

Implementation and Planning Guide

Shavlik NetChk[®] Protect



Copyright

Copyright © 2010 Shavlik Technologies, LLC. All rights reserved.

No part of this document may be reproduced or retransmitted in any form or by any means electronic, mechanical, or otherwise, including photocopying and recording for any purpose other than the purchaser's personal use without written permission of Shavlik Technologies.

Trademarks

Shavlik NetChk Protect, Shavlik NetChk Limited, Shavlik NetChk Configure, and Shavlik NetChk Deployment Tracker are registered trademarks of Shavlik Technologies. The Shavlik Technologies logo is a trademark of Shavlik Technologies. VMware is a registered trademark of VMware, Inc. Microsoft, Windows, and Microsoft Baseline Security Analyzer are registered trademarks of Microsoft Corporation.

All other trademarks, tradenames, or images mentioned herein belong to their respective owners.

Document Information and Print History

Document number: N/A

Date	Version	Description
February 2010	Shavlik NetChk Protect 7.2	Initial release of the Shavlik NetChk Protect Implementation and Planning Guide .
April 2010	Shavlik NetChk Protect 7.5	Update requirements, add skype to the firewall rules list.

Table Of Contents

CONSOLE INSTALLATION AND HARDWARE RECOMMENDATIONS	1
Recommend OS	1
Prerequisites	1
Recommended Console Machine Resources	1
DISTRIBUTED ENVIRONMENT MANAGEMENT	2
Agentless Environments	2
Distributed Consoles	2
Distribution Servers	3
Disconnected Environments	3
Agents	3
DMZs	3
AGENTLESS PATCH MANAGEMENT	4
Example	4
Time to Implement	4
Port Requirements	4
AGENT PATCH MANAGEMENT	5
Laptops users	5
Secure Environments	5
Low Bandwidth Connections	6
AGENT ROLLOUT OPTIONS	7
Push Install from Console	7
Manual Installation	7
Scripted Installation	7
GPO	8
Custom Patch	8
Other	8
PORT REQUIREMENTS AND FIREWALL CONFIGURATION	9
SQL DATABASE MAINTENANCE	11
Data Retention	11
Reporting	11
DB Backups	11
DB Maintenance Schedule:	12
Full SQL Maintenance Guidance	12
PURGE OLD DATA USING A POWERSHELL SCRIPT	13
Prerequisites	13
Steps to Setup	13

This page intentionally left blank.

The document is designed for duplex printing.

CONSOLE INSTALLATION AND HARDWARE RECOMMENDATIONS

Recommend OS

- Windows Server 2003 Family, SP2 or later
- Windows Server 2008 Family, excluding Server Core
- Windows Server 2008 Family R2, excluding Server Core

Prerequisites

- MSXML 6.0 SP2 Hotfix (only if using Vista SP1 or earlier)
- Windows Installer 4.5 or later (only if installing SQL Express 2008 during NetChk Protect installation)
- Use of SQL Server 2005 or 2008, full or express edition
- SQL Native Client or SQL 2008 Native Client
- Microsoft .NET Framework 3.5, SP1 or later

Average Console Install: 30-60 mins depending on prerequisites and Internet speed

Recommended Console Machine Resources

1 – 250 seats

- **Processor:** 2 Proc Cores 2 GHZ or faster
- **Memory:** Minimum 2 GB RAM (recommended 3+ GB RAM if SQL is local)
- **Database:** SQL 2005 or 2008 Express or Full SQL 2005 or 2008

251-1000 seats

- **Processor:** 2 Proc Cores 2 GHZ or faster (4 Proc Cores+ recommended)
- **Memory:** Minimum 2 GB RAM (recommended 3+ GB RAM if SQL is local)
- **Database:** Recommended SQL 2005 or SQL 2008

1001+ seats

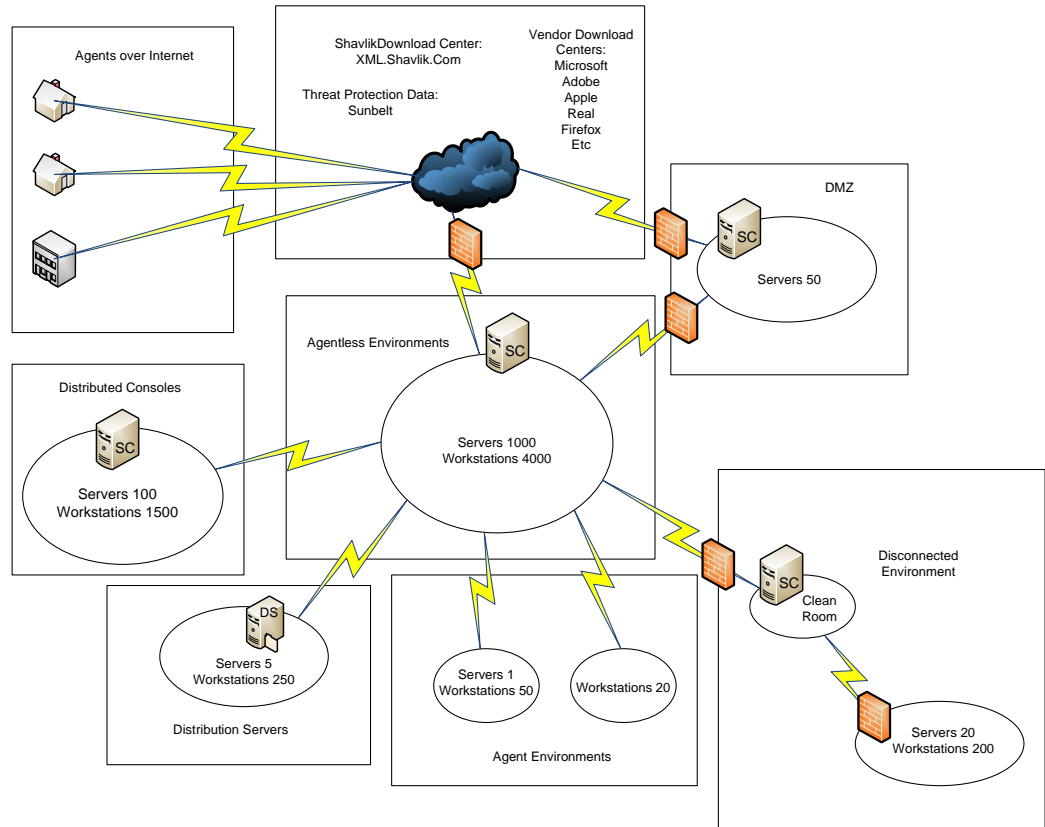
- **Processor:** 4+ Proc Cores 2 GHZ or Faster
- **Memory:** Recommended 4 GB RAM
- **Database:** SQL 2005 or 2008

Agentless High Performance Console

- **Processor:** 8 Proc Cores 2 GHZ or Faster
- **Memory:** 8 GB RAM
- **Database:** Full SQL 2005 or later

DISTRIBUTED ENVIRONMENT MANAGEMENT

NetChk Protect provides a number of features to allow for management of an environment. In the following diagram you will see a variety of different scenarios that we will discuss features for supporting.



Agentless Environments

NetChk Protect can be rapidly configured to support large numbers of machines Agentlessly. By creating a Machine Group from Hostnames, Domain, IP Ranges or Addresses, or using OUs you can identify machines you want to scan and specify administrative credentials for them.

Distributed Consoles

Multiple NetChk Protect Consoles can be configured to distribute workload across large numbers of machines and across WANs to distribute management of the Enterprise. Data can then be rolled up to a central point for reporting purposes.

Distribution Servers

To reduce WAN traffic a Distribution Server can be setup at a remote site to distribute data, engines, patches, and SPs to a remote site reducing data transfer.

Disconnected Environments

NetChk Protect can be configured to pull data, engines, patches, and SPs from an internal source. For partially disconnected environments you can use these features to download from an internet connected machines. Move the files to a Clean Room and from there into the Disconnected Environment. In a fully disconnected environment the connected console would download everything and through a sneaker net process the files can easily be moved from one console to another.

Agents

NetChk Protect also includes Agents. With the Agent admins are able to work around many obstacles and reach machines where Agentless scans may not gain total coverage. Low bandwidth WAN links, Laptops that typically leave the environment, and secure environments such as DMZs are cases where an Agent may be ideal.

DMZs

NetChk Protect can be configured in a number of ways to support DMZs. The option to use Agentless scans by IP address or Ranges allows admins to work around name resolution if it is not available in the DMZ. A distributed console can be configured in the DMZ if IP rules on the firewall are not acceptable to support agentless scanning from the Network into the DMZ.

AGENTLESS PATCH MANAGEMENT

The fastest way to get NetChk Protect configured and patching machines is to do so Agentlessly. Once the console is installed it can take as little as a few minutes to setup machine groups to scan an environment. The time to configure will be based on the complexity of the environment and range of maintenance windows.

Example

An office with three floors and around 500 end users per floor as well as a data center with 200 servers could be broken out many ways. We can create a Machine group per floor based on IP Range for the workstations. Servers we will assume are to be broken down into test, development, and production. Production may have a couple of groups like Domain Controllers and Exchange or SQL broken into separate groups to allow flexibility in scheduling jobs.

Time to Implement

5 mins to create three groups covering the 3 floors of workstations. 10-30 minutes to create around 3-6 server groups by OU or by browsing and selecting machines or import from a file. Another 15 minutes to schedule jobs to scan or scan and auto deploy patches. Overall time for this environment to configure groups and schedule scans up to 45 minutes and we are configured and ready to go.

Port Requirements

For Agentless scans you will need to be able to resolve the machine by the method you created the machine group and be able to access port 139 or 445 on the target machine. File and Print sharing and Remote Registry must be enabled to do the scan. For added security firewall rules between vLANs or on the local machine firewall can be applied to limit traffic to only allow our console's IP to access the port. Depending on the environment, complexity or rules, and change control requirements the amount of time this will add to the initial configuration may vary.

AGENT PATCH MANAGEMENT

Rolling out agents takes a little more prep work to configure the environment to get the first scans coming in from machines. In the long run an Agent can be a great benefit to automate management of some hard to reach machines so it can be worth the effort to rollout to machines.

To configure and install an agent Admins would create agent policies to manage different groups of machines. This may be based on geographic location, or role, or a combination. Each policy can be configured to execute multiple tasks on hourly, daily, weekly, or monthly basis. Once Agent policies are setup the agent can be installed on a machine. Once the agent successfully installs it will start to manage the machine as it was configured to do.

If we assume a single agent policy and delivery of the agent to be a push from the console to a group of machines it may take 10-20 minutes to configure the Agent Policy then depending on the number of machines rollout from the console may be several hours or several days to reach the entire environment. Other rollout options are available and utilizing multiple rollout options can result in faster rollout of the Agents to the desired machines, but typically will still take more time than an agentless configuration.

So, if Agents require a longer rollout time why would they be ideal in some cases? Agents typically come up in three cases as being a better option than agentless.

Laptops users

Laptops come and go in environments today. With the Agent loaded on a machine it will not matter if the Laptop is not online during my maintenance window. The agent policy can be configured to fill in the gaps and supplement agentless coverage to give a solid support model for these hard to reach road warriors.

Secure Environments

DMZs are a good example of an environment that may not allow file and print sharing regardless of firewall rules and security measures in place. With the agent the target can be locked down and only require an outbound port to the console to update its policy. This port is 3121 (Inbound port on the console). Internet connectivity may be required if the admin specifies Vender over the Internet as a source for data and patches. And if internal distribution servers are to be utilized the DS will require 139, 445, or if using http DS whatever port IIS is configured on for the virtual directory.

Low Bandwidth Connections

The agent offsets the scans to the local machine. One of the advantages of Agentless is that the bulk of the processing is done on the console. This involves network traffic and high CPU usage on the console, but the target is virtually unaware of any of this. If the agent runs the scan locally we reduce the WAN traffic from 2-4mb on average to 20kb to 100kb on average to accurately scan and report the results to the DB. An agent that also utilizes DSs for deployment purposes would do all of its major traffic on the local LAN and only a fraction of the WAN traffic would be required reducing the impact on lower bandwidth connections.

AGENT ROLLOUT OPTIONS

Deploying the agent to an environment is no small task no matter what the product. Installing a piece of software on a machine that has to meet local prerequisites as well as communicate with a remote machine to pull down policy data involved a great many variables that can complicate matters. Then there is the delivery mechanism. Some environments may require multiple ways of installing an agent to do a successful rollout. Shavlik has simplified the Agent Installation down to a single universal MSI. It is the same installer for all customers. Whether you use your local copy or grab a copy from xml.shavlik.com directly, if you had the full path available it would be the same msi. This gives us a payload that can be delivered using a number of different methods.

Push Install from Console

Using Shavlik's Agentless technology you can push an agent install out by simply selecting a machine from the machine group or from machine view and installing an agent with the policy desired. This is the quickest and easiest way to rollout the agent, but is bound by the same requirements our Agentless scans are bound by.

Manual Installation

In cases where there are machines that are not reachable agentless, but there are very few to consider a manual agent install allows execution of the MSI and entering of a few variables (Console URI, Passphrase or Credentials, and Policy) to allow install of the agent. This can be done quickly and easily for small numbers of machines, but becomes less viable the larger the target base becomes.

Scripted Installation

The AgentInstaller.msi allows cmd line execution so install can be scripted and delivered in a number of different ways. The command line options are all detailed in the help files under "Manual Agent Installation Script" in the help index. Delivery of the AgentInstaller.msi and execution using a cmd line can be delivered in a number of different ways quickly and easily. The problem here typically is how many agent policies need to be rolled out. Each policy means a different script and then sorting out which machines get what can be more difficult.

GPO

One way to deliver multiple agent policies to the environment quickly and easily is through Group Policy. Shavlik has a tool to create an MST. The MSI and MST delivered to a specific OU will ensure the proper policy is installed on each machine. Switch out the MST for different policies and you can rollout as many variations as you like to your environment.

Custom Patch

The ability to push the Shavlik agent from the Shavlik NetChk Protect console is nice to have, but if I am pushing the agent to 1000 machines in a machine group setup by an IP range and I hit 90% of the targets first time through what do I do with the remaining 10%? Using NetChk Protect's custom patch feature we can scan the same group and detect if the agent is installed and only install on machines it is not installed on. This can be helpful in many ways. Every so many weeks or months an additional scan can be run to pick up machines that may have slipped through the build process and not received an agent.

Other

From the options above one can get creative with scripting options and deliver the agent in any number of ways. The MSI can be wrapped in a self extracting zip to extract and execute cmd line install to be delivered via email (MSI is roughly 5mb) to reach some users who are rarely in office. The same could be delivered through a hosted weblink. Click on the download and run the file to execute. In this case the user executing would need local admin rights, but in many cases where the users is at a standalone site or heavy remote user this is often the case. For customers who image using ghost or sysprep or other means you cannot install the agent and make it part of your image due to the nature of how the agent registers itself with the console for security purposes. You can imbed the scripted install into the image so first time booting it can run the agent install and be up and running as part of the build process.

PORT REQUIREMENTS AND FIREWALL CONFIGURATION

These port requirements can also be found in the help files. With the following tables an admin can configure firewalls in the environment and on the local machines to allow proper traffic in and out of machines for NetChk Protect to manage the environment.

	Inbound Ports (Basic NAT Firewall)							
	TCP 80	TCP 135	TCP 139 OR TCP 445		TCP 3121	TCP 4155	TCP 5120	TCP 443
Client System		X (For asset scans)	X	X		X (For listening agents)	X	
Console System					X			
Distribution Server	X		X	X				X

	Outbound Ports (Highly Restricted Network Environment)					
	TCP 80	TCP 139 OR TCP 445		TCP 3121	TCP 5120	UDP 9
Client System	X (For agents)	X	X	X (For agents)		
Console System	X	X	X		X	X (for WoL & error reporting)
Distribution Server						

If exceptions need to be made to the external firewall to allow download of patches, Service Packs, Engines and XML the following would need to be added to the firewall rules. This list will evolve over time as Shavlik adds support for additional vendors and is subject to change.

- xml.shavlik.com
- License.shavlik.com
- <http://updates.sunbelt-software.com> (threat protection 7.x)
- <http://hfnetchk4.shavlik.com> (threat protection 7.x)
- download.microsoft.com
- download.adobe.com
- www.real.com
- [ftp.mozilla.org](ftp://ftp.mozilla.org)
- www.winzip.com
- qtinstall.info.apple.com
- <http://javadl.sun.com>
- <http://fpdownload.macromedia.com>
- <http://download.skype.com>

SQL DATABASE MAINTENANCE

If you are at a company that is running Shavlik products on a full SQL environment and have a DBA on staff with SQL maintenance and backup policies already running against our DBs, great! If you are running SQL Express or full SQL but don't have a maintenance and backup plan in place, please keep reading.

DB instability and corruption is the single biggest cause of an upgrade issue that is encountered and the root cause of many GUI performance issues that can be mitigated and, in many cases, resolved by proactive maintenance on the DB. Below are our recommendations for good regular maintenance on your DB so you keep it running slim and clean for good performance and to reduce issues.

Keep in mind this is a starting point. If you have regulatory needs that require more data kept live you should adjust to keep more data live. If that is the case you may want to analyze how frequently you are scanning. 1000 agents scanning 8 times a day will grow your DB at a much more rapid rate than once per day or once per week. And in most cases, you don't really need all of that data.

The following are our recommendations for regular DB maintenance.

Data Retention

Determine the amount of data that needs be kept on hand for operational purposes. Typically 60-90 days is acceptable for operational purposes. Configure PurgeOldProtectData utility to cleanup anything older than that number of days and schedule task to run monthly to clean up the DB. (Read on in the next section of this doc for details on setup for PurgeOldProtectData)

Reporting

Determine what report data is required for audit/regulatory requirements. Run monthly reports fulfilling these needs and keep on file as far back as policy requires. Typically 13 months is acceptable.

DB Backups

It is recommended to run weekly incremental and monthly full backups. The backup should be run just before your scheduled your purge. Keep backups as far back as the reporting data.

DB Maintenance Schedule:

- **Backups:** full monthly, just after patch maintenance for that month. Incremental weekly, end of each week (after weekend patch windows preferably).
 - **Purge Data:** After Full Monthly backup is run
 - **Reindex:** After Purge Data is run
 - **Integrity:** After Reindex is run
-

Full SQL Maintenance Guidance

If you are using full SQL it is easiest to setup maintenance plans using the maintenance wizard. Microsoft has some documentation around common SQL maintenance at the following link including how to use the SQL Wizard to setup and maintenance plan:

<http://www.networkworld.com/subnets/microsoft/110107-ch8-sql-server.html?page=2>

If you are using SQL Express the maintenance wizard is not available. In that case you can use the SQLCMD command line interface to run stored maintenance procedures or you may look into some tools created by DBAs to wrap these commands into an easier interface. One tool that works very well is ExpressMaint. Using either of these options you can write a script to handle the maintenance and schedule using the Microsoft Scheduler on the frequency you desire.

<http://www.sqldbatips.com/showarticle.asp?ID=29>

Example script for SQL Express to do a full backup, reindex, and integrity check using the ExpressMaint utility:

```
Expressmaint -S (local)\SQLEXPRESS -D ShavlikScans -T DB -R C:\Expressmaint -RU WEEKS -RV 1 -B C:\Expressmaint -BU WEEKS -BV 1 -V -C
```

```
ExpressMaint -S (local)\SQLEXPRESS -D ShavlikScans -T REINDEX -R C:\Expressmaint -RU Weeks -RV 1
```

```
ExpressMaint -S (local)\SQLEXPRESS -D ShavlikScans -T CheckDB -R C:\Expressmaint -RU Weeks -RV 1
```

PURGE OLD DATA USING A POWERSHELL SCRIPT

We have had requests from many of our customers to provide an easy way to purge old data from NetChk Protect, and we now have made it available through a PowerShell Script. NOTE: To use this script you must be running NetChk Protect v.7.2 or later. This script enables you to execute a purge of data older than xx number of days. Below you will find details on prerequisites to run the script and instructions on how to set it up.

Prerequisites

- NetChk Protect 7.2 or later – Download [Here](#).
- Windows Powershell – Download [Here](#).
- Open Shavlik’s PurgeOldProtectData.zip File – Open [Here](#).

Steps to Setup

1. Upgrade to Protect 7.2 or later.
2. Install Microsoft Powershell 1.0 (link to download page provided above, make sure you download the version for your System OS)
3. Extract PurgeOldProtectData.ps1 to c:\program files\shavlik technologies\netchk (Link Provided above)
4. Create PurgeScript.bat file – in c:\program files\shavlik technologies\netchk\ and paste the following syntax in the bat file replacing values for SQL instance, DB name, and number of days to purge as necessary:

```
powershell -command "set-executionpolicy Unrestricted"
```

```
powershell -command "& .\purgeoldprotectdata.ps1 -sqlinstance  
"sqlserver\instance" -database "ShavlikScans" -purgeAfterDays xx -timeout 30"
```

```
powershell -command "set-executionpolicy restricted"
```

5. Testing - For testing purposes I like to throw a pause command at the end of the bat in case an error occurs while testing. Run the bat on its own first time out to ensure it will execute as expected. First time it may take a while to purge depending on how much data there is.

Example Result:

GAC	Version	Location
False	v2.0.50727	C:\Program Files (x86)\Shavlik Technologies\NetChk\ST....
False	v2.0.50727	C:\Program Files (x86)\Shavlik Technologies\NetChk\ST....
False	v2.0.50727	C:\Program Files (x86)\Shavlik Technologies\NetChk\ST....

False v2.0.50727 C:\Program Files (x86)\Shavlik Technologies\NetChk\ST....

False v2.0.50727 C:\Program Files (x86)\Shavlik Technologies\NetChk\ST....

False v2.0.50727 C:\Program Files (x86)\Shavlik Technologies\NetChk\ST....

Delete operation complete. All data older than

+
48
+

days old was removed from the ShavlikScans database.

6. Scheduling Reoccurring Task- Once you have tested the bat successfully we can move on to scheduling. We will use the windows task scheduler to schedule the job. Create a new scheduled task browse to the bat file, set reoccurrence pattern, set credentials. Execute the task in the scheduler by right clicking and saying run to ensure the scheduled task will run as well.

Shavlik Technologies
Web: www.shavlik.com
E-mail: info@shavlik.com