Global Information Assurance Certification Paper

GSEC Gold Certification

VPNScan: Extending the Audit and Compliance Perimeter

Last Revised 12-Sep-2006

Author: Rob VandenBrink
rvandenbrink@metafore.ca
Adviser: Jim Purcell

Table of Contents:

Business Issue

In my work as a security consultant, I have a large number of clients with Remote Access or Remote Computing Security Policies. All of these policies have wording that encompasses some or all of:

- All VPN or Dialup connections to the Corporate Network will be made from Corporately owned hardware
- Any Internet connection made from a non-Corporate location will use a properly configured (or Corporate owned and configured) hardware firewall
- All Corporate owned laptops will have a Corporate approved, properly configured personal firewall installed.

However, what struck me was that none of these companies had a good method of auditing these policies to ensure compliance. After some research, it was found that there are not a lot of solutions to accomplish this without a large budget commitment. The tool outlined in this paper addresses this issue, and has been deployed at several customer sites to date.

Solution Overview

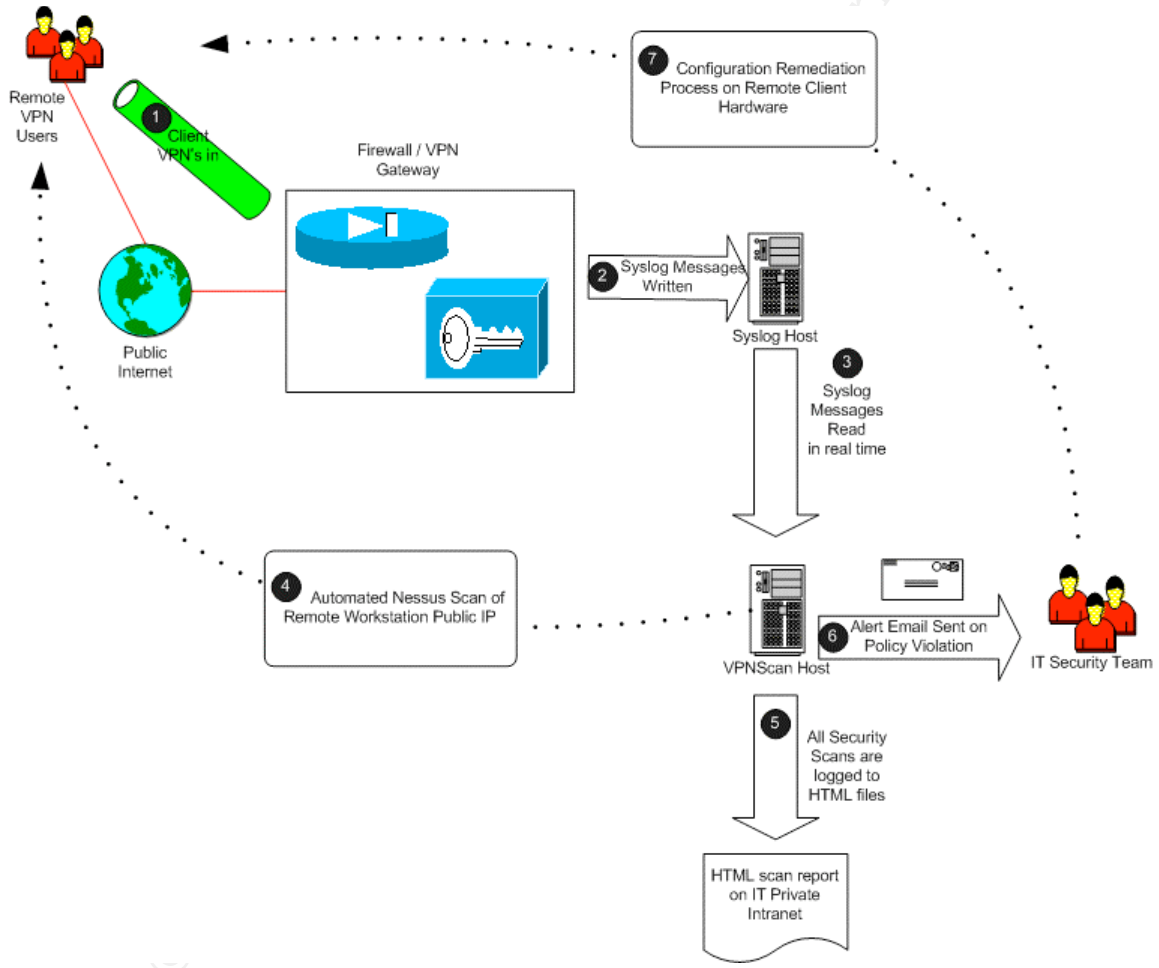
VPNSCAN uses several tools to accomplish this goal:

1. Swatch is used to monitor syslog from the vpn gateway. It waits for a successful VPN connection, then feeds that entire syslog event to a shell script
2. The shell script parses out the command line arguments, then uses Nessus to scan the external (public internet ip) of the person who just vpn'd in.
3. The Nessus scan is saved to an HTML report file
4. If the Nessus scan indicates a violation of company policy (ie if the remote firewall fails the scan on some or all identified tests), an alert email is sent to the IT team responsible for security.

This paper outlines specifically how VPNSCAN was built, with policy and implementation issues found in various customer environments.

Solution Overview Diagram

This diagram outlines the process flows, showing all processes, data and people involved. Processes are numbered in order, starting with a client VPNing into the corporate network.



Audit, Policy Violations and Compliance

The purpose of this tool is to audit remote ip addresses as clients vpn in, to test them for compliance against an existing Remote Access Policy (or a Remote Access section or paragraph in an Acceptable Use Policy).

Policies of this type typically state something similar to “Users who access the corporate network remotely will not connect to the public internet without a Corporate approved firewall”. Corporate approved firewalls typically are a short list of home/SOHO hardware firewalls, some policies also list personal (software based) firewalls.

The challenge is to Audit a remote client’s public ip address against this phrase. VPNSCAN employs the Nessus security scanner running on a Linux host, so a reasonable toolset is available to accomplish this task. It was decided to simply use grep to look for strings in the Nessus output to evaluate for two conditions:

1. The remote user has no Firewall
2. The remote user’s firewall is configured incorrectly

The grep search strings to evaluate these conditions are outlined below:

The presense of any of the ports or strings below would normally indicate a Microsoft Windows PC with no firewall present, or a very badly configured firewall:

Service	Description
445/tcp	Microsoft-DS
135/tcp	DCE Endpoint Resolution
139/tcp	Netbios Session Service
137/udp	WINS Name Service
138/udp	Netbios Datagram Service
135/udp	DCE Endpoint Resolution
3389/tcp	RDP (Remote Desktop Protocol)
DCE/RPC	DCE / RPC
389/tcp	LDAP

5901/tcp	VNC
5900/tcp	VNC
5631/tcp	PC Anywhere
2967/tcp	Norton Antivirus Corporate

These ports would generally indicate a firewall that has been poorly configured (or in some cases connected with an internal port facing the internet).

Service	**Description**
23/tcp	telnet
6000/tcp 6001/tcp 6002/tcp 6003/tcp	X-Windows
2049/tcp	NFS
2049/udp	NFS

Policy Violation and Remediation

In the application as it's written currently, the only automated result is an email to the IT group responsible for security. In many environments, almost all VPN connections are made after hours, so dealing with events of this type in a timely fashion can be a challenge.

There is a continuum of responses to alerts of the type VPNSCAN generates – a surprising number of IT departments are still completely project-centric and do not apply significant resources to security. In situations like this, alerts are routinely ignored.

In almost all implementations, VPNSCAN alerts are simply dealt with via a face-to-face discussion during business hours the following day, where a policy compliant solution is negotiated. However, other options are certainly available.

VPNSCAN alerts could also be sent directly to the USERID involved in the security event. However, end users do not typically have the appreciation to interpret the alert email into practical action to resolve the issue without assistance. This option is not normally implemented.

If the IT group views events of this type to be of the same importance as hostile intrusion events, then responses to VPNSCAN alerts could result in on-call pager events and phone calls to the users' home or cellphone within a time period defined in the Company Security Procedures.

The example Remote Access Policy (included in the Appendix) outlines penalties for policy violation at a real company. The maximum penalty described is removal of all remote access privileges. All other penalties are purposefully not included in this policy, giving corporate management as much flexibility as possible in dealing with these situations.

A further escalation of response would be to consider an automated response. VPNSCAN could be very easily modified to take advantage of a feature on the Cisco PIX call a “shun list”. Addresses that are “shunned” have any active connections cleared, and any new traffic from them is immediately dropped. Addresses in the shun list remain there until manually removed. Active connections to shunned ip’s are immediately terminated. The most basic syntax (which is probably the most appropriate in this application) is:

```
shun 111.222.33.44
```
(this will shun all traffic from this ip address)

Since there is no “expiry timer” on shuns, in most IDS implementations a timer is placed on shuns in the application, and the IDS application removes the entry.

A “shun module” could be implemented for VPNSCAN using EXPECT or the Net::Telnet perl module. This function can certainly be emulated by updating the inbound access list on the external interface of other VPN Gateways (for instance, VPN 3000s or IOS routers).

There is some very real business risk in deploying any automated security response. Automated responses do not have any appreciation for who might be VPNing in. For instance, it would be undesirable to shun the Finance Vice President during a month-end window, or to shun a key salesperson who might be responding to an RFP due the next morning.

An additional risk in deploying an automated response is that even though SSH is used to deliver the command to the VPN gateway, a valid userid and password with administration rights must be stored in plaintext and available to the VPNSCAN host. If this is done, the VPNSCAN host should be heavily secured, and even then, this may be one of those proverbial “bad ideas”. In many environments cleartext storage of administration level passwords will violate existing security policies.

Decision Points in Design

It was decided to implement the pilot for VPNSCAN on VMWare (http://www.vmware.com), using CentOS (www.CentOS.org) for a guest operating system.

VMware was chosen so that VPNSCAN could be easily moved from to additional clients – the entire solution can be image backed-up to a single DVD, and restored at a different site. Also, no hardware purchase is required if the client has a VMWare environment (at this point most clients who have any written policies and a vpn solution also have at least one VMWare server). If deployed to a physical platform, the memory, CPU and disk requirements are extremely modest for VPNSCAN, almost any workstation or server hardware could be used.

CentOS was chosen for an Operating System because it is a linux distribution that has a goal of binary compatibility with Redhat Enterprise Linux (though they no longer mention Redhat by name on their site). This ensures that the base OS is stable and the binaries are well tested in combination. Using CentOS also makes any future move to a licensed Enterprise OS (namely Redhat), perhaps after a pilot phase, a simple procedure. CentOS is also free, as are all of the tools used in VPNSCAN. This ensures that VPNSCAN can be built for and delivered to clients for testing without worrying about software budgets or license fees of any kind. However, if your client has budget, or especially if you are building VPNSCAN as part of a consulting engagement, ensure that a donation is made to continue the excellent work at CentOS (easiest by paypal).

SWATCH was chosen as it is a simple, widely deployed tool that will monitor syslog logs and trigger configurable events in near-realtime.

NESSUS was chosen as the security scanner because it is still freely available, widely used in the security community, and has an excellent reputation. Nessus could, however, be replaced by any security scanner, and could in fact be replaced or

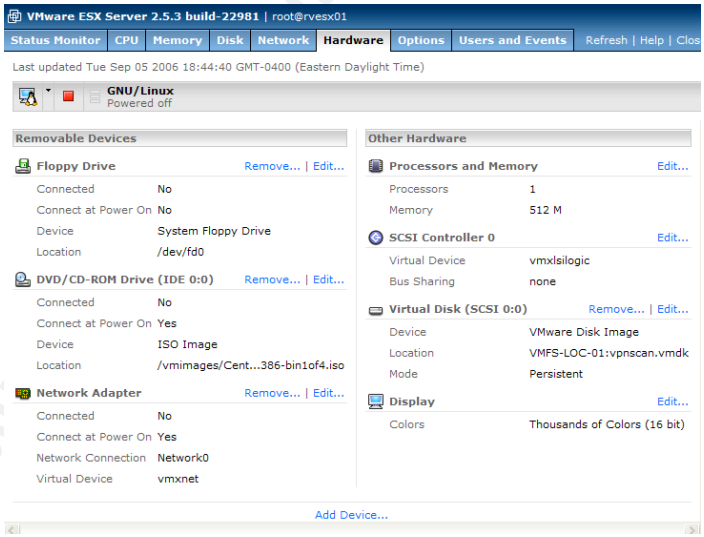
augmented with any other command-line security assessment tool, command-line virus scanner or any other command-line tool that might be required to assess a remote station.

Build Procedure

OS installation

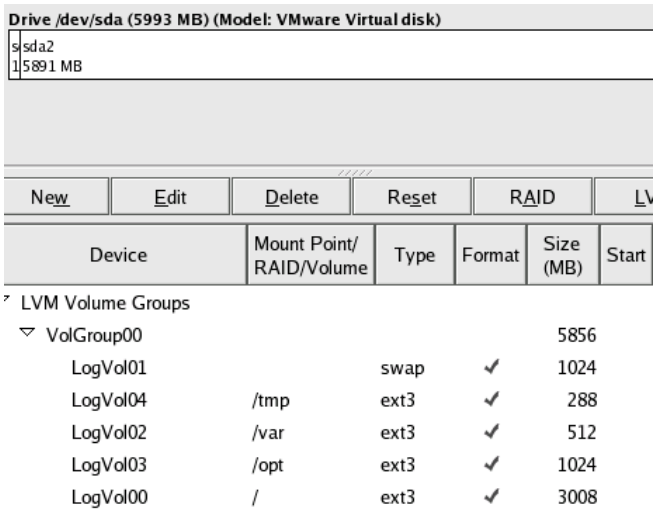
First, a basic OS is required - CentOS in a VMWare virtual machine in this case. The virtual machine was allocated 512MB Dram, and 6GB of disk. The Operating System in the VM was installed it without a GUI. However, VMWARE tools was installed. This improves the console responsiveness, memory and CPU utilization, enables the configuration of a vmxnet network card (amongst other benefits).

Note that the SCSI controller has been changed from the older (default) buslogic to the newer lsilogic driver. This provides a higher-performance disk subsystem, also CentOS no longer ships with the buslogic driver. If this driver is chosen, make the change before you install the OS, or booting the VM will be problematic.



The detailed mechanics of installing and securing Linux will not be covered in this document – there are a large number of excellent resources that outline how to do this well, and requirements will vary depending on the environment. However, key decision points along the way that have a bearing on the final solution will be discussed.

At a minimum, partition out /tmp and /var . Ideally /mnt would also be on it's own partition, but that is no longer an option in this version of CentOS. Instead, partition out /opt (a /opt/mnt directory will be used for the remote mountpoints). This server was built on a 6GB partition. Larger is fine, but since this is server was going into an existing Corporate environment, all of the logs and reports must reside on other (existing) hosts.



The main goal of partitioning is to ensure that system logs do not fill up the root. In the case of the scans themselves, if for some reason the remote mount disappears, the scan reports themselves do not full up the root.
Choose a custom install. Very little is required for this build, past the basic Operating System.

Since there is no GUI, no web server, and this box delivers no network services, the OS firewall was enabled, permitting only SSH inbound.

At the package selection screen, de-select X-Windows, Gnome, Graphical Internet, Office Productivity and Sound/Video, Graphics and Printing
However, ensure that Development tools is selected – GCC will be required for some of the swatch dependancies, and perl will certainly be required. Also ensure that you have smbclient selected (under System Tools) if you plan to do remote mounts to Windows hosts. Ntpd will also be required to provide accurate time.

This can probably be trimmed further, but the final footprint resulting from these selections fits nicely on the 6GB image allocated for this build.

Linux Configs

After installation, install vmware tools, then run netconfig to configure the network card. To bring the nic up, run “ifup eth0” or “service network restart”.

Ensure that a host name and domain are set (/etc/sysconfig/network and /etc/hosts).

Sendmail is *NOT* required to send the alert emails (or any mail for that matter). Disable sendmail entirely using chkconfig (or edit the init directories manually if the preference is for doing things “the hard way”). Disabling sendmail is an excellent security measure on almost all linux hosts (or at least on any linux host that isn’t *receiving* mail)

```
[root@vpnscan ~]# chkconfig --list sendmail
sendmail        0:off   1:off   2:on   3:on   4:on 5:on 6:off
[root@vpnscan ~]# chkconfig --level 2345 sendmail off
[root@vpnscan ~]# chkconfig --list sendmail
sendmail        0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@vpnscan ~]#
```

What *IS* required to send mail is a simple update /etc/submit.cf to point it to a valid mail host, as sendmail is no longer routing SMTP mail via DNS. In this case we’ll point it at mail bastion host/spam filter:

By default, submit.cf points to localhost, so that sendmail sends it.

```
D{MTAHost}[127.0.0.1]
```

This is the modified entry

```
D{MTAHost}[172.16.1.22]
```

However, if for one reason or another an email does not leave the linux host (for instance if the target host is down at the time of a send), it will be queued. To process mail queues, a cron job will be required – add this line to /etc/crontab to process the mail queues at the top of each hour:

```
0 * * * * root /usr/sbin/sendmail -Ac -q
```

Also, printing will not be required from this host, so disable CUPS. Even if printing support is deselected during the OS install, CUPS is installed and enabled. To see what runlevels CUPS is active on:

```
[root@vpnscan ~]# chkconfig --list cups
cups            0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

To disable CUPS on all runlevels:

```
[root@vpnscan ~]# chkconfig --level 2345 cups off
[root@vpnscan ~]# chkconfig --list cups
cups            0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@vpnscan ~]# service cups stop
```

Package Installation

Download and install the latest Nessus version (installation from the OS CD's is not recommended). Note that the Nessus install will alert you to performance issues in the VMWare environment. In this case it is not normally an issue, as the scan volume should be relatively low (in most environments less than 1,000 scans per day).

```
[root@vpnscan install]# rpm -iv Nessus-3.0.3-es4.i386.rpm
Preparing packages for installation...
Nessus-3.0.3-es4


**** This host seems to be running under VMware.
**** Nessus performance is abysmal when running under VMware
**** We do not recommand you use this setup in production




**** This host seems to be running under VMware.
**** Nessus performance is abysmal when running under VMware
**** We do not recommand you use this setup in production


nessusd (Nessus) 3.0.3. for Linux
(C) 1998 - 2006 Tenable Network Security, Inc.

Processing the Nessus plugins...
[##################################################]

All plugins loaded
 - Please run /opt/nessus//sbin/nessus-add-first-user to add an admin user
 - Register your Nessus scanner at http://www.nessus.org/register/ to obtain
   all the newest plugins
 -   You can start nessusd by typing /sbin/service nessusd start
```

create the first Nessus User

```
[root@vpnscan install]# /opt/nessus/sbin/nessus-add-first-user
```

```
**** This host seems to be running under VMware.
**** Nessus performance is abysmal when running under VMware
**** We do not recommand you use this setup in production



**** This host seems to be running under VMware.
**** Nessus performance is abysmal when running under VMware
**** We do not recommand you use this setup in production


Using /var/tmp as a temporary file holder

Add a new nessusd user
----------------------


Login : vpnscan
Authentication (pass/cert) [pass] :
Login password :
Login password (again) :

User rules
----------
nessusd has a rules system which allows you to restrict the hosts
that vpnscan has the right to test. For instance, you may want
him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules syntax

Enter the rules for this user, and hit ctrl-D once you are done :
(the user can have an empty rules set)



Login             : vpnscan
Password          : ***********
DN                :
Rules             :



Is that ok ? (y/n) [y] y
user added.
Thank you. You can now start Nessus by typing :
/opt/nessus//sbin/nessusd -D
```

Activate the nessus account (go to www.nessus.org/register to get the activation code for this installation).

```
To activate your account, simply execute the following command :

        [root@vpnscan install] /opt/nessus/bin/nessus-fetch --register ABCD-12AB-5ABC-12AB-1234

        Now fetching the newest plugin set from plugins.nessus.org...
        Your Nessus installation is now up-to-date.
        If auto_update is set to 'yes' in nessusd.conf, Nessus will
        update the plugins by itself.

        Verify that nessud.conf is set to auto update
        # Automatic plugins updates - if enabled and Nessus is registered, then
        # fetch the newest plugins from plugins.nessus.org automatically
        auto_update = yes
        # Number of hours to wait between two updates
        auto_update_delay = 24
```

Now install the Nessus client, so that this host can initiate scans:

```
[root@vpnscan install]# rpm -iv NessusClient-1.0.0.RC5-es4.i386.rpm
Preparing packages for installation...
NessusClient-1.0.0.RC5-es4
[root@vpnscan install]#


Start the service (this will take a while)
[root@vpnscan nessus]# service nessusd start
Starting Nessus services:                                  [  OK  ]
[root@vpnscan nessus]#
```

To start the Swatch install:

```
# tar -xvzf swatch-3.2.1.tar.gz
# cd swatch-3.2.1/
```

The command sequence for swatch and each of the perl module dependencies for swatch is:

```
perl Makefile.PL
make
make test
make install
make realclean
```

However, after running Makefile.PL, it is found that some prerequisites are missing:

```
[root@vpnscan swatch-3.2.1]# perl Makefile.PL
Checking if your kit is complete...
Looks good
Warning: prerequisite Date::Calc 0 not found.
Warning: prerequisite Date::Format 0 not found.
Warning: prerequisite Date::Manip 0 not found.
Warning: prerequisite File::Tail 0 not found.
Warning: prerequisite Time::HiRes 1.12 not found.
Writing Makefile for swatch
[root@vpnscan swatch-3.2.1]#
```

These modules can all be found at http://search.cpan.org

After trying to install the prerequisites, additional prerequisites are discovered! When this phase of the installation is complete, the following additional packages will be installed:

```
=head2 Wed Sep  6 00:17:41 2006: C<Module> L<TimeDate|TimeDate>
C<VERSION: 1.16>
=head2 Wed Sep  6 00:19:54 2006: C<Module> L<Bit::Vector|Bit::Vector>
C<VERSION: 6.4>
=head2 Wed Sep  6 00:20:31 2006: C<Module> L<Carp::Clan|Carp::Clan>
C<VERSION: 5.3>
=head2 Wed Sep  6 00:22:16 2006: C<Module> L<Date::Calc|Date::Calc>
C<VERSION: 5.4>
=head2 Wed Sep  6 00:24:00 2006: C<Module> L<Date::Manip|Date::Manip>
C<VERSION: 5.44>
=head2 Wed Sep  6 00:24:41 2006: C<Module> L<File::Tail|File::Tail>
C<VERSION: 0.99.3>
=head2 Wed Sep  6 00:28:31 2006: C<Module> L<Time::HiRes|Time::HiRes>
C<VERSION: 1.90>
=head2 Wed Sep  6 00:30:01 2006: C<Module> L<File::Tail|File::Tail>
C<VERSION: 0.99.3>
=head2 Wed Sep  6 00:30:34 2006: C<Module> L<swatch|swatch>
C<VERSION: 3.2.1>
```

(this is a filtered dump of the file /usr/lib/perl5/5.8.5/i386-linux-thread-multi/perllocal.pod - as each module is installed, this file is updated.)

Mount Points:

A mountpoint is required to read the syslog file. It's certainly possible to direct syslog to the vpnscan host, but in this environment a centralized syslog server is already configured (using kiwi syslog on a windows host).

A mountpoint is also required to write the Nessus reports to. In this installation, they will be written directly to the IT group's private intranet. These files will be http browseable as they are posted.

Edit /etc/fstab to contain the mounts required for the installation:

```
//sysloghost/logs  /opt/vpnscan/mnt/syslog smbfs username=syslogview,password=Abc123 0 0
//fileprintserver/vpnscans /opt/vpnscan/mnt/vpnscans smbfs
username=dom\vpnscans,password=Abc123 0 0
```

The Scripts:

The scripts will be built in /opt/vpnscan. This keeps it away from the operating system, in it's own volume. This approach prevents it from filling up the root, and also clearly dillineates "who is where" to make backups and documentation simpler.

Swatch.conf

The file /opt/etc/swatch.conf contains only 2 lines:

```
watchfor /109005/
   exec /opt/vpnscan/bin/swatch_scan_ext.sh $_
```

The trigger event is an occurance of the string “109005” in syslog. Event id 109005 is what a cisco PIX writes to syslog when a user successfully connects to the vpn gateway. The event has the format:

```
2006-09-06 21:35:11 Local4.Info 172.16.1.2 Sep 06 2006 21:35:11: %PIX-6-109005:
Authentication succeeded for user username from 111.222.33.44/0 to 99.88.77.66/0 on
interface outside
```

This event is used to trigger swatch because it provides all the information required for the scan in a single line – the user who is authenticating and their public internet ip (either native on their pc, or their firewall’s ip).

When the event is seen in syslog, swatch executes the command shown. $_ indicates that the entire syslog message should be passed to the command. The swatch documentation indicates that discrete arguments can be passed (which would work better here). This function has worked in previous versions, but is broken in the current release of swatch.

Swatcher.sh

The script file /opt/vpnscan/bin/swatcher.sh actually calls swatch:

```
#!/bin/bash
swatch -c /opt/vpnscan/etc/swatch.conf  -t /opt/vpnscan/mnt/syslog/swatchlog.txt --pid-
fil /var/run/swatcher.pid –daemon
```

Note the filename being monitored (or “swatched”) is swatchlog.txt. It was found that “swatching” a logfile in this environment actually locked the file on the windows host, which in turn broke log rotation (the log would never be moved to archive, and simply grew forever). It was decided to have Kiwi create a second logfile for swatch monitoring, which could then be deleted periodically from the Linux side (more on that later). It was decided to go this route because the syslog host already has the appropriate security and space allocated, and rebuilding this on the Linux host would result in two locations hosting this sensitive information. In addition, this approach

leaves the VPN gateway with a single log trap host, which (theoretically) reduces the CPU load on the VPN gateway by some small amount.

The "—daemon" parameter instructs swatch to run as a daemon – it is not dependant on any terminal session being logged in to run.

Also, note the PID (Process IID) file "/var/run/swatcher.pid". This provides a handy way to reset the daemon from a script (again, more on this later).

Swatchrst.sh, rc.local and crontab

The file /opt/vpnscan/bin/Swatchrst.sh provides an automated method of resetting the swatch process that is now running independent of any console or vty session:

```
if /bin/ps -ef | grep swatch | grep pid; then  kill $(cat /var/run/swatcher.pid ); fi
echo 'a' >/opt/vpnscan/mnt/syslog/swatchlog.txt
/opt/vpnscan/bin/swatcher.sh
```

In short, this script first checks to see if the swatch process is running (checking for the pid string filters out children and the ps parent command itself) – if the process is running, kill it using the contents of the pid file. The echo command shown both deletes the swatchlog.txt file, and creates a new one (swatch will not start if the logfile does not exist). Finally the swatch process itself restarts.

Many of the examples that are listed on the internet use pkill –HUP to restart swatch. In fact, swatch also has a command line argument to restart the process periodically. The decision to using "kill" with no HUP parameter was made for two reasons:

the process actually needs to be stopped, so that the swatchlog file can be deleted before restarting the daemon (remember the file lock discussed?). Using the –HUP (Hangup) parameter restarts the process, but would not address the logfile issue.

By scheduling this hourly, the size of swatchlog.txt is kept under tight control, this also ensures that if the daemon stops for some reason, it will not be stopped for too long:

Adding this to /etc/crontab to restart at 10 minutes past each hour

```
10 * * * * root /opt/vpnscan/bin/swatchrst.sh > /dev/null 2>&1
```

Also, add swatchrst.sh to /etc/rc.d/rc.local so that it will start if this server is reloaded

```
/opt/vpnscan/bin/swatchrst.sh > /dev/null 2>&1
```

In addition, on some Operating Systems there can be a race condition between the execution of /etc/fstab, and rc.local. If this is the case during a given install, adding the line “mount –a” into rc.local, ahead of swatchrst.sh will resolve this.

The ‘> /dev/null 2>&1’ string sends both STDIN and STDERR to /dev/null. In some cases text output from a daemon-type process can hang the process.

swatch_scan_ext.sh

/opt/vpnscan/bin/swatch_scan_ext.sh is the script that actually runs when a user vpn's in. It scans the public ip of the remote user, then tests for policy compliance. If a policy violation is found, an email is sent to the IT group, who follows up with the remote user.

`#!/bin/bash`	This runs in the bash shell, so that the shift command available (amongst other things)

`function getip {`
` echo $AD | sed -f /opt/vpnscan/bin/fixip.sed.in`
`}`
	This function that will "fix" an ip address (see sed file below) that is available both as a variable and as a filename.

`function timetofn {`
`  date | sed -f /opt/vpnscan/bin/fixdate.sed.in`
`}`
	A function to similarly fix the output of the "date" command

`function nocrlf`
`{`
`exec sed -e :a -e '$!N;s/\n/\\\|/;ta' -e 'P;D'`
`</opt/vpnscan/etc/services.deny`
`}`
	This function reads in the services.deny file, then replaces all carriage return (0xA) characters with '|'. This in effect builds a regular expression OR-ing all of the conditions. This is then compared against the entire Nessus scan report to generate an alert trigger. The services.deny file lists all conditions that define a Policy Violation Event.

`#`
`# get all of the variables that drive this script`
`#`
 	
`source ../etc/vpnscan.cf`
	Vpnscan.cf is used to identify all all site and most program specific variables.

`#`
`# get the list of denied services`
`#`
`FAILSTR=$(nocrlf)`
	This reads in the file services.deny (see function nocrlf above). It is echoed simply for troubleshooting.

`#`
`# test echo to verify services.deny file format`
`#`
`echo $FAILSTR`	

`#`
`# test echo for diag testing of swatch.conf`
`#`
`echo "all args $*"`
	Similarly, echo the entire command line argument string for troubleshooting purposes.

`# /bin/sh will only see the first 9 args`
`# /bin/bash will go past 9, using shift`
 	
`# first, kill the first 13 tokens in the syslog statement`
`shift 13`
 	
`# now, arg 1 is the userid`
`#  arg 3 is the public (internet) address`
	"shift" the command lines arguments over so that the command line arguments required are available. Swatch has the functionality to simply pass these values as $1 and $2, however, this feature is broken in the current release.

`AD=$3`	Get the ip address from the command line, as 111.222.333.444/0 (note the subnet mask on the end)
`WHO=$1`	Get the userid
`# Trim the mask off the ip` `IP=$(getip)`	Run the getip function, so that the address now looks like 111.222.33.44 (trims off the subnet mask)
`# get the date with underscores instead of spaces` `NOW=$(timetofn)`	Get the time and date in a format that can be used for a filename, in the format "Fri_Sep__1_04_06_08_EDT_2006"
`# dump the ip into the input file for snort` `echo $IP >$BASEDIR/tmp/scan_$NOW`	Create a temp file, based on timestamp, with the ip to be scanned as its content. This will prevent any collisions or errors if 2 scan run times overlap.
`/opt/nessus/bin/nessus -T html -x -q $NESSUSSERVER 1241 $NESSUSUSR $NESSUSPWD $BASEDIR/tmp/scan_$NOW $REPORTDIR/$WHO\_$IP\_$NOW.html`	Run Nessus output to html to the vpnscans share, with a meaningful name similar to:

	USERID_111.222.33.44_Fri_Sep__1_04_06_08_EDT_2006 This format was useful in this environment, as it provides the userid, the ip and the date/time all in the filename. Rotation of these reports was done using KIWI syslog, based on file date. The input for nessus (ie – what host to scan) is take from the tmp file.
`rm -f $BASEDIR/tmp/scan_$NOW`	remove the tmp file
`if (cat $REPORTDIR/$WHO\_$IP\_$NOW.html | grep "$FAILSTR$")` `then`	Does the scan voliate policy?
`mutt -a $REPORTDIR/$WHO\_$IP\_$NOW.html -s "Userid $WHO failed Security Scan on VPN connect" $ALERTUSR <$BASEDIR/bin/failscan.msg.in`	If so, email the user or group smtp address identified in the vpnscan.cf file). MUTT is used for this send, but any mail client or method will work just as well.
`echo $WHO\_$IP\_$NOW >>/var/log/swatcher.log` `fi`	Finally, add a record of the failure to the local logfile

/opt/vpscan/etc/vpscan.cf

This shell script that is treated as a config file - it is used to set all of the variables used in swatch_scan_ext.sh. This file is commented heavily, so should not require additional documentation. All site-specific information is located in this file, so deploying VPNSCAN at additional sites should not require editing of the main scripts.

```
#
# These Variables are typically ok as-is
#
#This is where the app is installed
BASEDIR=/opt/vpnscan
# The Nessus Server - typically this is localhost
NESSUSSERVER=127.0.0.1
# Credentials for accessing Nessus
NESSUSUSR=vpnscan
NESSUSPWD=Passw0rd123
# Where should we deposit the Nessus Reports
REPORTDIR=/opt/vpnscan/mnt/vpnscans
# Where should we look for our syslog file
SYSLOGDIR=/opt/vpnscan/mnt/syslog
#
# These variables are site-specific and should be tailored
#
#
# The Corporate mail server, spam filter or other valid smtp host
SMTPSRV=172.16.1.22
# Which user or group should receive alerts
ALERTUSR=itservices@metafore.ca
```

/opt/vpnscan/etc/services.deny

This file holds a list of strings that are used as triggers for alerts. In most cases, they are tcp or udp protocols, but note that the last one is a simple ASCII string "Security Hole Found". Similar to the vpnscan.cf file, this decouples the events that trigger alerts from the actual code of the script.

445/tcp	Microsoft-DS
23/tcp	telnet
161/udp	SNMP
5631/tcp	PC/Anywhere
135/tcp	DCE Endpoint Resolution
139/tcp	Netbios Session Service
DCE/RPC	
137/udp	WINS Name Service
138/udp	Netbios Datagram Service
135/udp	DCE Endpoint Resolution
3389/tcp	RDP
5900/tcp	VNC
5901/tcp	VNC
6000/tcp	X-Windows
6001/tcp	X-Windows
6002/tcp	X-Windows
6003/tcp	X-Windows
389/tcp	LDAP
2049/tcp	NFS
2049/udp	NFS
2967/tcp	Norton Antivirus Corporate Client
Security Hole Found	Nessus Key String indicating a serious vulnerability

Ancillary files:

Sed input files: **Fixip.sed.in**

s/\/0//	Delete the trailing "/0" from the ip address that syslog gives us.

fixdate.sed.in

s/ /_/g	Replace all occurences of the space character in the date with underscores
s/:/_/g	Similarly, replace all colons with underscores.

Failscan.msg.in

```
This user failed the security scan that is done on a VPN connection
The remote host has one or more of the services below open to the public
internet:
                tcp/445         Microsoft DS Access
        tcp or udp/135, 137, 139 Microsoft / SMB Netbios / RPC
                tcp/3389        RDP
                tcp/5631        PCAnywhere data
                tcp/2967        Symantec Managed AV Client
                tcp/5900-5901   VNC
                tcp/23          telnet
                udp/161         snmp
                6000-6004       X-windows
                tcp or udp/2049 NFS


Please contact them as soon as possible to ensure that they install a corporate approved
firewall at their remote location, or correctly reconfigure their existing firewall.
```

Related Files: /opt/nessus/etc/nessus/nessusd.conf

The Nessus client was run with it's default config file (below). One key thing that might be considered for change would be to expand the port range. Note also that "safe" checks are used for our scans – this process is meant to be as non-intrusive as possible, so "crashing" a remote firewall would not be desirable. In this install, the default was deemed adequate to identify any station in violation of the Remote Access Policy.

```
# Configuration file of the Nessus Security Scanner


# Every line starting with a '#' is a comment

# Path to the security checks folder :
```

```
plugins_folder = /opt/nessus//lib/nessus/plugins

# Automatic plugins updates - if enabled and Nessus is registered, then
# fetch the newest plugins from plugins.nessus.org automatically
auto_update = yes
# Number of hours to wait between two updates
auto_update_delay = 24

# Maximum number of simultaneous hosts tested :
max_hosts = 40

# Maximum number of simultaneous checks against each host tested :
max_checks = 5

# Niceness. If set to 'yes', nessusd will not renice itself to -5.
be_nice = no

# Throttle scan when CPU is overloaded
throttle_scan = yes

# Log file :
logfile = /opt/nessus//var/nessus/logs/nessusd.messages

# Shall we log every details of the attack ? (disk intensive)
log_whole_attack = no

# Log the name of the plugins that are loaded by the server ?
log_plugins_name_at_load = no

# Dump file for debugging output, use `-' for stdout
dumpfile = /opt/nessus//var/nessus/logs/nessusd.dump

# Rules file :
rules = /opt/nessus//etc/nessus/nessusd.rules

# Users database :
users = /opt/nessus//etc/nessus/nessusd.users

# CGI paths to check for (cgi-bin:/cgi-aws:/ can do)
cgi_path = /cgi-bin:/scripts

# Range of the ports the port scanners will scan :
# 'default' means that Nessus will scan ports found in its
# services file.
port_range = default

# Optimize the test (recommanded) :
optimize_test = yes

# Language of the plugins :
language = english



# Optimization :
# Read timeout for the sockets of the tests :
checks_read_timeout = 5
# Ports against which two plugins should not be run simultaneously :
# non_simult_ports = Services/www, 139, Services/finger
non_simult_ports = 139, 445
# Maximum lifetime of a plugin (in seconds) :
plugins_timeout = 320


# Safe checks rely on banner grabbing :
safe_checks = yes


# Automatically activate the plugins that are depended on
auto_enable_dependencies = yes
```

```
# Do not echo data from plugins which have been automatically enabled
silent_dependencies = yes


# Designate hosts by MAC address, not IP address (useful for DHCP networks)
use_mac_addr = no


#--- Knowledge base saving (can be configured by the client) :
# Save the knowledge base on disk :
save_knowledge_base = no
# Restore the KB for each test :
kb_restore = no
# Only test hosts whose KB we do not have :
only_test_hosts_whose_kb_we_dont_have = no
# Only test hosts whose KB we already have :
only_test_hosts_whose_kb_we_have = no
# KB test replay :
kb_dont_replay_scanners = no
kb_dont_replay_info_gathering = no
kb_dont_replay_attacks = no
kb_dont_replay_denials = no
kb_max_age = 864000
#--- end of the KB section

# Can users upload their plugins ?
plugin_upload = yes
# Suffixes of the plugins the user can upload :
plugin_upload_suffixes = .nasl, .nasl3, .inc, .inc3, .nbin, .audit
# Name of the user who can remotely update the plugins
admin_user = vpnscan


# If this option is set, Nessus will not scan a network incrementally
# (10.0.0.1, then 10.0.0.2, 10.0.0.3 and so on..) but will attempt to
# slice the workload throughout the whole network (ie: it will scan
# 10.0.0.1, then 10.0.0.127, then 10.0.0.2, then 10.0.0.128 and so on...
slice_network_addresses = no

# Should consider all the NASL scripts as being signed ? (unsafe if set to 'yes')
nasl_no_signature_check = no

#end.
#
# Added by nessus-mkcert
#
cert_file=/opt/nessus//com/nessus/CA/servercert.pem
key_file=/opt/nessus//var/nessus/CA/serverkey.pem
ca_file=/opt/nessus//com/nessus/CA/cacert.pem
# If you decide to protect your private key with a password,
# uncomment and change next line
# pem_password=password
# If you want to force the use of a client certificate, uncomment next line
# force_pubkey_auth = yes
```


CHROOT

It was determined that a chroot implementation was “overkill” for this service. This is an internal host, and no users at all access it – it’s purely a service machine.

Immediate Futures (the "I didn't get to it" list)

Restricting access to files containing userids and passwords with chmod, is both a good idea and a best practice. This has not yet been done, in the interests of debugging program functionality first. Since the accounts in questions have severely restricted access, this was deemed an acceptable risk during the development phase. Even still, this should be almost the first update made to VPNSCAN.

Update swatch.conf and the shell scripts should be updated to include support for:

- Cisco VPN 3000
- Cisco IOS router as a VPN gateway (this will require support for multi-line events)
- Cisco PIX version 7.x
- Cisco dial devices
- Shiva dial devices
- Checkpoint Secure/Remote VPN

Implementation of an active response module, probably written in EXPECT, to update the shun list on a cisco PIX based on Policy Violation Events. Any such module should have an timer function, so that shuns expire after a reasonable, configurable time.

Using VPNScan with other Devices

Cisco VPN 3000

A successful VPN connection to a Cisco VPN 3000 will generate a syslog entry similar to the ones below:

This message indicates the initial successful authentication of the vpn group (this is the first log entry that has all the information required to intiate a scan):

```
2006-09-07 01:22:32    Local1.Notice  192.168.192.5  5788 09/07/2006 01:29:34.570 SEV=4 IKE/52
RPT=130 111.222.33.44  Group [kit_ipsec] User (flname) authenticated.
```

This message indicates the successful authentication of the vpn user (this message indicates that the user's vpn connection is completely established):

```
2006-09-07 01:22:34   Local1.Notice  192.168.192.5  5806 09/07/2006 01:29:36.410 SEV=4 IKE/49
RPT=144 111.222.33.44  Group [kit_ipsec] User [flname] Security negotiation complete for User
(flname) Responder, Inbound SPI = 0x0fb6e799, Outbound SPI = 0xb032a131
```

So for a VPN 3000 device, triggers could be event 5788 or event 5806 (depending on preference). Some minor modifications to the shell script would be required.

Cisco IOS Router

A successful VPN connection to a Cisco IOS router configured as VPN gateway (ie has a dynamic crypto map, with authentication to an internal RADIUS host) will generate a syslog entry similar to the ones below. However, in order to generate these messages, "debug radius" needs to be enabled

```
2006-09-07 13:42:04   Local7.Debug   192.168.21.254 1085541: 10w3d: RADIUS:  User-Name
[1]   15  "flname"
2006-09-07 13:42:04   Local7.Debug   192.168.21.254 1085542: 10w3d: RADIUS:  User-Password
[2]   18  *
2006-09-07 13:42:04   Local7.Debug   192.168.21.254 1085543: 10w3d: RADIUS:  Calling-Station-Id
[31]  16  "111.222.33.44"
```

Note that there is no single line entry with all of the information required to initiate a scan. What is required for this case is to generate a SWATCH configuration that will catch events that span 3 consecutive syslog lines. This might be accomplished with something similar to (note that the exact syntax for the keep_open switch approach has not been tested) :

```
watchfor /1085541/
   exec echo $_ >/somepath/sometempfile.txt
watchfor /1085543/
   exec echo $_ >>/somepath/sometempfile.txt
   exec /opt/vpnscan/bin/someshellscript.sh </somepath/sometempfile.txt

someshellscipt.sh would then call our existing swatch_scan_ext.sh script
```

alternatively, something similar to this might be deployed:

```
watchfor /1085541/
   pipe  'echo $_ | /opt/vpnscan/bin/someshellscript',keep_open
watchfor /1085543/
   pipe 'echo $_ | //opt/vpnscan/bin/someshellscript'
```

This swatch config will provide a 2 line input to the swatch_scan_ext.sh script. This script is not part of this paper, but will be addressed before VPNSCAN can go live at some of the target sites.

The discussions around the use of the shell script with other devices has highlighted that a method of handling multiple line events is required – the approach above should work nicely, as long as two users do not vpn in within a second of each other. More importantly, entirely de-coupling the parsing of event messages would permit us to support multiple device types with a single swatch.conf file and a single swatch_scan_ext.sh file, with a simple shell script for each device type to parse out the required parameters and call swatch_scan_ext. This will be simplified further when the parsing function is fixed in swatch (hopefully in the next release) – this would mean that parsing parameters out would only be required for multiple-line events.

Possible Futures ("the road not taken")

VPNSCAN could be expanded to scan unprotected ip's – for instance:

- Monitoring dhcp logs and scan machines as they connect to the corporate network
- Monitoring syslog and scan machines as they connect to Wireless Access Points.
- Monitoring syslog to scan vpn users' private ip's (ie – their internally routable ip's, not their internet ip's) after they successfully connect to the vpn gateway. This would effectively scan users' home PCs, even if they never connect that system to the corporate network. This approach may have some legal implications, as the hosts being scanned and the data on them may not be corporate property.

However, these approaches are not taken in this first deployment and are not planned for the immediate future, for several reasons. In the applications described above, scanning would not ensure compliance with any written policy, so the reason for doing it at all starts to become tenuous.

Secondly, the volume of data would be unreasonably large, any deployment along these lines would need a much better method of organizing data. Most likely a mysql database would be required, with a web front-end.

Finally, the triggers for alerting would be much harder to arrive at – the default services.deny file used by VPNSCAN would trigger on *every* scan for these expanded applications.

These applications are exactly what the competing NAC and NAP frameworks are meant to deal with, and while there is widespread interest in both products, the up-front costs involved in deployment have limited actual installations.

Appendix – Customer NESSUS Scans

Customer NESSUS Scan – Properly Configured Firewall

This scan does not generate an alert. As can be seen, Nessus finds that the dns name for the ip address can be resolved, and a traceroute is successful. No other items are flagged.

==========

Nessus Scan Report
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

Scan Details	
Hosts which were alive and responding during test	1
Number of security holes found	0
Number of security warnings found	0

Host List	
Host(s)	**Possible Issue**
44.55.66.77	Security note(s) found

Analysis of Host		
Address of Host	**Port/Service**	**Issue regarding Port**
44.55.66.77	general/udp	Security notes found
44.55.66.77	general/tcp	Security notes found

Security Issues and Fixes: 111.222.33.44		
Type	**Port**	**Issue and Fix**
Informational	general/udp	For your information, here is the traceroute from 172.16.2.128 to 44.55.66.77: 172.16.2.128 172.16.0.253

		209.183.146.106 209.183.143.105 64.39.160.97 209.183.132.129 198.32.245.29 64.71.240.58 66.185.80.42 66.185.81.62 66.185.81.85 64.71.241.70 66.185.90.251 44.55.66.77 Nessus ID : 10287
Informational	general/tcp	44.55.66.77resolves as CPE0005dd0c1923-CM000f9faa97bc.cpe.net.cable.rogers.com. Nessus ID : 12053
Informational	general/tcp	Information about this scan : Nessus version : 3.0.3 Plugin feed version : 200609042215 Type of plugin feed : Registered (7 days delay) Scanner IP : 172.16.2.128 Port scanner(s) : nessus_tcp_scanner Port range : default Thorough tests : no Experimental tests : no Paranoia level : 0 Report Verbosity : 1 Safe checks : yes Max hosts : 20 Max checks : 4 Scan Start Date : 2006/9/5 10:02 Scan duration : 68 sec Nessus ID : 19506

This file was generated by Nessus, the open-sourced security scanner.

Customer NESSUS Scan – Improperly Configured Firewall (LINUX based)

This scan generates an alert. Nessus finds that that SNMP is open to the internet, which generates an alert. Also, the phrase “security hole” is found, which generates an alert. The DNS entries are not typical for a home firewall, but are not flagged in this implementation as an alert. From the SNMP data returned, it appears that this is a Watchguard Wireless firewall (WG4500). From the data returned, it is most likely that:
a/ this unit is either severely mis-configured, or
b/ perhaps is hooked up backwards (with a “trusted” switch port connected to the internet, instead of the “untrusted” wan port). However, if this were the case any Microsoft ports on the PC would be seen.
It turned out after investigation that this unit *was* hooked up backwards, but the client PC had a software firewall installed.

==========

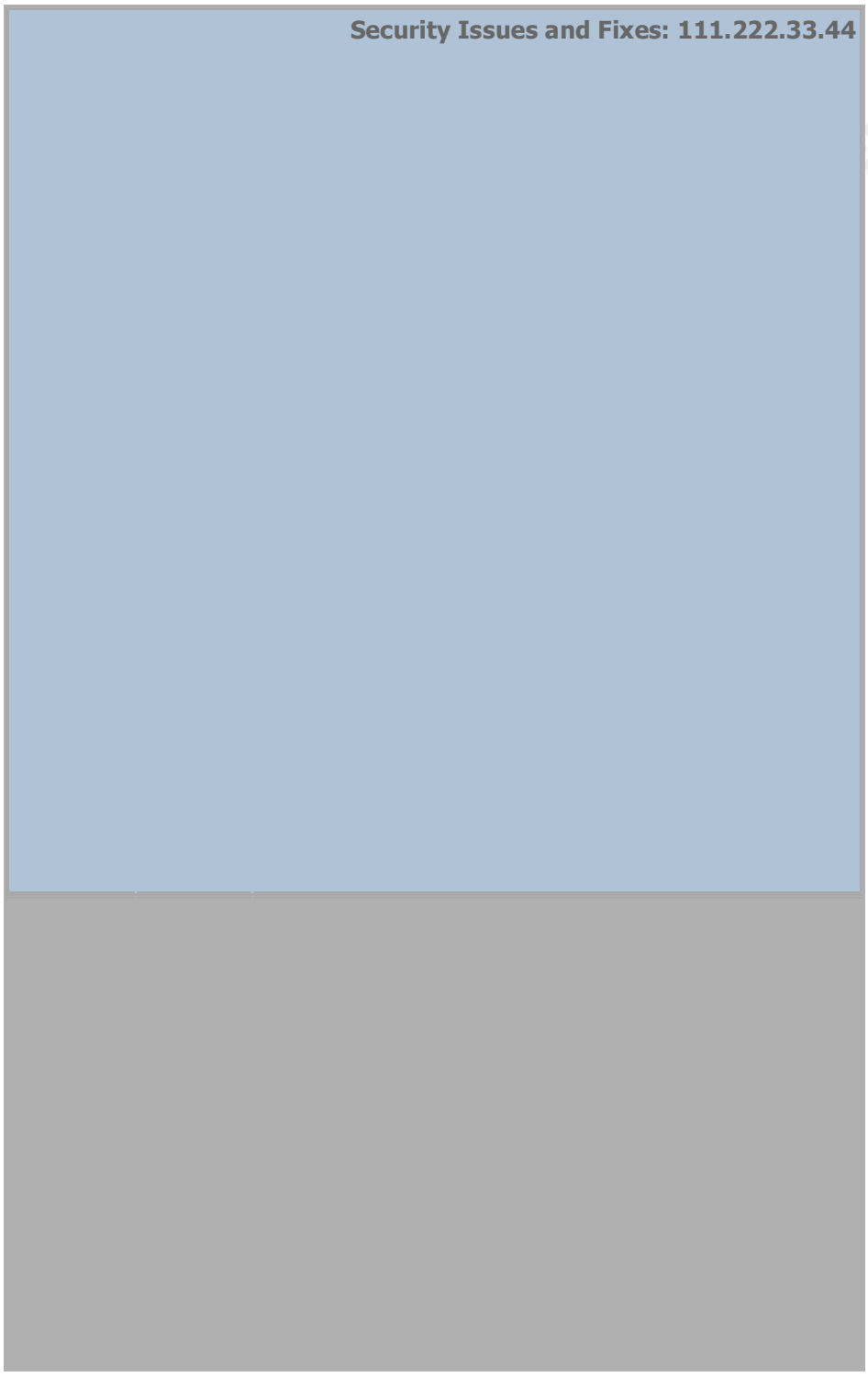
Nessus Scan Report
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

Scan Details	
Hosts which were alive and responding during test	1
Number of security holes found	1
Number of security warnings found	1

Host List	
Host(s)	**Possible Issue**
111.222.33.44	Security hole(s) found

Analysis of Host

111.222.33.44	general/tcp	Security notes found
111.222.33.44	domain (53/tcp)	Security notes found
111.222.33.44	general/udp	Security notes found
111.222.33.44	snmp (161/udp)	Security hole found



Informational	domain (53/udp)	Synopsis : Remote DNS server is vulnerable to Cache Snooping attacks. Description : The remote DNS server answers to queries for third party domains which do not have the recursion bit set. This may allow a remote attacker to determine which domains have recently been resolved via this name server, and therefore which hosts have been recently visited. For instance, if an attacker was interested in whether your company utilizes the online services of a particular financial institution, they would be able to use this attack to build a statistical model regarding company usage of aforementioned financial institution. Of course, the attack can also be used to find B2B partners, web-surfing patterns, external mail servers, and more... For a much more detailed discussion of the potential risks of allowing DNS cache information to be queried anonymously, please see: http://community.sidestep.pt/~luis/DNS-Cache-Snooping/DNS_Cache_Snooping_1.1.pdf Risk factor : Low / CVSS Base Score : 2 (AV:R/AC:L/Au:NR/C:P/A:N/I:N/B:N) Nessus ID : 12217
Informational	domain (53/udp)	It was not possible to fingerprint the remote DNS server. If you know the type and version of the remote DNS server, please send the following signature to dns-signatures@nessus.org : 0:2:2:0:0:0:t:t:t:0:2:0:0:0:0:0:0:0:0:0:0:2:0XD: Nessus ID : 11951
Informational	domain (53/udp)	A DNS server is running on this port. If you do not use it, disable it. Risk factor : Low Nessus ID : 11002
Informational	general/tcp	111.222.33.44 resolves as S01060004e2f7672a.vc.shawcable.net. Nessus ID : 12053
Informational	general/tcp	Information about this scan : Nessus version : 3.0.3 Plugin feed version : 200609042215 Type of plugin feed : Registered (7 days delay) Scanner IP : 172.16.2.128 Port scanner(s) : nessus_tcp_scanner Port range : default Thorough tests : no Experimental tests : no Paranoia level : 0 Report Verbosity : 1 Safe checks : yes

		Max hosts : 20 Max checks : 4 Scan Start Date : 2006/9/5 15:48 Scan duration : 53 sec Nessus ID : 19506
Informational	domain (53/tcp)	Synopsis : It is possible to obtain the version number of the remote DNS server. Description : The remote host is running BIND, an open-source DNS server. It is possible to extract the version number of the remote installation by sending a special DNS request for the text 'version.bind' in the domain 'chaos'. Solution : It is possible to hide the version number of bind by using the 'version' directive in the 'options' section in named.conf Risk factor : None Plugin output: The version of the remote BIND server is : PowerDNS Recursor 3.1.2 $Id: pdns_recursor.cc 858 2006-06-21 20:30:35Z ahu $ Nessus ID : 10028
Informational	domain (53/tcp)	A DNS server is running on this port but it only answers to UDP requests. This means that TCP requests are blocked by a firewall. This configuration is not RFC-compliant. Contrary to common belief, TCP transport is not restricted to zone transfers (AXFR) : - answers bigger than 512 bytes are always transmitted over TCP. - for all other requests, UDP is only 'preferred' for performance reasons. i.e. RFC1035 (STD0013) does not forbid a DNS client from issuing its queries directly over TCP. ** If you are sure that your DNS server will never return ** answers bigger than 512 bytes and that the client ** software prefers UDP (which is nearly certain), you may ** disregard this message. Read RFC1035 (STD0013) for more information. Risk factor : None Nessus ID : 18356
Informational	general/udp	For your information, here is the traceroute from 172.16.2.128 to 111.222.33.44: 172.16.2.128 172.16.0.253 209.183.146.106 209.183.143.105

		64.39.160.126 209.183.132.129 198.32.245.12 66.163.66.13 66.163.76.74 66.163.76.78 66.163.76.161 66.163.69.69 64.59.158.170 111.222.33.44 Nessus ID : 10287
Vulnerability	snmp (161/udp)	Synopsis : The community name of the remote SNMP server can be guessed. Description : It is possible to obtain the default community names of the remote SNMP server. An attacker may use this information to gain more knowledge about the remote host, or to change the configuration of the remote system (if the default community allow such modifications). Solution : Disable the SNMP service on the remote host if you do not use it, filter incoming UDP packets going to this port, or change the default community string. Risk factor : High Plugin output : The remote SNMP server replies to the following default community strings : private public CVE : CVE-1999-0517, CVE-1999-0186, CVE-1999-0254, CVE-1999-0516 BID : 11237, 10576, 177, 2112, 6825, 7081, 7212, 7317, 9681, 986 Other references : IAVA:2001-B-0001 Nessus ID : 10264
Informational	snmp (161/udp)	Synopsis : The System Information of the remote host can be obtained via SNMP. Description : It is possible to obtain the system information about the remote host by sending SNMP requests with the OID 1.3.6.1.2.1.1.1. An attacker may use this information to gain more knowledge about the target host. Solution :

		Disable the SNMP service on the remote host if you do not use it, or filter incoming UDP packets going to this port. Risk factor : Low Plugin output : System information : sysDescr : Wireless Router sysObjectID : 1.3.6.1.3.9999 sysUptime : 2d 4h 54m 52s sysContact : support@vendor sysName : WG4005E-17 sysLocation : Country sysServices : 79 Nessus ID : 10800
Informational	snmp (161/udp)	Synopsis : The list of network interfaces cards of the remote host can be obtained via SNMP. Description : It is possible to obtain the list of the network interfaces installed on the remote host by sending SNMP requests with the OID 1.3.6.1.2.1.2.1.0 An attacker may use this information to gain more knowledge about the target host. Solution : Disable the SNMP service on the remote host if you do not use it, or filter incoming UDP packets going to this port. Risk factor : Low Plugin output : Interface 1 information : ifIndex : 1 ifDescr : LOCAL_LOOPBACK ifPhysAddress : 000000000000 Interface 2 information : ifIndex : 2 ifDescr : LAN ifPhysAddress : 0004e2f743bc Interface 3 information : ifIndex : 3 ifDescr : WAN ifPhysAddress : 0004e2f7672a Interface 4 information : ifIndex : 4 ifDescr : WLAN_g

		ifPhysAddress : 0004e2f743be Interface 5 information : ifIndex : 7 ifDescr : WDS-1 ifPhysAddress : 0004e2f743bc Interface 6 information : ifIndex : 8 ifDescr : WDS-2 ifPhysAddress : 0004e2f743bc Interface 7 information : ifIndex : 9 ifDescr : WDS-3 ifPhysAddress : 0004e2f743bc Interface 8 information : ifIndex : 10 ifDescr : WDS-4 ifPhysAddress : 0004e2f743bc Nessus ID : 10551

This file was generated by Nessus, the open-sourced security scanner.

Customer NESSUS Scan – No Firewall

This scan generates an alert as well. Nessus finds that several Microsoft ports are opened to the internet. Also, this client has a PC/Anywhere server set up on their workstation (probably so IT staff could remote control his PC the previous day).

==========

Nessus Scan Report
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

Scan Details	
Hosts which were alive and responding during test	1
Number of security holes found	0
Number of security warnings found	0

Host List	
Host(s)	**Possible Issue**
222.33.44.55	Security note(s) found

Analysis of Host		
Address of Host	**Port/Service**	**Issue regarding Port**
222.33.44.55	epmap (135/tcp)	Security notes found
222.33.44.55	microsoft-ds (445/tcp)	Security notes found
222.33.44.55	general/tcp	Security notes found
222.33.44.55	ntp (123/udp)	Security notes found
222.33.44.55	general/udp	Security notes found
222.33.44.55	pcanywheredata (5631/tcp)	Security notes found
222.33.44.55	ssc-agent (2967/tcp)	No Information
222.33.44.55	microcom-sbp (1680/tcp)	No Information
222.33.44.55	vfo (1056/tcp)	Security notes found
222.33.44.55	ansyslmd (1055/tcp)	Security notes found
222.33.44.55	blackjack (1025/tcp)	Security notes found

Security Issues and Fixes: 222.33.44.55		
Type	**Port**	**Issue and Fix**
Informational	epmap (135/tcp)	Synopsis : A DCE/RPC service is running on the remote host. Description : By sending a Lookup request to the port 135 it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe. Risk factor : None Plugin output : The following DCERPC services are available locally : Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 4b112204-0e19-11d3-b42b-0000f81feb9f, version 1.0 Description : SSDP service Windows process : unknow Type : Local RPC service Named pipe : LRPC00000110.00000001 Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : OLEAB0BF6DB15414981AA7851517A68 Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : wzcsvc Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : IcaApi Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : Wireless Link Notification Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5, version 1.0 Description : DHCP Client Service Windows process : svchost.exe

		Annotation : DHCP Client LRPC Endpoint Type : Local RPC service Named pipe : dhcpcsvc Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : dhcpcsvc Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : Infrared Transfer Send Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : Wireless Link Notification Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : IcaApi Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : wzcsvc Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : OLEAB0BF6DB15414981AA7851517A68 Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : dhcpcsvc Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : Infrared Transfer Send Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : Wireless Link Notification

		Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : IcaApi Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : wzcsvc Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : OLEAB0BF6DB15414981AA7851517A68 Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : dhcpcsvc Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Local RPC service Named pipe : Infrared Transfer Send Nessus ID : 10736
Informational	microsoft-ds (445/tcp)	Synopsis : A DCE/RPC service is running on the remote host. Description : By sending a Lookup request to the port 135 it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe. Risk factor : None Plugin output : The following DCERPC services are available remotely : Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 4b112204-0e19-11d3-b42b-0000f81feb9f, version 1.0 Description : SSDP service Windows process : unknow

		Type : Remote RPC service Named pipe : \PIPE\winreg Netbios name : \\L-WA-STATIONNAME Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 4b112204-0e19-11d3-b42b-0000f81feb9f, version 1.0 Description : SSDP service Windows process : unknow Type : Remote RPC service Named pipe : \PIPE\DAV RPC SERVICE Netbios name : \\L-WA-STATIONNAME Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service Named pipe : \PIPE\atsvc Netbios name : \\L-WA-STATIONNAME Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service Named pipe : \pipe\Ctx_WinStation_API_service Netbios name : \\L-WA-STATIONNAME Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service Named pipe : \pipe\Ctx_WinStation_API_service Netbios name : \\L-WA-STATIONNAME Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service Named pipe : \PIPE\atsvc Netbios name : \\L-WA-STATIONNAME Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service Named pipe : \pipe\Ctx_WinStation_API_service Netbios name : \\L-WA-STATIONNAME Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service Named pipe : \PIPE\atsvc Netbios name : \\L-WA-STATIONNAME Nessus ID : 10736
Informational	general/tcp	222.33.44.55 resolves as bas1-guelph22-1177636122.dsl.bell.ca. Nessus ID : 12053
Informational	general/tcp	Information about this scan :

		Nessus version : 3.0.3 Plugin feed version : 200609042215 Type of plugin feed : Registered (7 days delay) Scanner IP : 172.16.2.128 Port scanner(s) : nessus_tcp_scanner Port range : default Thorough tests : no Experimental tests : no Paranoia level : 0 Report Verbosity : 1 Safe checks : yes Max hosts : 20 Max checks : 4 Scan Start Date : 2006/9/5 15:17 Scan duration : 74 sec Nessus ID : 19506
Informational	general/tcp	The remote host is running Microsoft Windows XP Nessus ID : 11936
Informational	ntp (123/udp)	An NTP (Network Time Protocol) server is listening on this port. Risk factor : Low Nessus ID : 10884
Informational	general/udp	For your information, here is the traceroute from 172.16.2.128 to 222.33.44.55 : 172.16.2.128 172.16.0.253 209.183.146.106 209.183.143.105 64.39.160.126 209.183.132.129 64.187.25.233 66.59.191.121 66.59.191.18 64.230.218.241 64.230.229.9 64.230.242.105 64.230.163.90 64.230.163.102 64.230.207.68 222.33.44.55 Nessus ID : 10287
Informational	pcanywheredata (5631/tcp)	An unknown server is running on this port. If you know what it is, please send this banner to the Nessus team: 0x00: 00 58 08 00 7D 08 0D 0A 00 2E 08 50 6C 65 61 73 .X..}.....Pleas 0x10: 65 20 70 72 65 73 73 20 3C 45 6E 74 65 72 3E 2E e press <Enter>. 0x20: 2E 2E 0D 0A .... Nessus ID : 11154
Informational	vfo (1056/tcp)	This port was detected as being open by a port scanner but is now closed. This service might have been crashed by a port scanner or by a plugin Nessus ID : 10919
Informational	ansyslmd (1055/tcp)	This port was detected as being open by a port scanner but is now closed.

		This service might have been crashed by a port scanner or by a plugin Nessus ID : 10919
Informational	blackjack (1025/tcp)	Synopsis : A DCE/RPC service is running on the remote host. Description : By sending a Lookup request to the port 135 it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe. Risk factor : None Plugin output : The following DCERPC services are available on TCP port 1025 : Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service TCP Port : 1025 IP : 222.33.44.55 Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 1ff70682-0a51-30e8-076d-740be8cee98b, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service TCP Port : 1025 IP : 222.33.44.55 Object UUID : 00000000-0000-0000-0000-000000000000 UUID : 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1.0 Description : Scheduler Service Windows process : svchost.exe Type : Remote RPC service TCP Port : 1025 IP : 222.33.44.55 Nessus ID : 10736

This file was generated by Nessus, the open-sourced security scanner.

Appendix – Example Remote Access Policies

Example Policy 1

This is an excerpt from "The United States House of Representatives Information Security Guidelines to Protect Member and Committee Office Systems from Unauthorized Use (HISPOL 002.1)"

http://www.house.gov/cao-opp/PDFSolicitations/HISPOL_2-1.pdf#search=%22%22security%20policy%22%20home%20firewall%20scan%22

Virtual Private Network (VPN) Users:

The House provides a Virtual Private Network (VPN) service for single-person District Offices, telecommuters, and House staff to access the House Network via House-owned PCs and laptops using their high-speed connections, SecurID cards and the Internet. Secure use of this service requires the following three security measures:

- Current antivirus software must be installed on the system and operational at all times,
- A secure authentication device (SecurID card) must be used to access the House network,
- A personal firewall supported by the House VPN solution must be installed on the system and operational at the time of each connection to the House network.

Example Policy 2

This is not meant as a complete Remote Access Policy for all environments or situations. This is a current policy in use by a customer, used with permission. It is likely that clauses from this policy were sourced from any one of several websites.

ACME Corporation Remote Access Policy

Remote access to ACME Corporate IT resources includes all dial-up, VPN (Virtual Private Network), and secured web access to ACME resources.

SUPPORT LEVELS

All supported workstations will be set up and configured by ACME IT Personnel. All supported workstations will be owned by ACME. Workstations include desktop, server, laptop, palmtop and mobile phone workstations.

Best-efforts support will be provided to home computing resources owned by ACME personnel. However, software approved by the ACME IT Division must be used in all cases (for instance, non-standard operating systems, vpn and/or firewall software). All self-installed workstations of this type will supported on a best-efforts basis only. Best-efforts means that access and/or support can be revoked at any time.

WAIVERS TO POLICY

Waivers to this policy are to be discouraged, except in a limited number of emergency situations.

Formal written approval for waivers is required, in advance, from the CIO, Corporate Services, IT, as well as the senior executive management of the affected department(s).

REMOTE ACCESS CONTROLS

All remote access to ACME networks and computer systems, including intelligent workstations, must be protected by a supplemental security layer that will authenticate users and restrict them to only those networks and computer systems for which they have been properly authorized.

Remote Access controls must be implemented only through ACME approved combinations of hardware and software security tools that meet the following requirements:

- unique identification or access code (user ID) for each user,
- capability to restrict access to specific nodes or network applications,
- access control software/hardware that protects stored data and the security system from tampering,
- audit trails of successful and unsuccessful login attempts,
- automatic system reboot or session cleanup following the disconnection of incoming sessions,
- capability to limit the number of unsuccessful login access attempts, and
- verification of user ID by the use of a secret password, linked to but separate from the operating system user ID/password assigned to the user. Security tokens or software challenge/response methods that generate dynamic passwords are the preferred methods for authenticating dialup access users for systems connected to the ACME network.
- All access from the Internet to ACME Internal System resources (non-dmz) must use encryption services and an approved firewall solution to protect the confidentiality of the session. ACME approved remote access products must be used to assure interoperability for remote access server encryption technologies.

User Responsiblities for use of Remote Access Facilities:

- Information regarding access to company computer and communication systems, such as dial-up modem phone numbers, userids or passwords, are considered confidential. This information must not be posted on electronic bulletin boards, listed in telephone directories, placed on business cards, emailed, or made available to third parties without the written permission of the CIO.
- Care should be taken when using mobile computing facilities in public places, meeting rooms and other unprotected areas outside of the organization's premises. It is important that when in public places care is taken to avoid the risk of overlooking by unauthorized persons.
- Mobile computing facilities should also be physically protected against theft especially when left, for example, in cars and other forms of transport, hotel rooms, conference centres and meeting places. Equipment carrying important, sensitive and/or critical business information should not be left unattended and, where possible, should be physically locked away, or special locks should be used to secure the equipment.
- *Use of ACME computing resources by family members and/or non-ACME personnel must be approved by the CIO.*

SECURITY POLICY AGREEMENT

In order to receive remote access privileges, this document must be read, agreed to and signed by the user who requires access, their department manager, and a representative of Corporate Services, IT. All exceptions to the standard supported image (see Support Levels) must be outlined in the exceptions section of this document.

Violation of this remote access policy is grounds for removal of all remote access privileges.

I have read and understand the policy attached.

Signed	Date
(Please Print)	

Business Unit Leader

Signed	date

IT Representative

Signed	date

Connection Description:

Asset Tag: Make, Model, Serial #	
Access Method (Dial / VPN)	
Internet Particulars (Fixed IP?, DSL / Cable / Look?, Remote Office?, etc)	

Exceptions (All exceptions must be approved as noted in the attached document):

System Ownership:	
Operating System:	
VPN Client:	
Personal Firewall:	
Other: (attach extra sheets if required)	

References

The United States House of Representatives Committee on House Administration (2002*). The United States House of Representatives Information Security Guidelines to Protect Member and Committee Office Systems from Unauthorized Use (HISPOL 002.1)*, Page 23. Retrieved 11 Sept, 2006 from the World Wide Web: http://www.house.gov/cao-opp/PDFSolicitations/HISPOL_2-1.pdf

The example Remote Access Policy #2 is sourced from a customer, with permission.

Components Used:

CENTOS	http://www.centos.org
Swatch	http://sourceforge.net/projects/swatch/
Nessus	http://www.nessus.org
Cisco Documentation	http://www.cisco.com/univercd
VMWare	http://www.vmware.com
KIWI Syslog	http://www.kiwisyslog.com/syslog-info.ph

Components Referenced for Future Development:

Expect	http://expect.nist.gov/
CPAN Net::SSH Perl Module	http://search.cpan.org/~ivan/Net-SSH-0.08/SSH.pm
CPAN Net:Telnet Perl Module	http://search.cpan.org/~jrogers/Net-Telnet-3.03/lib/Net/Telnet.pm