"
```

```
# WEB-PHP DNSTools authentication bypass attempt
SecFilterSelective THE_REQUEST "/dnstools\.php" chain
SecFilter "user_logged_in=true"

# WEB-PHP DNSTools access
SecFilterSelective THE_REQUEST "/dnstools\.php" log,pass

# WEB-PHP Blahz-DNS dostuff.php modify user attempt
SecFilterSelective THE_REQUEST "/dostuff\.php\?action=modify_user"

# WEB-PHP Blahz-DNS dostuff.php access
SecFilterSelective THE_REQUEST "/dostuff\.php" log,pass

# WEB-PHP Messagerie supp_membre.php access
SecFilterSelective THE_REQUEST "/supp_membre\.php" log,pass

# WEB-PHP php.exe access
SecFilterSelective THE_REQUEST "/php\.exe" log,pass

# WEB-PHP directory.php arbitrary command attempt
SecFilterSelective THE_REQUEST "/directory\.php" chain
SecFilter "\;"

# WEB-PHP directory.php access
SecFilterSelective THE_REQUEST "/directory\.php"

# WEB-PHP PHP-Wiki cross site scripting attempt
SecFilterSelective THE_REQUEST "<script"

# WEB-PHP phpbb quick-reply.php arbitrary command attempt
SecFilterSelective THE_REQUEST "/quick-reply\.php" chain
SecFilter "phpbb_root_path="

# WEB-PHP phpbb quick-reply.php access
SecFilterSelective THE_REQUEST "/quick-reply\.php" log,pass

# WEB-PHP read_body.php access attempt
SecFilterSelective THE_REQUEST "/read_body\.php" log,pass

# WEB-PHP calendar.php access
SecFilterSelective THE_REQUEST "/calendar\.php" log,pass

# WEB-PHP edit_image.php access
SecFilterSelective THE_REQUEST "/edit_image\.php" log,pass

# WEB-PHP readmsg.php access
SecFilterSelective THE_REQUEST "/readmsg\.php" log,pass

# WEB-PHP external include path
SecFilterSelective THE_REQUEST "\.php" chain
SecFilter "path=http\://"

# WEB-PHP Phorum admin access
SecFilterSelective THE_REQUEST "/admin\.php3"

# WEB-PHP piranha passwd.php3 access
```

```
SecFilterSelective THE_REQUEST "/passwd\.php3"

# WEB-PHP Phorum read access
SecFilterSelective THE_REQUEST "/read\.php3"

# WEB-PHP Phorum violation access
SecFilterSelective THE_REQUEST "/violation\.php3"

# WEB-PHP Phorum code access
SecFilterSelective THE_REQUEST "/code\.php3"

# WEB-PHP admin.php file upload attempt
SecFilterSelective THE_REQUEST "/admin\.php" chain
SecFilter "file_name="

# WEB-PHP admin.php access
SecFilterSelective THE_REQUEST "/admin\.php"

# WEB-PHP smssend.php access
SecFilterSelective THE_REQUEST "/smssend\.php" log,pass

# WEB-PHP PHP-Nuke remote file include attempt
SecFilterSelective THE_REQUEST "index\.php" chain
SecFilter "file=http\://"

# WEB-PHP Phorum /support/common.php attempt
SecFilterSelective THE_REQUEST "/support/common\.php" chain
SecFilter "ForumLang=\.\./"

# WEB-PHP Phorum /support/common.php access
SecFilterSelective THE_REQUEST "/support/common\.php"

# WEB-PHP Phorum authentication access
SecFilter "PHP_AUTH_USER=boogieman"

# WEB-PHP strings overflow
SecFilterSelective THE_REQUEST "\?STRENGUR"

# WEB-PHP PHPLIB remote command attempt
SecFilter "_PHPLIB\[libdir\]"

# WEB-PHP PHPLIB remote command attempt
SecFilterSelective THE_REQUEST "/db_mysql\.inc"

# WEB-PHP Mambo uploadimage.php upload php file attempt
SecFilterSelective THE_REQUEST "/uploadimage\.php" chain
SecFilter "\.php"

# WEB-PHP Mambo upload.php upload php file attempt
SecFilterSelective THE_REQUEST "/upload\.php" chain
SecFilter "\.php"

# WEB-PHP Mambo uploadimage.php access
SecFilterSelective THE_REQUEST "/uploadimage\.php" log,pass

# WEB-PHP Mambo upload.php access
SecFilterSelective THE_REQUEST "/upload\.php" log,pass
```

```
# WEB-PHP phpBB privmsg.php access
SecFilterSelective THE_REQUEST "/privmsg\.php" log,pass

# WEB-PHP p-news.php access
SecFilterSelective THE_REQUEST "/p-news\.php" log,pass

# WEB-PHP shoutbox.php directory traversal attempt
SecFilterSelective THE_REQUEST "/shoutbox\.php" chain
SecFilter "\.\./"

# WEB-PHP shoutbox.php access
SecFilterSelective THE_REQUEST "/shoutbox\.php" log,pass

# WEB-PHP b2 cafelog gm-2-b2.php remote command execution attempt
SecFilterSelective THE_REQUEST "/gm-2-b2\.php" chain
SecFilter "b2inc=http"

# WEB-PHP b2 cafelog gm-2-b2.php access
SecFilterSelective THE_REQUEST "/gm-2-b2\.php" log,pass

# WEB-PHP TextPortal admin.php default password (admin) attempt
SecFilterSelective THE_REQUEST "/admin\.php" chain
SecFilter "password=admin" log,pass

# WEB-PHP TextPortal admin.php default password (12345) attempt
SecFilterSelective THE_REQUEST "/admin\.php" chain
SecFilter "password=12345" log,pass

# WEB-PHP BLNews objects.inc.php4 remote command execution attempt
SecFilterSelective THE_REQUEST "/objects\.inc\.php4" chain
SecFilter "Server\[path\]=http"

# WEB-PHP BLNews objects.inc.php4 access
SecFilterSelective THE_REQUEST "/objects\.inc\.php4" log,pass

# WEB-PHP Turba status.php access
SecFilterSelective THE_REQUEST "/turba/status\.php" log,pass

# WEB-PHP ttCMS header.php remote command execution attempt
SecFilterSelective THE_REQUEST "/admin/templates/header\.php" chain
SecFilter "admin_root=http"

# WEB-PHP ttCMS header.php access
SecFilterSelective THE_REQUEST "/admin/templates/header\.php" log,pass

# WEB-PHP test.php access
SecFilterSelective THE_REQUEST "/test\.php" log,pass

# WEB-PHP autohtml.php directory traversal attempt
SecFilterSelective THE_REQUEST "/autohtml\.php" chain
SecFilter "\.\./\.\./"

# WEB-PHP autohtml.php access
SecFilterSelective THE_REQUEST "/autohtml\.php" log,pass

# WEB-PHP ttforum remote command execution attempt
```

```
SecFilterSelective THE_REQUEST "forum/index\.php" chain
SecFilter "template=http"
```