Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

GIAC Certified Windows Security Administrator (GCWN)
Practical Assignment
Version 5.0

Jim Hendrick

January 27, 2005

In partial fulfillment of the requirements for GCWN Certification

Host Based Intrusion Detection:
(Staying Afloat in a Sea of Changes)

Abstract

With systems and software becoming more and more complex, the number of patches and updates increasing, and IT departments expected to manage more systems with less staff, how are we to know whether or not our systems are safe, if they are behaving badly (just because they can) or have been compromised?

In this too-brief format, I will at least outline the problem, identify a few methods for addressing it, evaluate a couple of products that attempt a solution, and document the basics of implementing one.

Acknowledgements

Table of Contents

1. The problem: Are you compromised or not?

"Bad guys" or "Bit rot"?

One of the key issues in implementing layered security is being able to identify when an incident has occurred. Many systems exhibit behavior that might indicate there is a problem. But with the complexities of various software packages, and the increasing number and frequency of patches and updates that get applied to any given system, it is difficult to know for certain whether a system has or has not been compromised. Faced with this uncertainty, IT staff will often resort to treating the symptoms (updating virus definitions and running a scan, scanning for "spyware", and visiting "windowsupdate"), and then if the symptoms seem to disappear they declare the problem solved.

Far more importantly, systems that show no signs of compromise are nearly always ignored. This leaves a virtual "playground" of systems available to intruders as long as they can go unnoticed. Consider the situation where a system (think corporate server this time for full effect) shows no signs of trouble, but is silently allowing remote access because of unnoticed changes.

Whether or not a system has been attacked, how many of us have when faced with a system problem, only too quickly realize that (since we have no knowledge about what might have changed since it did not have the problem) the best answer would be to wipe the entire system and reinstall everything from scratch? Note the implication here. It is not that we lack the *skills* to fix the problem, but rather the *knowledge* about how the system has changed, and it is faster to simply reinstall a standard image. However, this leads to loss of (user) data, and certainly productivity.

Consider if it were possible to "catch" the system at an early enough point in the chain of (change) events, *and* to know exactly what had changed. If it is not a security incident, the problem might be (more) easily fixed. And if it *is* a security incident, it could be responded to far sooner, potentially saving the cost and repercussions of it going on unnoticed.

Please note that this issue is not unique to Windows systems. Any system, whether Windows, UNIX, Mac or anything you can imagine (where the code image or data can be changed) is susceptible to unauthorized or simply undocumented changes. In fact, all devices that store a state by which they operate (including hardware based "appliances" like switches, routers, firewalls, load balancers, etc.) all are vulnerable.

Unfortunately, a very prevalent option is to "do nothing" and this becomes more costly as time goes on. Even if the problem of system monitoring is recognized, without the ability to do so the best we can do is try to "harden" them as best we can and hope for the best. We have still not changed the game in our favor.

So, what can be done? We need a way to watch our systems and let us know when something changes. There are several generic terms for applications that

do this. The two most common are either "Host Based Intrusion Detection" or "Integrity Checking". I prefer the former term, since it implies a security focus although "Integrity Checking" is certainly valid as well.

Be aware that many applications that call themselves "Host Based Intrusion Detection Systems" monitor *network connections* to or from the machine and *not* the state of the system itself. They can only check packets going across the interface (rather like a gatekeeper).[1] And after the fact, they can do little or nothing to identify whether or not an attack was successful[2]. For that reason, they do not address the problem described here, and I classify them as (very useful tools, but) addressing a different component of layered security, that being in the realm of (personal) firewalls or (Network) IDSs.

Therefore, for the purposes of this paper, I will define "Host Based Intrusion Detection" to be:

> "a system with the ability to detect whether or not a system has been changed based on the contents of its persistent data"

What should we look for?

Host-based intrusion detection systems (as I have defined them) must have the ability to identify intrusions that have already taken place against a host. How quickly they can do this from the time of the incident depends on their methods for monitoring changes (usually by being run at regular intervals). Additionally, their effectiveness is limited by how well they can discriminate between "good" changes; that are part of the normal operation of the system and "bad" changes; that either have not been authorized (perhaps done maliciously as part of an intrusion or attack) or *were* authorized, but have had an unintended affect on the system. I make this last point because even authorized changes that cause unintentional problems would benefit from the ability to identify exactly when and what changed. But this is a security paper and not one about good system management (hmmm… is there *really* a difference?)

The theory of current[3] host-based intrusion detection is fairly simple. You somehow take a snapshot of a system at a known good state, and then compare the same system at some future point to identify what changed. Surprisingly, while there are some attempts at it, there are not very many practical solutions to this problem in the Windows world.

An ideal application would be able to monitor all aspects of the system:

- *Detect* all changes to the system (no false negatives) for all files and directories including ownership, access permissions, timestamps, etc. as well as content. A special case for Windows is the registry, which is

---

[1] "Stop! Who would cross the Bridge of Death must answer me these questions three, 'ere the other side he see." Scene 23, Monty Python and the Holy Grail
[2] Even if you maintain some sort of network event trail or packet capture you can only infer that specific network behavior by your system was caused by a compromise.
[3] It is theoretically possible to alert the moment an unauthorized change took place, but this would seemingly need to be built into the filesystem and kernel functionality, and as far as I know, no such exist.

stored as files but needs much finer granularity. It would need to monitor all entries *within* the registry to detect system configuration settings. A system that alerted on any changes to the registry would not be useful in that area since the registry changes in normal use.

- *Decide* which are the "unwanted" changes (no false positives). To do this, it would (somehow) need to be aware of all valid changes including new software installed, patches applied, user files created, renamed, changed or deleted, as well as changed user preferences. It would also need to know what normal system activity to ignore such as additions to logs, registry entries reflecting changing system state, etc.

- *Respond* based on the nature of the change. This may range from:
  - Logging the change (including data about what changed and when)
  - Generating an alert (via email, SNMP, pager, etc.)
  - Taking corrective action (perhaps restoring the good data.)
  - Taking preventive action (to limit the damage. Perhaps stopping a given service or halting the entire system)

Sounds pretty nearly impossible, doesn't it? In fact it is a very difficult problem for just a single system, let alone an enterprise full of different systems with widely different user needs. But is it hopeless?

Investigating these three areas more closely may show whether an "imperfect" solution might still be "good enough" to provide real value.

Detect all changes to the system

We begin by making the assumption that for our purposes, "the system" consists of the set of files and directories that reside on its permanent storage.[4] Changes to all files on the system must be able to be monitored. And while it is possible to store a copy of each file for comparison this can also be done by examining the characteristics of the files:

  i. File size – Commonly used by people, this is extremely weak for our purposes. It is trivial to change a file and leave it the same size.

  ii. File modification time – Also very commonly used by us, and also very poor at determining file integrity. In fact, it may be considered not truly a characteristic of the file itself but rather an attribute of that file within the filesystem. Like size, it is quite easy to spoof.

  iii. Checksums – Somewhat better, since they are computed on the contents of the file. But they can be tricked relatively easily. Consider, if all I know about a set of numbers is the last digit(s) of their sum, I

---

[4] It may be possible to do something with the active memory space, or other aspects of a computer, but this is beyond the scope of this paper.

can modify those numbers as long as it remains the same.

    iv.   Cryptographic "hash" functions – These are the strongest single means of verification. Based on mathematic functions that purport to be "one-way"[5], it is assumed that any changes whatsoever to that file will perturb the resulting hash signature. *Note carefully* that using a (previous) hash signature on a file to determine if it has changed implies you have some certainty that the previous value itself has not changed. So the system must guard against this somehow.

    v.   Public-key cryptographic signatures – These provide a similar mechanism to validate the contents of a file to hash functions. In addition, they have the property of providing information about *the source* of that signature. For example, not only would you know that the file had not changed since the signature was generated, but that it was the "John Doe" key that generated the original signature. This gives some added assurance that the signatures have not been tampered with. It also implies that the signature was created by someone possessing the signing key (hopefully the rightful owner). This is useful, since if an intruder could replace the original hash of the (changed) file with one of their own, but did not have the correct key, they could not correctly *sign* that hash. Even with such mechanisms in place, It is important to store copies of critical signatures on "read only" media (burning a CD for example).

- Changes to directories – (they may be thought of as aspects of files).

    i. File modification times – If modification (or creation) times change unexpectedly, even though file contents may not have changed it could signal that an intrusion has taken place.

    ii.   Access permissions – Any change to the permissions that are granted to users or groups may also indicate signs of intrusion.

    iii.   Directory attributes – Similar to the attributes of a file, each directory (folders and subfolders) have their own attributes such as modification times and access permissions.

- The Registry – While this is really a collection of files[6], it is critical to the entire system and deserves separate attention.

    i. File characteristics (including cryptographic signatures) – These can tell you that changes have been made to the file(s) storing the registry. However, since the registry is so dynamic (storing many entries that change during normal operation), this alone will be of little use.

---

[5] "So far, no one has proved the existence of a true one-way function". Slide 15
http://www.inf.ed.ac.uk/teaching/courses/cs/0304/lecs/cryptoI-6up.pdf

[6] For a description of the Microsoft registry, visit http://support.microsoft.com/default.aspx?scid=kb;EN-US;256986

ii.   Content of specific entries – Here is where the real value of
monitoring the registry exists, if a host-based intrusion detection
system can monitor them it would be of great use.

Decide which are the "unwanted" changes

This is a harder task to be sure than simply detecting changes. It involves
monitoring the *right* files, and keeping up with approved changes to those files.
This depends more on human diligence and adherence to process than on
programmatic correctness. However, even though doing the job *perfectly* is not
realistic, let us think about how good it might have to be to be useful.

- (Minimize) False Positives – When alerts are given too often with no real
  (or perceived) problem, the level of perceived threat for subsequent alerts
  goes down so when a real threat exists, the alert may go unnoticed. In
  our situation, this is likely if changes generate "too many" alerts, including
  ones that don't signal trouble. To help this, several things can be done:
  - o Reduce the number of objects (files & directories) monitored to
    only those that are deemed critical to the system. This is
    easonable, since it is more troubling if a .dll file in %Systemroot%
    changes than if the size of the event log increases. To do this well,
    one may:
    - ▪ start by only monitoring the things you are really concerned
      with (like %systemroot%*) and add other items until you
      believe you are watching "closely enough". Unfortunately,
      this can cause a false sense of security (false negatives).
    - ▪ start by monitoring everything and prune out the alerts that
      are obviously benign. This is less likely to miss something,
      but will be more time consuming.
  - o Monitor different attributes for different objects. For example, while
    file size is a poor indicator of whether or not a file has *changed*, it
    can be useful in determining whether or not it is a *valid* change.
    Consider log files that are expected to change in size, but not to
    shrink (only grow or stay the same size). However you might want
    to monitor .dll or .exe files for *any* changes including access
    permissions as well as their content.
  - o Tie the detection system into the change control system. This can
    be a manual step to "update the intrusion detection baseline
    whenever you make system changes".
- (Minimize) False Negatives – Somewhat simpler than reducing false
  positives. It can be done by taking the "start with monitoring everything"
  strategy. But will be much easier if you have a good baseline to start with.
  For example, on Windows systems we would want to monitor
  %systemroot% more closely than "Temporary Internet Files". Once a
  reasonable set of defaults are built for basic system types; XP, Windows

2000, Windows 2003 Server, etc. they can be re-used for many systems with modifications as needed for other services (e.g. Web root on IIS)

Respond Appropriately

Given a reasonably flexible policy, it should be fairly straightforward to extend a system to taking action. These actions may be:

- *Logging the change* – This is pretty standard and should include the item affected including the characteristics that flagged the change.

- *Generating an alert* – Standard fare. Based on the "criticality" of the change, send an alert to an individual (or potentially to another system).

- *Taking corrective action* – This may be possible if you have enough information about the item *and* you have the ability to restore it to its original state. This may be as simple as changing access permissions on a folder or file. It may include replacing a changed binary if you have access to a trusted copy (similar to Windows File Protection). It should also be accompanied by logging and generating an alert.

- *Taking preventive action* – For very specific events on some special systems you may want to try to directly stop an intrusion that is under way. Consider a web server that becomes compromised. It may be desirable in some situations to stop the web service, down the network interface, or even halt the system in an effort to limit further intrusion or damage. Although I hasten to add, any time you implement a security system that reacts on its own based on the actions of a (potential) intruder, you introduce the possibility for that system to cause damage. It may even be possible for an intruder to make use of this for their own purposes (because your system is taking action based on their actions). Consider a situation where an intruder somehow knows your web server will shutdown if it detects a specific type of change. They may be able to cause a "denial of service" attack by making your own system halt itself. This is not to say that you are helpless, but that you must be aware that it is very difficult to make your system foolproof so that no possible action on the part of an intruder could make your response potentially cause another problem. So think carefully before you act.

How good is good enough?

Can less than ideal systems address the problem? I believe so. Even a limited system that can be effective at addressing these areas will be of value.

Applications can use combinations of detection methods (including modification times, access permissions, hash signatures) so that the entire state of the system can be verified with good certainty against modifications.

And while it is hard to have a system make determination between "good" and "bad", having reasonably good "generic" templates for the more common system types and providing relatively easy ways to update this valid state (as authorized

changes are made) do allow these systems to be fairly effective.

So, Detecting changes and Deciding which ones to alert are clearly possible. However, I would (again) urge restraint when allowing any system to respond in its own (except to send an alert and log events). While computers are excellent at doing specific things very quickly (like detecting a change), I am less convinced at their ability to make accurate judgment decisions on their own. I may be conservative here, but I prefer a person every time when actively responding to an incident.

2. Are there tools to help?

When I was determining which products to evaluate, I looked at several freely available tools. I had hoped that one would serve reasonably well for at least a small number of systems. Unfortunately I was disappointed, and I ended up with a commercial one. Was this then a fair evaluation? Perhaps not, but I believe the one I ultimately selected is arguably the one to beat.

The Judging Criteria

Based on the criteria for *Detecting* change, *Deciding* which ones matter and *Responding* (even if this response is simply a log entry) my criteria included:

1. Must be able to "recurse" through the filesystem (and not require an explicit list of every item to be monitored).
2. Must use more than basic file attributes (size, date) when determining whether changes had been made.
3. Must be customizable by the user to add or exclude files or directories.
4. Must be able to "update" signatures for authorized changes.
5. Should have a reasonable set of defaults for monitoring basic Windows systems.
6. Should alert on added and deleted files or directories.
7. Should be able to be scheduled to run automatically.
8. Should automatically send alerts (e.g. email, SNMP, syslog) when changes are detected.
9. Should be scalable to more than a very few systems (e.g. going from one system to ten should not require ten times the effort). [7]

A Wide Field of Options

Windows File Protection (WFP) – While this is not specifically designed as an Intrusion Detection system, Microsoft does provide a built-in feature to it's newer OSs primarily there to protect users from themselves (replacing accidentally corrupted or deleted files automatically). It does not address our needs since it can only verify MS distributed files using code signing, and cannot monitor for all system changes.

"Osiris"

This is a utility that actually performs well. It is available for download from http://osiris.shmoo.com/download.html as either source code or a Windows binary installer. The documentation is very limited (nothing comes in the

---

[7] This really therefore implies central management and reporting, and probably itself excludes any free tool.

package, rather the user is directed to the web site). The functionality is similarly sparse, but hits all the basic points.

- It recurses through the filesystems specified in its configuration file.
- It uses cryptographic checksums for monitoring file content, using other characteristics for ownership, timestamp, size and special attributes.
- Configuration is a simple text file, so adding or removing areas to be scanned is relatively simple.
- It allows the user to specify whether or not to automatically update its signature database each run (While this can be changed, by default, it will only alert the first time a change is seen. Afterward it considers that change part of the new baseline).
- It comes with some default configuration files, allows scheduled scans and is easily configured to send email with the output from each run.
- It is designed to run in a networked environment with a hierarchy of systems reporting up through a manager.

I only found Osiris a couple weeks ago, and have been gradually discovering more that can be done with it. It does appear quite Spartan, and there are security limitations due to its reliance on files on the system for configuration data, but for the price, in a small (and mixed Windows and Linux or UNIX environment) it may provide enough value to be worth a closer look.

GFI Languard System Integrity Monitor

This is a free utility from the makers of GFI Languard Security Scanner, and the features described on their web page[8] make it look quite promising. It claims to:

- use MD5 hashing to create "fingerprints" of files.
- multiple scan jobs can monitor different sets of files at different intervals.
- send alerts via email or log changes to the Windows event log.
- integrate with the "GFI LANguard Security Event Log Monitor" (S.E.L.M.) to allow centralized logging from multiple systems.

Well, the price is right (the integrity monitor is free, although the event log monitor is not) and the features look like it meets our criteria. So starting with the basic S.I.M. product, if it passes muster, I will look at the S.E.L.M. as an add on to provide the consolidated logging.
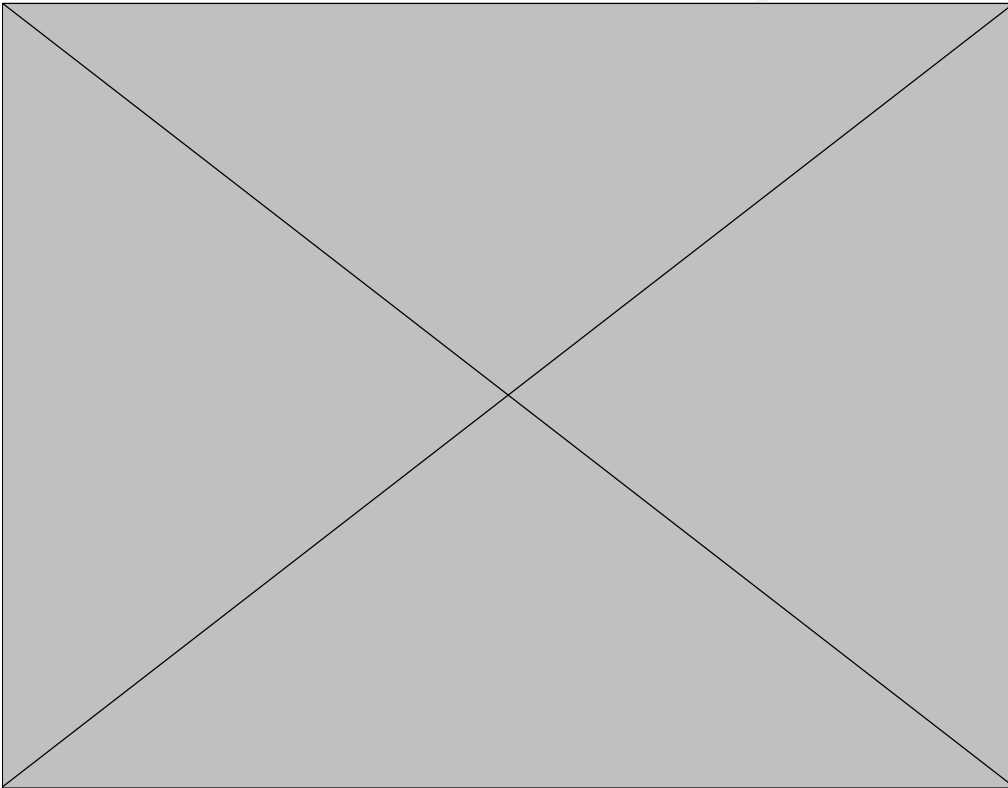
The install is a simple "click the defaults" and give it an address for email alerts.

---

[8] http://www.gfi.com/lansim/lansimfeatures.htm
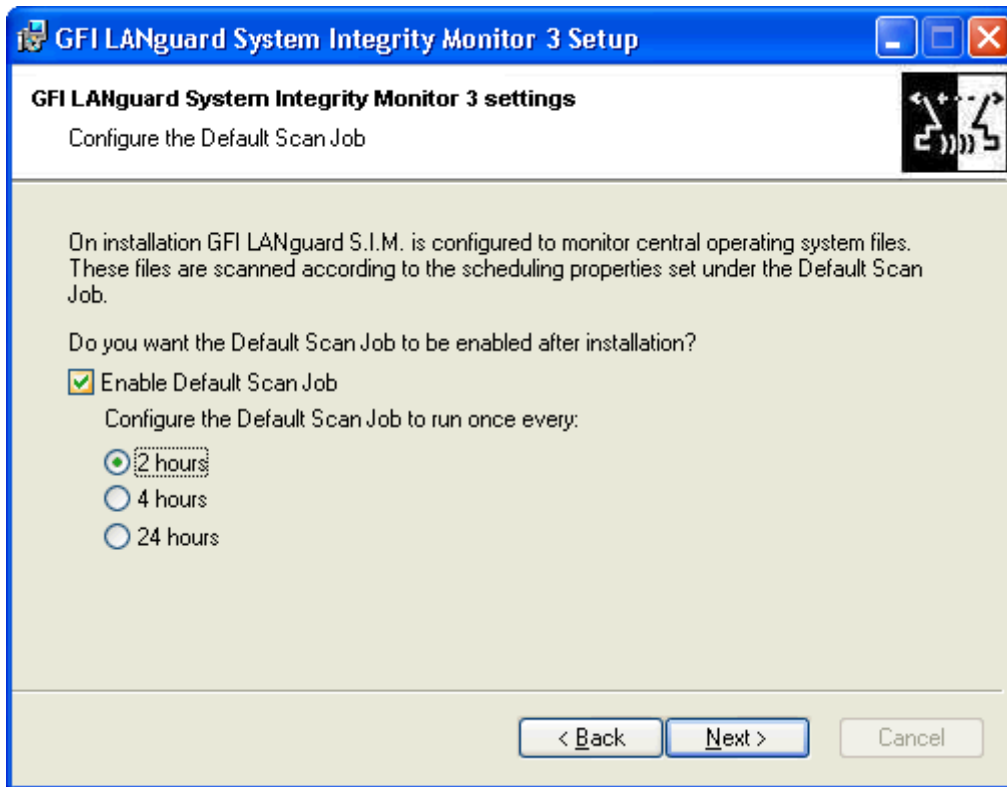
It has a "default scan" preconfigured, and gives you the option to choose how often to run the scan. These are fairly "Spartan" options for scheduling, but for the price, it seems easy to install and get running quickly.
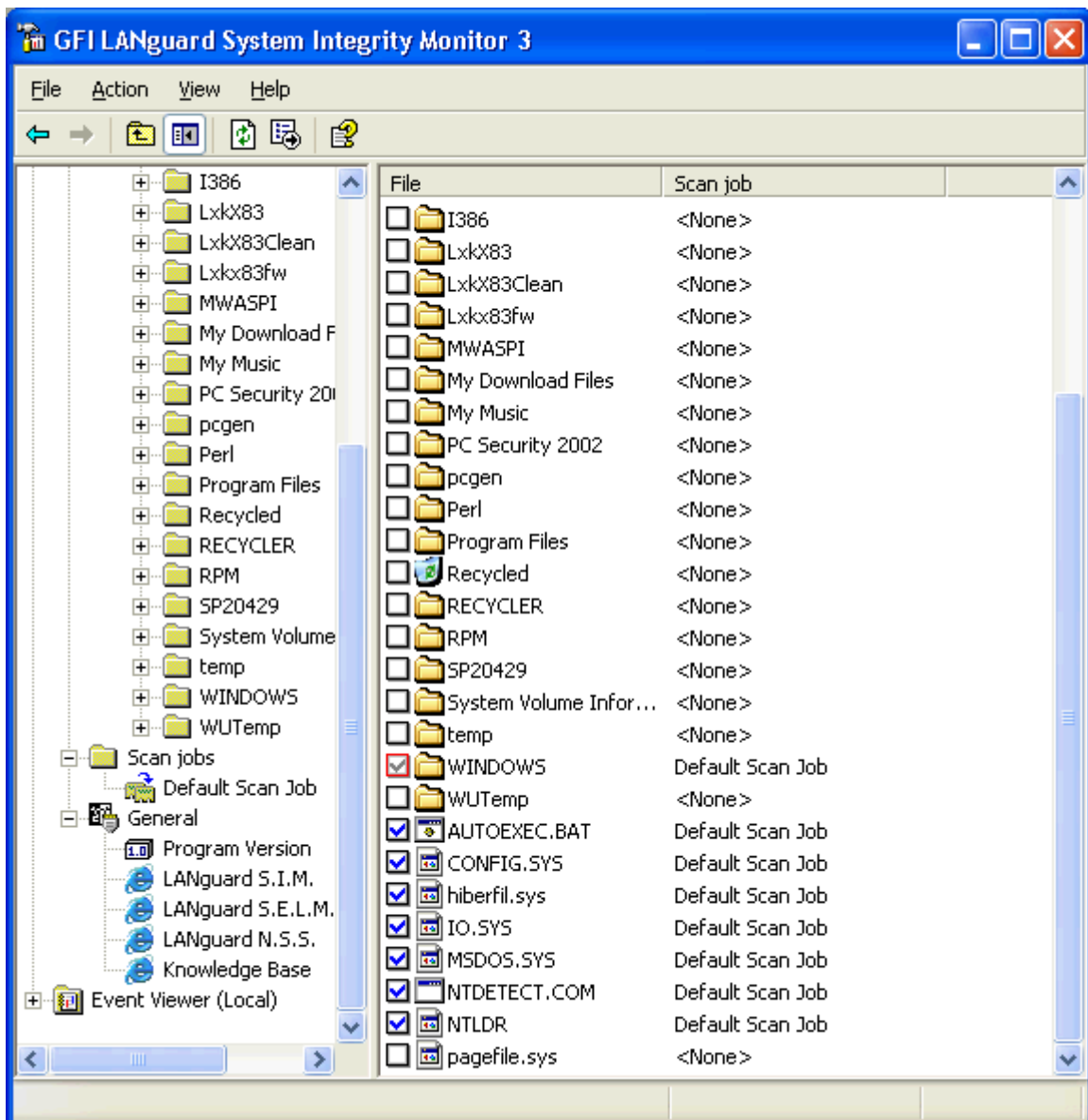
The default configuration looks reasonable for a generic Windows system, and it appeared easy to customize adding or deleting files or directories, or even to add a new scan job entirely.

On this screen, you can create several "Scan Jobs" each configured to monitor different parts of the system at different intervals.
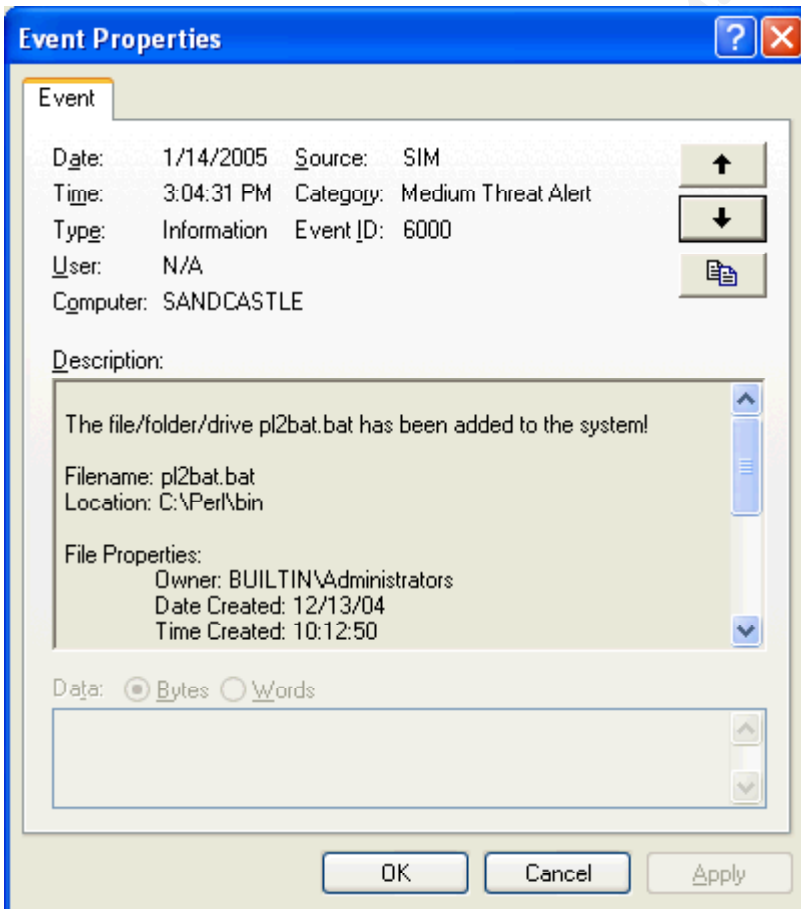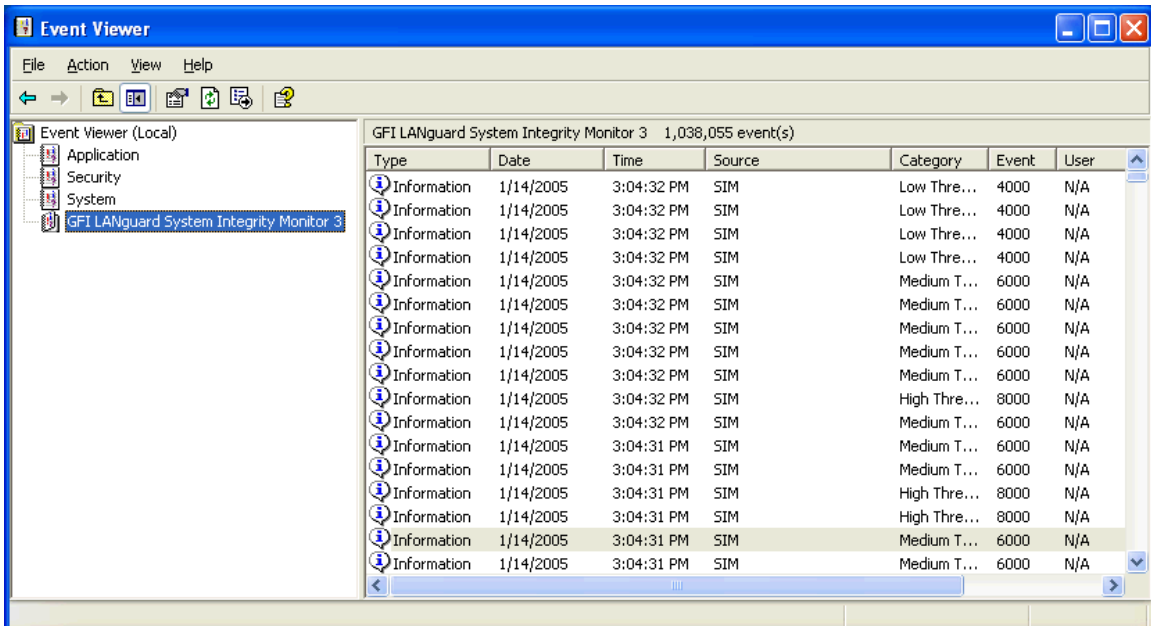
The alerting goes both to email (configurable at the scan-job level) and the Event log. These logs go to a separate event category and can provide somewhat of a "trail" of changes.

However, I did encounter several limitations during the evaluation.

- It seems difficult to handle "normal" changes. One of our critical requirements is the ability to "update" file signatures (for authorized changes) in the scan profile without deleting and creating a "new" scan task. It is not clear how that is accomplished with this monitor.

- Also with this running for several weeks, email alerts suddenly stopped working altogether. It remained running (as a service) and generates event log entries, but no longer sends email.

- It now seems to have hit some sort of limit and has occasional filesystem errors that halt the scan.
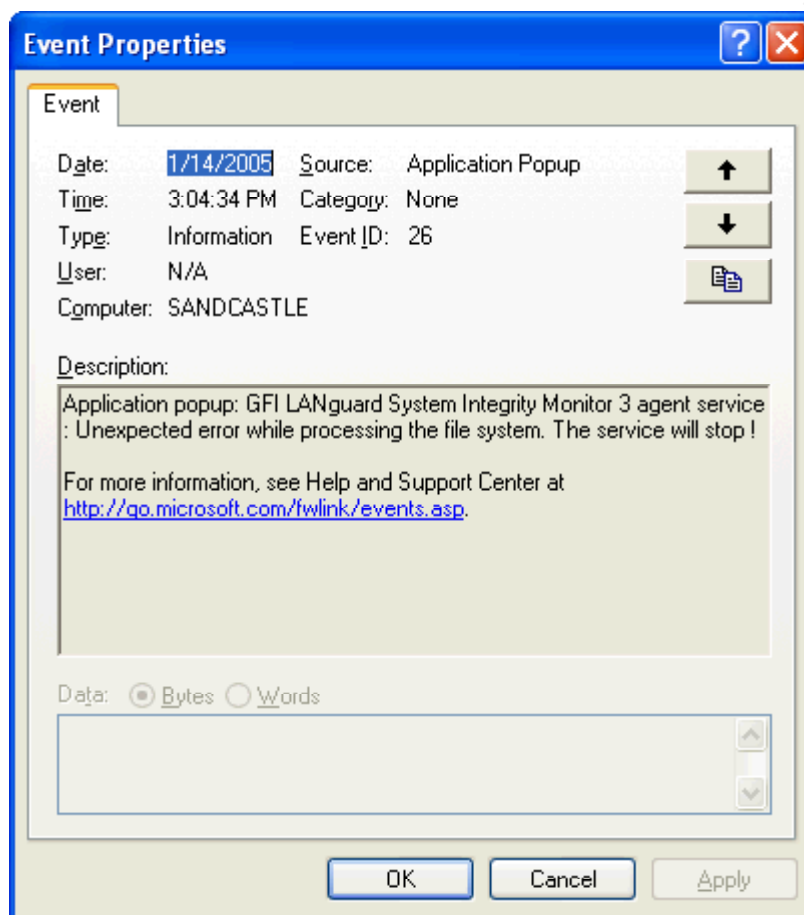
- One other minor nit is that once it was installed and configured (including schedule) it does not begin scanning until you click "Scan now" after setting the job properties. (It allows you to save the job, without prompting you to start it.)

- The events themselves also are a bit confusing. After running this I shortly had both an event log entry and an email message warning me of a "High

Threat Alert" that C:\WINDOWS\SYSTEM32\MMC.EXE had been changed! However it did not show any difference in MD5 sum (it did not show an MD5 sum *at all*) or any property other than "Time last accessed".

- Since this event (in my opinion) should be classified as a "False Positive", I wanted to modify the configuration (to ignore such "access time" changes). But after finding this file in the file default scan job, my only option was to completely remove the file from the configuration.

This lack of flexibility makes it essentially unusable for more than the most basic monitoring. And unfortunately, it doesn't appear that GFI intends to do much further work with it[9], because for the price, it looks like they are very close to having a nice tool for the small shop or individual user.

Note that While GFI's S.I.M. appears to have its limitations, the Security Event Log Monitor (S.E.L.M.) itself may provide some usefulness in this area. But because it does not provide any file checking functionality, I did not evaluate it. Note that it *might* be used with your site Security Policy and Auditing to monitor the Event Log for object access entries or other Event Log monitoring.

---

[9] The current release has been out for at least two years.

Tripwire for Servers

Tripwire has been considered the "gold standard" since it was initially developed in 1992 as a project at Purdue University by Gene Kim and Professor Eugene Spafford. Their original paper: The Design and Implementation of Tripwire: A Filesystem Integrity Checker was released in November of 1993 and the current commercial version traces its roots back to that seminal work.

My experience with Tripwire goes back to this initial work and I used the free open-source version for UNIX extensively on corporate servers in a large environment during the mid 1990s, and found it very useful. However the management aspect of keeping up with approved changes was considerable, and to be honest, there were often periods of time where the email "alerts" went past with nothing but a quick scan to see if the changed files "looked about right" for the authorized changes since the last baseline.

Tripwire has changed a lot since then, and it now consists of several products:

- Tripwire for Servers – the standalone product is the basic component of an implementation. It monitors multiple attributes, configurable to the individual file level or in user-configurable classes of objects.

- Tripwire Manager – This is really makes Tripwire shine (take it from me, it beats editing config files by hand and reading through change reports from multiple systems). It connects securely to multiple Tripwire for Servers machines, allowing central management and reporting.

- Tripwire for Network Devices – targeted at maintaining standard configurations across large enterprises, it advertises a web console and multi vendor support including Cisco, CheckPoint, Nortel and others. I did not evaluate this product.

There is a limited functionality, "simulated" version of both Tripwire for Servers (TFS) and Tripwire Manager (TWM) that demonstrates tripwire functionality through "scripted" exercises available at: http://www.tripwire.com/downloads/tmtfs/index.cfm However, in order to do a better evaluation, I requested and obtained fully functional versions for a 30 day eval. (Thanks Tripwire!!)

Following their download instructions, I generated license certificates for Tripwire Manager and Tripwire for Servers. I then proceeded to download both TW Manager (~35MB) and TW for Servers (~11MB) for Windows. Also available for download is a rather full set of documentation (which I highly recommend).

Tripwire's functionality includes:

- Flexible configuration – Tripwire has its own rich syntax for specifying what files and directories it scans and what attributes it looks for. This is

based on text files, so can be edited easily (after you learn the syntax!)

- Multiple attributes – Along with normal filesystem attributes, it is able to use several cryptographic hash algorithms to validate file integrity (including using multiple algorithms on individual critical files if desired). Tripwire will also detect changes in the *number* of alternate data streams[10] associated with an object. (Many integrity scanners miss this completely).

- Updating signatures – It is very simple to "approve" individual changes.

- Default configurations – Tripwire provides basic policies to start from.

- Alerts – are able to be sent to a file, emailed, or sent via SNMP.

- Response to alerts – Tripwire can run commands based on scan results.

To try and remain (somewhat) consistent with the other products, my evaluation consisted of using Tripwire for Servers on a single workstation. This was then extended to include Tripwire Manager in the implementation section.

Tripwire is a very simple install, with only a few decisions you need to make:

- How (or if) you want the alerts sent.

- Whether or not you will communicate with a Tripwire Manager

- Setting the Site and Local Key passphrases. Used to protect the keys that "sign" the critical files (tripwire configuration as well as filesystem data).
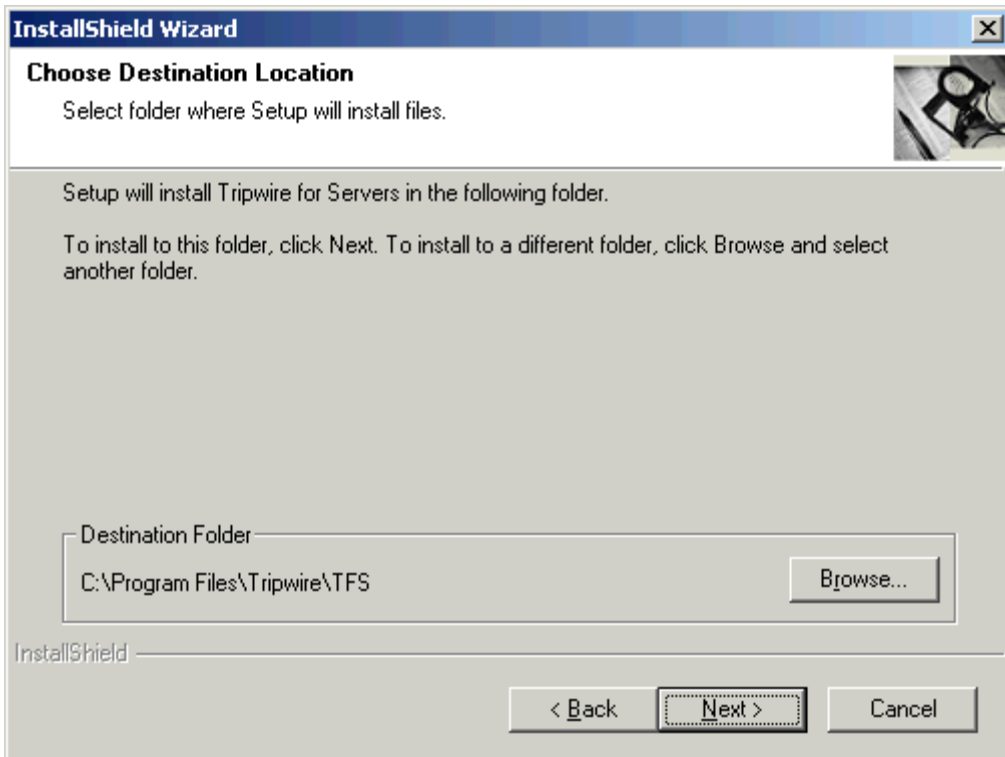
Once installed, the basic set of steps to begin using Tripwire is:

1. Create the policy – Edit the text file that describes what to monitor. (TFS comes with a good default as a starting point).

2. Install the policy - This takes the text based policy file and generates a cryptographically signed one actually used by Tripwire.

3. Run a baseline – This initializes the database of attributes created by applying the policy to your machine.

4. Run the machine – Do some normal operations (logon & off, reboot, etc.)

5. Run an integrity check – This compares the state of the system with that stored in the database according to the policy.

6. View the report - Decide what (if anything) should be modified in the policy.

7. Modify the policy – Adding, removing or changing what is monitored.

8. "rinse and repeat" – Initially, it may be best to re-generate a new baseline each time, but once you get close (i.e. small number of false positives) you can start updating the database. This simply updates the attributes for specific items without re-writing the entire baseline.

The installation itself is very simple:

---

[10] For an excellent paper on Alternate Data Streams, see:
http://www.giac.org/practical/GCWN/Ryan_Means_GCWN.pdf
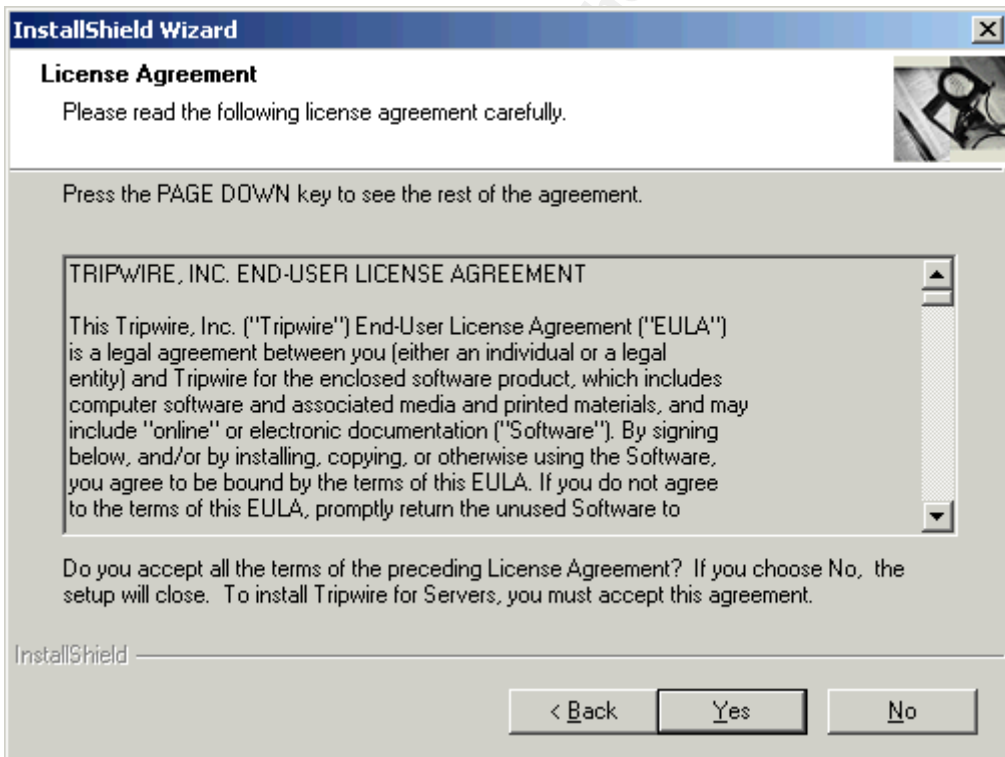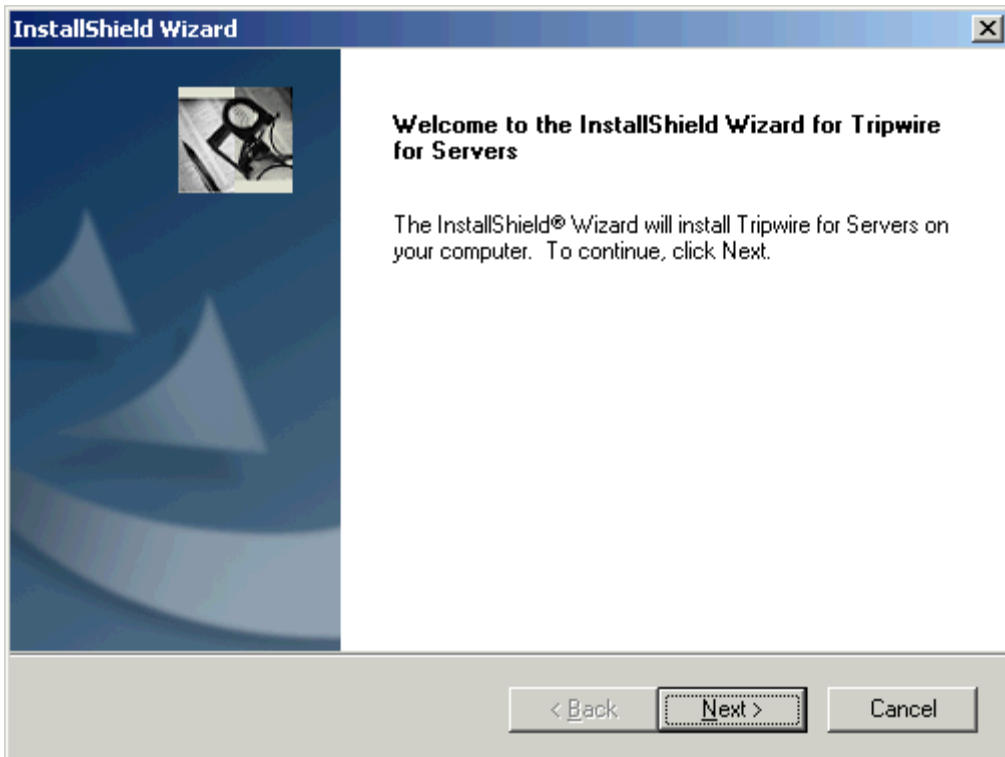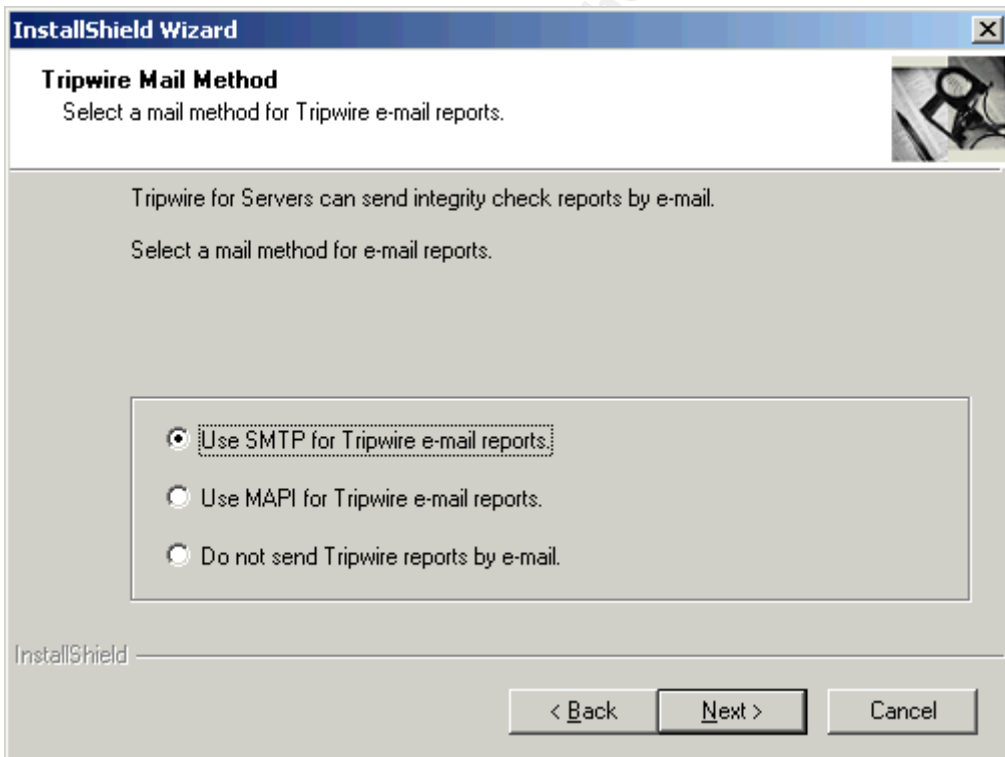
InstallShield Wizard

Tripwire Operation
Specify IP address for communication with Tripwire Manager.

If you plan to manage this installation with Tripwire Manager, specify which IP address to use for communication.

Select your preference.

○ Communicate with Tripwire Manager via this machine's default IP address.

○ Communicate with Tripwire Manager through a specific IP address.

● Run Tripwire for Servers alone, without Tripwire Manager.

InstallShield

< Back   Next >   Cancel

InstallShield Wizard

Site Key Passphrase
Specify a passphrase to protect the site key.

The site passphrase protects the site key. The site key cryptographically signs site-specific Tripwire data files. Passphrases must be at least 8 characters long and contain at least one number and at least one letter or symbol.

Passphrase :   ********************

Re-enter :   ********************

InstallShield

< Back   Next >   Cancel
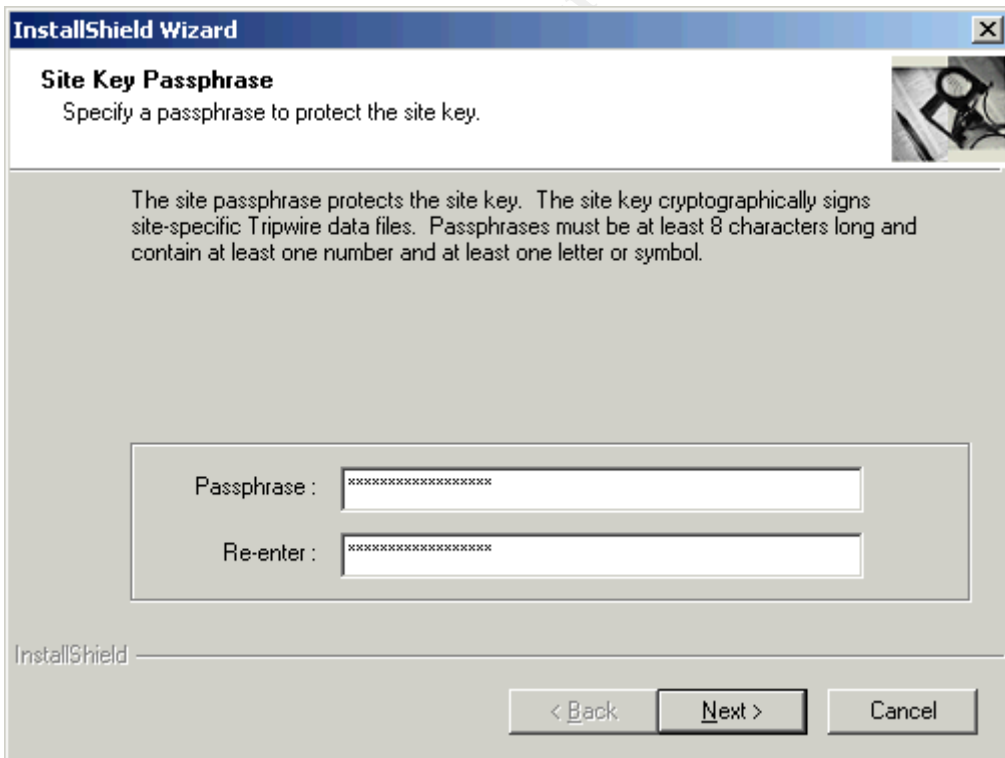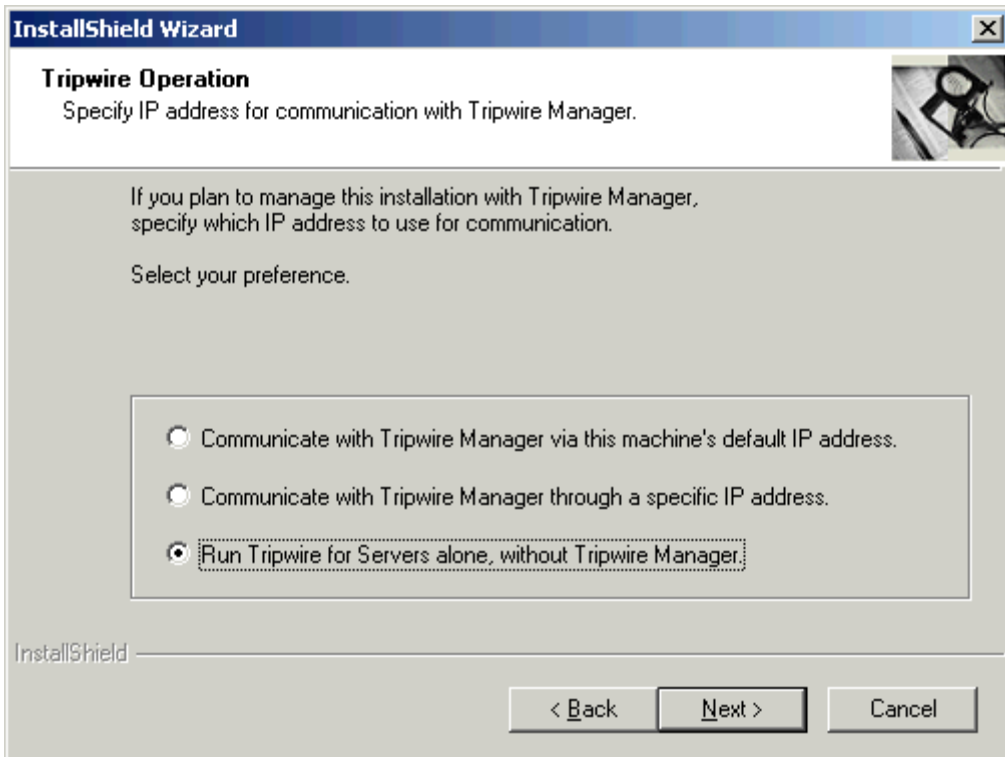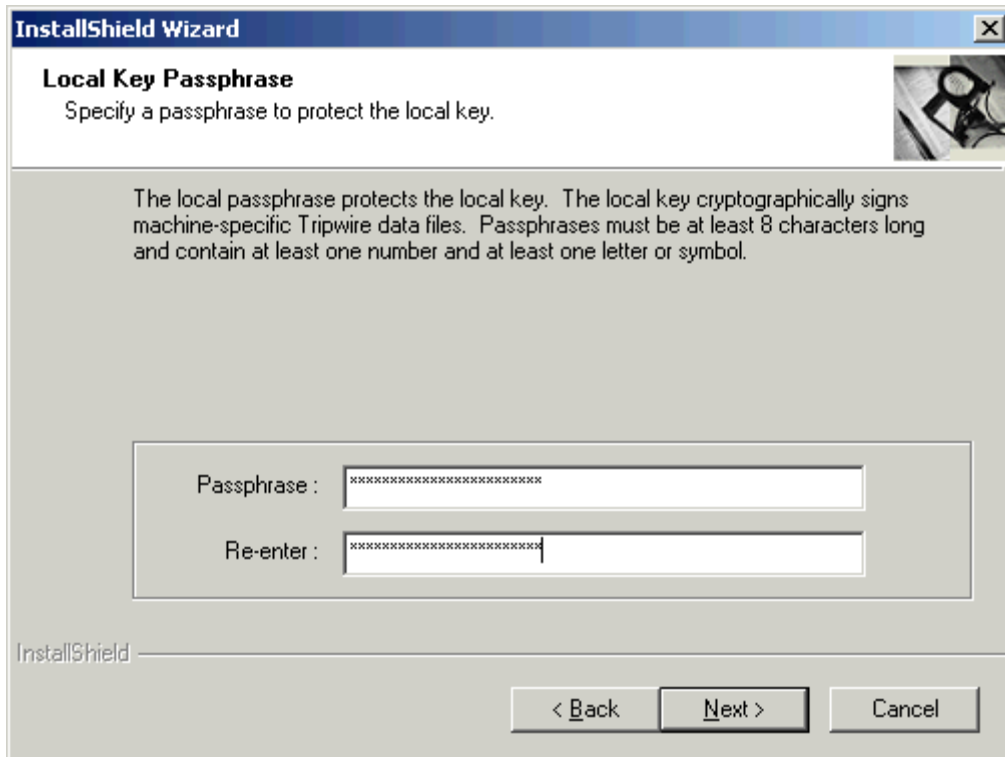
After allowing the install to add Tripwire to the PATH, and waiting for it to generate the keys from your passphrases, installation is complete.

At this point, I copied the twserver.cert file into the Tripwire\TFS\bin\ (as directed in the email that delivered the license certificate.)

Create the policy

The next step is to create the initial policy. This is the text file that describes the areas you want tripwire to monitor, and the type of attributes to be monitored in each area. Getting this configuration right is one of the most important areas of using tripwire. I cannot stress this enough. You need to become very familiar with the syntax, variables and format of this file in order to make the most of your tripwire configuration. Fortunately, it is fairly straightforward (although lengthy). Don fret too much over the first one, I suggest getting some experience running it before making many changes.

Here is a very brief example of the syntax. A full tutorial on this file is well beyond the scope of this paper. I highly recommend reading the documentation that comes with this product for a tutorial on learning policy file syntax.

This section shows some basic variable definitions. These define the attributes used for various checks. For example:

- FILE_CRITICAL defines files that should see no changes, yet should be allowed to be accessed. It is itself using another reference to "$(IgnoreNone)" which expands to cover nearly 30 attributes, checksums and cryptographic hashes and then excludes "&haval" one way hashing

function[11] and "&access" which simply ignores file access time.

- FILE_HASHES defines the set of checksum and hash calculations to be used to include (all of) a 32 bit CRC, an MD5, an SHA, etc.) but does not monitor any file attributes such as owner, group, timestamps, etc.

```
##################################################################################
#                                                                              ##
################################################################################ #
#                                                                            # #
#              Filesystem Section Variable Definitions                       # #
#                                                                              ##
##################################################################################

@@section NTFS

FILE_CRITICAL    = $(IgnoreNone) -&haval &access ;    # Critical files that may be accessed but should not have their
contents or security changed.
FILE_STATIC      = $(ReadOnly)  -&archive ;           # Files that should not change - archive bit ignored for backup
compatibility.
FILE_DYNAMIC     = $(Dynamic)   -&archive ;           # Files/directories where the content may change, but not other
attributes (e.g. owner, acl, etc)
FILE_ACCESSTIME  = $(ReadOnly)  +&access ;            # Monitor access times - if this is used, 'Reset Access Time'
*must* be set in the config file.
FILE_HASHES      = +&crc32 &md5 &sha &haval &strm_crc32 &strm_md5 &strm_sha &strm_haval ; # Monitor only the
content hashes, and ignores all other security settings.
```

Further definitions define standard system locations to make things easier to read later in the file:

```
######################
#                  ##
###################### #
#                  # #
# Path Definitions   # #
#                  ##
######################

SYSTEMDIR        = $(SYSTEMROOT)\System ;
SYSTEM32DIR      = $(SYSTEMROOT)\System32 ;


##################################################################################
#                                                                              ##
################################################################################ #
#                                                                            # #
#              Registry Section Variable Definitions                         # #
#                                                                              ##
##################################################################################

@@section NTREG

REG_CRITICAL     = $(IgnoreNone) -&haval ;
REG_STATIC       = $(ReadOnly) ;
REG_DYNAMIC      = $(Dynamic) ;
REG_CLASS        = $(REG_DYNAMIC) -&sacl ;

# Define some shorter names to things managable.

HKLM_CCS         = $(HKLM)\SYSTEM\CurrentControlSet ;
HKLM_CCS_SM      = $(HKLM)\SYSTEM\CurrentControlSet\Control\"Session Manager" ;
```

---

[11] http://citeseer.ist.psu.edu/zheng93haval.html

```
HKLM_WCV              = $(HKLM)\Software\Microsoft\Windows\CurrentVersion ;
HKCU_WCV              = $(HKCU)\Software\Microsoft\Windows\CurrentVersion ;
HKLM_WNTCV            = $(HKLM)\Software\Microsoft\"Windows NT"\CurrentVersion ;
HKCU_WNTCV            = $(HKCU)\Software\Microsoft\"Windows NT"\CurrentVersion ;
HKLM_Services         = $(HKLM)\SYSTEM\CurrentControlSet\Services ;
HKLM_EventLog         = $(HKLM_Services)\Eventlog ;
HKCU_Windows_Policies = $(HKCU)\Software\Microsoft\Windows\CurrentVersion\Policies ;
WINDOWS_REG           = $(HKLM)\Software\Microsoft\Windows ;
```

Installing the policy

Once the initial configuration is set you need to *encode and install* it. This uses
the keys previously generated, to make it more secure (if these config files were
to be changed, you would not be checking what you thought you were. Signing
these files provides an added layer).

```
C:\Program Files\Tripwire\TFS>twadmin -create-polfile Policy\twpol.txt
Please enter your site passphrase:
Wrote policy file: C:\Program Files\Tripwire\TFS\Policy\tw.pol
```

Of note is how "clean" tripwire is, in that it stores all its files in one directory
hierarchy and (shouldn't this be the norm for all software?) tells you exactly what
it is modifying.

Running a baseline
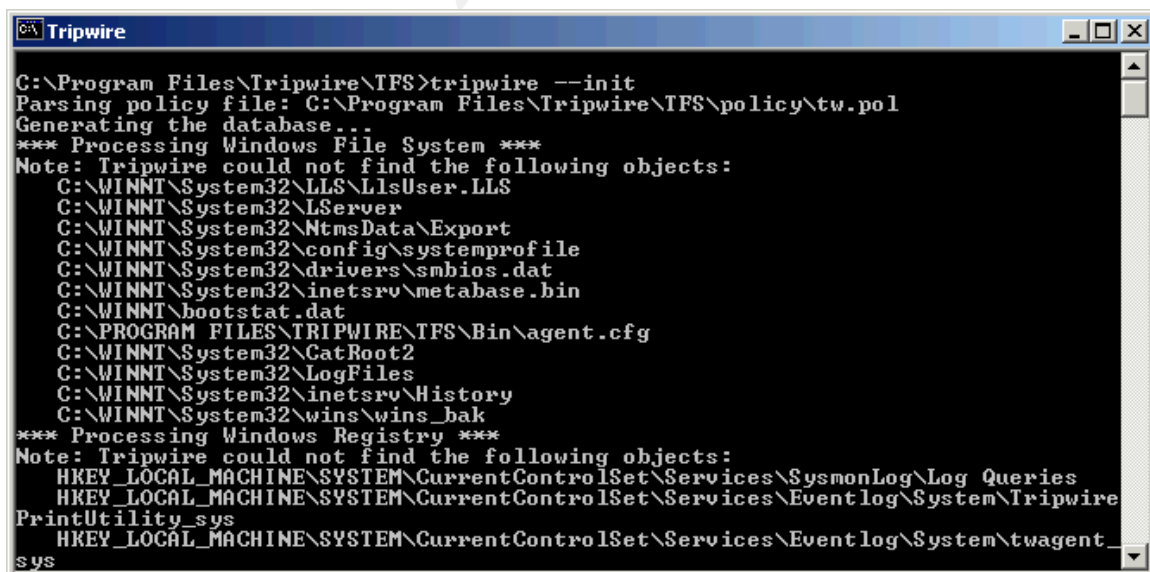
Next step is to run Tripwire to generate the database using your new policy.

```
C:\Program Files\Tripwire\TFS>tripwire -init
```

Tripwire will then apply the defined policy to the objects in the filesystem and
create a database to be used later when looking for changes. If objects are
defined in the configuration, but not present on the system, Tripwire will notify
you (so you can remove them from the config).

As you can see, the default config file "expected" to find some files and registry entries that were not present on the system. This is (really) the beginning of tuning. At this point, you should edit the config file to remove the entries for items that are not present (or be prepared to live with these warnings until you do).

Running an integrity check

Now, when you run tripwire in "integrity checking" mode, it compares the saved attributes and lets you know of any changes.

```
Command Prompt                                                    _ □ ×
Rule Summary:
================================================================
----------------------------------------------------------------
  Section: Windows File System
----------------------------------------------------------------

  Rule Name                      Severity Level   Added   Removed   Modified
  ---------                      --------------   -----   -------   --------
  Critical System Startup files  1000             0       0         0
  (C:\)
  OS Support Files               35               0       0         0
  System32 Folder                100              0       0         0
  Network Configuration Files    100              0       0         0
  Critical Drivers               35               0       0         0
  System Folder                  35               0       0         0
  (C:\WINNT\System)
  Program Files Folder           35               0       0         0
  (C:\Program Files)
* Tripwire for Servers Configuration Files
                                 1000             1       0         0
  Tripwire for Servers Executables
                                 1000             0       0         0
  Tripwire for Servers Log and Support Files
                                 1000             0       0         0
  Temporary Files Folder         15               0       0         0

Total objects scanned:  5,708
Total violations found:  1
```



```
Command Prompt                                                    _ □ ×
----------------------------------------------------------------
  Section: Windows Registry
----------------------------------------------------------------

  Rule Name                      Severity Level   Added   Removed   Modified
  ---------                      --------------   -----   -------   --------
  Hardware keys                  35               0       0         0
  Service Registry Keys          100              0       0         0
  Critical Tripwire Registry keys 1000            0       0         0
  Critical Security Account Keys 1000             0       0         0
  Security Information keys      100              0       0         0
  Local Admin Activity           1000             0       0         0
  (HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F4)
  Local Admin Login              1000             0       0         0
  (HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F4!F)
  Local Admin Password Change    1000             0       0         0
  (HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F4!V)
  Guest Account Activity         1000             0       0         0
  (HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F5)
  System Startup Executables     1000             0       0         0
  Critical System Registry Keys  100              0       0         0
  Software keys                  35               0       0         0
  Current User Registry keys     15               0       0         0
  Class keys                     35               0       0         0

Total objects scanned:  74,151
Total violations found:  0

================================================================
Object Detail:
```

```
Command Prompt                                                    _ □ ×

================================================================
Object Detail:
================================================================

----------------------------------------------------------------
  Section: Windows File System
----------------------------------------------------------------

----------------------------------------------------------------
Rule Name: Tripwire for Servers Configuration Files (C:\PROGRAM FILES\TRIPWIRE\T
FS\DB)
Severity Level: 1,000
----------------------------------------------------------------

  ----------------------------------------------
  Added Objects: 1
  ----------------------------------------------

Added object name:  C:\PROGRAM FILES\TRIPWIRE\TFS\DB\database.twd

----------------------------------------------------------------
  Section: Windows Registry
----------------------------------------------------------------

No violations.

================================================================
Error Report:
================================================================
```

```
Command Prompt                                                    _ □ ×

================================================================
Object Detail:
================================================================

----------------------------------------------------------------
  Section: Windows File System
----------------------------------------------------------------

----------------------------------------------------------------
Rule Name: Tripwire for Servers Configuration Files (C:\PROGRAM FILES\TRIPWIRE\T
FS\DB)
Severity Level: 1,000
----------------------------------------------------------------

  ----------------------------------------------
  Added Objects: 1
  ----------------------------------------------

Added object name:  C:\PROGRAM FILES\TRIPWIRE\TFS\DB\database.twd

----------------------------------------------------------------
  Section: Windows Registry
----------------------------------------------------------------

No violations.

================================================================
Error Report:
================================================================

No Errors

----------------------------------------------------------------
*** End of report ***

Report generated by:
Tripwire(R) for Servers version 4.5.0.178 for Windows(R) Operating Systems
Tripwire is a registered trademark of Tripwire, Inc. All rights reserved.
Integrity check complete.

C:\Program Files\Tripwire\TFS>_
```

Page 30 of 77

© SANS Institute 2005   Author retains full rights.

As you can see, Tripwire is very thorough in its reporting, and (using a default config file) so far has found no *changes* to the system files but did alert us that a new file had been *added* (the tripwire database we just generated).

Example of introducing change

Now let's introduce something. I copied the twconfig.txt to twconfig.txt.DIST, made some minor changes (nothing that affects policy, simply changing the name that received the email reports).

```
Command Prompt                                                    _ □ ×
Performing integrity check...
Wrote report file: C:\Program Files\Tripwire\TFS\report\04725-20050105-215751.tw
r


Tripwire Integrity Check Report version 4.0.0
Tripwire(R) for Servers version 4.5.0.178

Report generated by:        hendrick
Report created on:          Wed, 05 Jan 2005 21:57:51 -0500
Database last updated on:   Never

================================================================================
Report Summary:
================================================================================

Host name:                  04725
Host IP address:            192.168.0.11
Host ID:                    S-1-5-21-2056409907-1297513098-1761878145
Policy file used:           C:\Program Files\Tripwire\TFS\policy\tw.pol
Configuration file used:    C:\PROGRA~1\Tripwire\TFS\bin\tw.cfg
Database file used:         C:\Program Files\Tripwire\TFS\db\database.twd
Command line used:          tripwire --check

================================================================================
```

```
Command Prompt                                                    _ □ ×
================================================================================
Rule Summary:
================================================================================

--------------------------------------------------------------------------------
  Section: Windows File System
--------------------------------------------------------------------------------

  Rule Name                       Severity Level   Added   Removed  Modified
  ---------                       --------------   -----   -------  --------
  Critical System Startup files   1000             0       0        0
  (C:\)
* OS Support Files                35               0       0        1
  System32 Folder                 100              0       0        0
  Network Configuration Files     100              0       0        0
  Critical Drivers                35               0       0        0
  System Folder                   35               0       0        0
  (C:\WINNT\System)
  Program Files Folder            35               0       0        0
  (C:\Program Files)
* Tripwire for Servers Configuration Files
                                  1000             2       0        1
  Tripwire for Servers Executables
                                  1000             0       0        0
  Tripwire for Servers Log and Support Files
                                  1000             0       0        0
  Temporary Files Folder          15               0       0        0

Total objects scanned:  5,709
Total violations found:  4

--------------------------------------------------------------------------------
```

Page 31 of 77

```
Command Prompt                                                      _ □ ×

-------------------------------------------------------------------------
  Section: Windows Registry
-------------------------------------------------------------------------

  Rule Name                          Severity Level   Added   Removed  Modified
  ---------                          --------------   -----   -------  --------
  Hardware keys                      35               0       0        0
  Service Registry Keys              100              0       0        0
  Critical Tripwire Registry keys    1000             0       0        0
  Critical Security Account Keys     1000             0       0        0
  Security Information keys          100              0       0        0
  Local Admin Activity               1000             0       0        0
  (HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F4)
  Local Admin Login                  1000             0       0        0
  (HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F4!F)
  Local Admin Password Change        1000             0       0        0
  (HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F4!V)
  Guest Account Activity             1000             0       0        0
  (HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F5)
  System Startup Executables         1000             0       0        0
  Critical System Registry Keys      100              0       0        0
  Software keys                      35               0       0        0
  Current User Registry keys         15               0       0        0
  Class keys                         35               0       0        0

Total objects scanned:  74,151
Total violations found:  0
```

```
Command Prompt                                                      _ □ ×
Object Detail:
=========================================================================


-------------------------------------------------------------------------
  Section: Windows File System
-------------------------------------------------------------------------


-------------------------------------------------------------------------
Rule Name: OS Support Files (C:\WINNT)
Severity Level: 35
-------------------------------------------------------------------------

  -------------------------------------------------
  Modified Objects: 1
  -------------------------------------------------

Modified object name:   C:\WINNT\hpbafd.ini
            MD5 Expected   38f3fe823ac1758ff23b4080a38b5648
*               Observed   d1a5015ed85f98e440cf9d96e8408b41

          CRC32 Expected   99d12847
*               Observed   9500978e

     Write Time Expected   Wednesday, January 05, 2005 08:55:27 PM
*               Observed   Wednesday, January 05, 2005 09:48:57 PM




-------------------------------------------------------------------------
Rule Name: Tripwire for Servers Configuration Files (C:\PROGRAM FILES\TRIPWIRE\T
FS\Policy)
Severity Level: 1,000
-------------------------------------------------------------------------
```

```
Command Prompt                                                          _ □ ×
--------------------------------------------------------
  Added Objects: 1
--------------------------------------------------------

Added object name:  C:\PROGRAM FILES\TRIPWIRE\TFS\Policy\twpol.txt.DIST

        --------------------------------------------------------
  Modified Objects: 1
--------------------------------------------------------

Modified object name:  C:\Program Files\Tripwire\TFS\Policy\twpol.txt
           Size Expected   38,358
*               Observed   38,790

           SHA Expected    5c30299d2e49ff5f63d8edb1aabef88c977866c1
*               Observed   b087aea9a00f484825487738377c0b1a5bd50b0f

           MD5 Expected    0ad59b7a1e5bc0ccdce3c5ce44861596
*               Observed   53b4e0ff7406531d584dcedd300982e5

         CRC32 Expected    fe9c4b58
*               Observed   ad129435

    Write Time Expected    Wednesday, January 05, 2005 08:33:52 PM
*               Observed   Wednesday, January 05, 2005 09:03:26 PM



--------------------------------------------------------------------------
Rule Name: Tripwire for Servers Configuration Files (C:\PROGRAM FILES\TRIPWIRE\T
FS\DB)
Severity Level: 1,000
```

```
Command Prompt                                                          _ □ ×
--------------------------------------------------------------------------
--------------------------------------------------------
  Added Objects: 1
--------------------------------------------------------

Added object name:  C:\PROGRAM FILES\TRIPWIRE\TFS\DB\database.twd

--------------------------------------------------------------------------
  Section: Windows Registry
--------------------------------------------------------------------------

No violations.

==========================================================================
Error Report:
==========================================================================

No Errors

--------------------------------------------------------------------------
*** End of report ***

Report generated by:
Tripwire(R) for Servers version 4.5.0.178 for Windows(R) Operating Systems
Tripwire is a registered trademark of Tripwire, Inc. All rights reserved.
Integrity check complete.

C:\Documents and Settings\hendrick>_
```

Note that it correctly identified the added twpol.txt.DIST and the modified
twpol.txt. It also found another change from "normal" system operation:
WINNT\hpbafd.ini.

The Only Real Choice

Tripwire clearly meets the initial requirements. It detects changes to any file on the system, recursively following directory trees (or not) according to a user configurable policy. Tripwire also allows updating signatures for identified changes to individual files (although not shown here, running from the command line you simply use "tripwire –update" and it interactively prompts for each identified change).

At this point, I declared it the clear winner and moved on to the next phase.

3. Implementing Tripwire with Tripwire Manager

When writing this implementation guide, rather than attempt to duplicate the detail that is in the Tripwire documentation, I followed a more "organic" model, including detailing several minor problems I encountered along the way in hopes it will help someone else avoid them.

Planning the Deployment:

As you can see from the initial evaluation, running a large site using only Tripwire for Servers and trying to manage updates, integrity checks, and reports would quickly become overwhelming.

Tripwire Manager goes a long way toward making this administration as easy as possible. The console offers many views into the enterprise, a group of systems, or a single system. It also allows you to manage the configuration files and databases for every server using the GUI while maintaining the text format for the config files, so they can be used without the management console, and sections can be easily copied from system to system.

Since maintaining the configuration files and monitoring the results is where you will spend a lot of your time, I strongly recommend using Tripwire Manager to govern the operations and reports generated by Tripwire for Servers.

The communication model is client-server using encryption, and you can use a single or multiple managers, (though only one manager can make changes to a server at a given time.) The workload on the manager machine is quite low, so this can easily be installed on an administrator's workstation.

Each TW server must be "registered" with a TW manager (it is possible to import a list using a text file for registering large numbers of servers with a manager). There are several pieces of information provided for each machine:

- machine name and IP address

- group – a tripwire concept that allows managing sets of similar machines more easily. Not required

- TCP port # - This defines where the Tripwire for Servers Agent will listen for connections from the Tripwire Manager.

- site-passphrase & local-passphrase – Used to provide authentication and encryption of the critical Tripwire files.

The first time you run TW Manager, you need to create an authentication key, and also select a "console" passphrase. TW imposes some basic restrictions on the strength of this passphrase.

Tripwire Concepts:

Tripwire files– Tripwire uses a standard set of files to control its operation and interaction. These are:

- Policy file – controls what is monitored using rules and system objects
- Database file – the result of applying a policy to a given system, it stores attributes about system objects to be compared to identify changes
- Configuration file – parameters for how tripwire acts on a given system
- Report files – the output of an integrity check
- Site and local key files – store keys used in signing tripwire files to provide additional security against changes to those files themselves.
- Agent configuration files – controls how a server communicates with the tripwire manager.

Grouping machines – Tripwire allows (but does not require) sets of machines to be grouped for ease of management. Machines can be moved between groups so your decisions are not "locked in".

Creating "policies" – Tripwire policies are the set of rules by which system changes are flagged as significant or not. This is the most time consuming part of the implementation (and in fact, never ends, as ongoing approved system changes, patches, updates, etc. all may require changes to the policy). The key point here is to stress the administrators come to know their site (what is normal for my site may be an alert at yours, etc.)

The quick start guide recommends starting with a single server, tuning the policy on it, and then distributing this to other similar servers. I caution that even similar servers will likely require some custom tuning.

Running Tripwire - For each server, the TW Manager machine must:

1. push the policy
2. run a baseline (initializing the database) for that machine
3. perform some normal operations (including a server reboot)
4. run an integrity check
5. view the report
6. modify the policy
7. "rinse and repeat"

Note that the temptations when tuning policies are to either:

- monitor all files for all changes – resulting in a huge amount of "false positives". The danger is that administrators who see them become conditioned to "noisy reports" with many changes, don't have time to properly fix the policy, and start ignoring the reports
- tune the policy to eliminate all errors – potentially resulting in missing critical components and a "false negative" situation. Consider the example of not monitoring the webroot on an IIS server (because the web site changes often), and then have the site defaced and tripwire did not even report it! It is much better to adjust and watch only the attributes

(like access controls) on folders, monitoring content only on critical files.

Be aware that a balance takes time to achieve, and a successful approach for you must depend on the amount of resources you have to dedicate to monitoring and managing TW. If you have enough staff, you may choose to monitor liberally, and let your administrators tune the policies to reduce the "noise". This has the advantage of your being fairly sure that unauthorized changes will be caught. However, be aware that you need to have this process integrated with your change control process (so that authorized changes do not set off false alarms).

If you are "running lean" like many shops, this approach may fail, since administrators will not have time to dedicate to it, and come to view TW as a nuisance that gets in the way of them doing their "real work". If you choose to run Tripwire in such an environment, it may be better in this case to set policies that monitor for fewer changes, and make sure you are in fact turning *up* the sensitivity regularly. Another option for the "lean" shop may be to choose fewer servers at first, and use them to create well tuned policies, then push them to more servers and have less final tuning to do.

Configuring Responses

Tripwire is flexible in its response to changes in monitored systems. The most basic is to just alert which it can do via email, writing to a log file, or sending SNMP traps (using its own MIB). Logging can take place to a file on the server, use the Windows Event log, or be sent to a "syslog" server on the network.

The syslog detail can be selected to either send:

- one summary line with total violations (added, removed or changed files)
- a separate line for each violation, showing only that a violation occurred
- a separate line including the properties that were violations of policy.

In addition, the syslog host (or hosts), the facility and priority can be specified[12]

TW for Servers also has the capability to run commands either in response to violations or as part of each integrity check according to specified triggers:

- onviolation – Runs if *a particular rule* is violated
- Global On Violation – runs if *any rule* is violated, overrides the onviolation setting for individual commands
- Always Run Once – runs a command every time an integrity check is run

Additionally, the configuration can specify limits on the processes themselves:

- Execute as User – the account under which the commands are to be run
- Max Command Processes – controls the number of processes that can be run during a single integrity check (not including commands triggered by the Always Run Once specification)

---

[12] Syslog defines events according to the logging facility (email, authorization, user or kernel level processes, etc.) and message priority (critical, emergency, informational, debugging, etc.)

Installing Tripwire Manager

Following the "twm_quickstart" guide, we "run and follow the instructions on the installer". After unzipping the compressed download, here we go…

Please read the information below.

Tripwire® Manager version 4.5
©2004 Tripwire, Inc. Tripwire is a registered trademark of Tripwire, Inc. All rights reserved.

http://www.tripwire.com/

326 SW Broadway, 3rd Floor
Portland, OR 97205
USA

InstallShield

< Back   Next >   Cancel
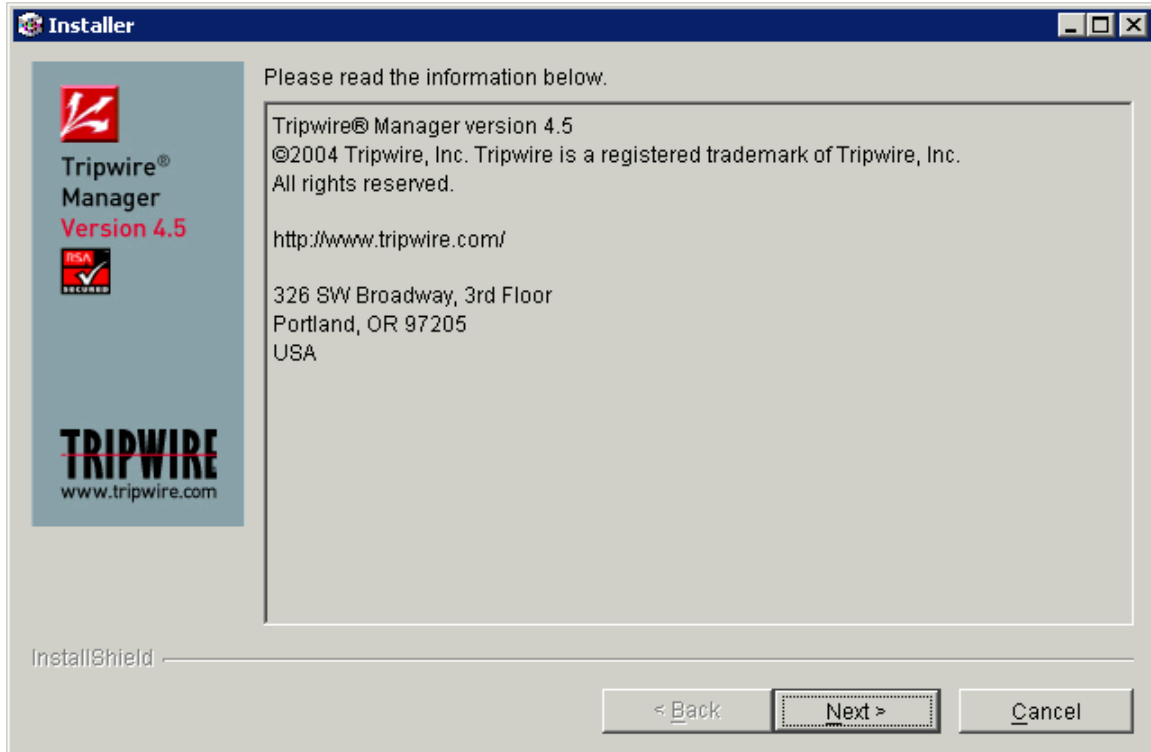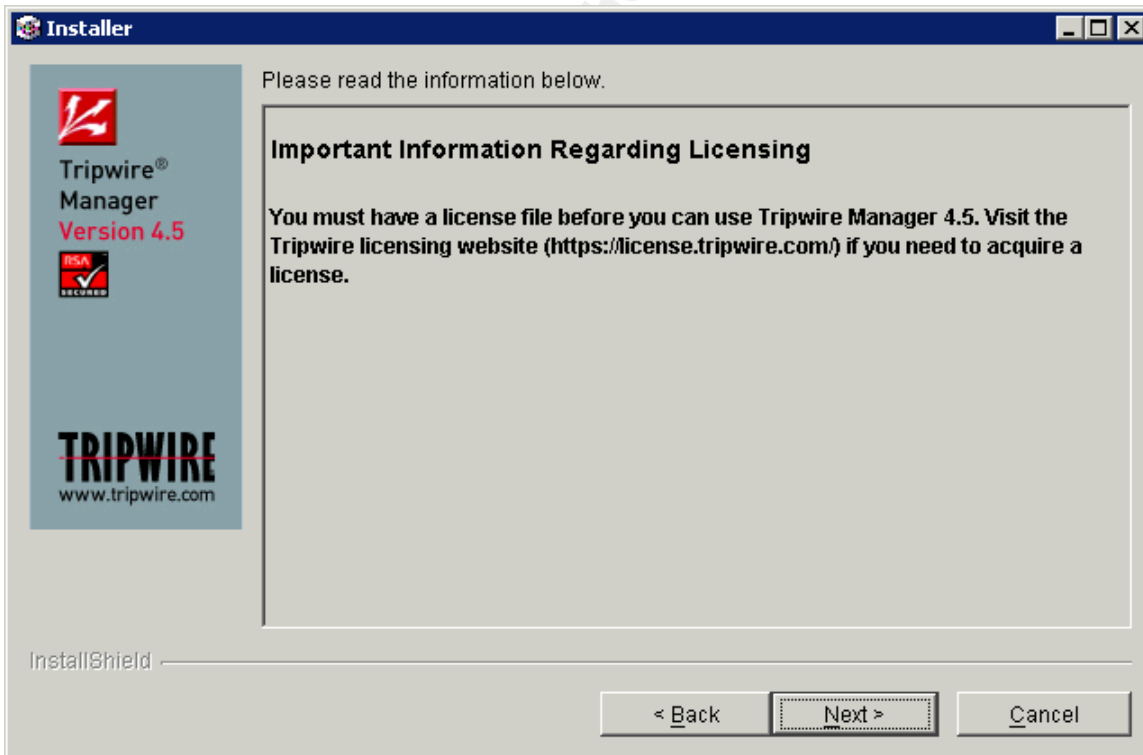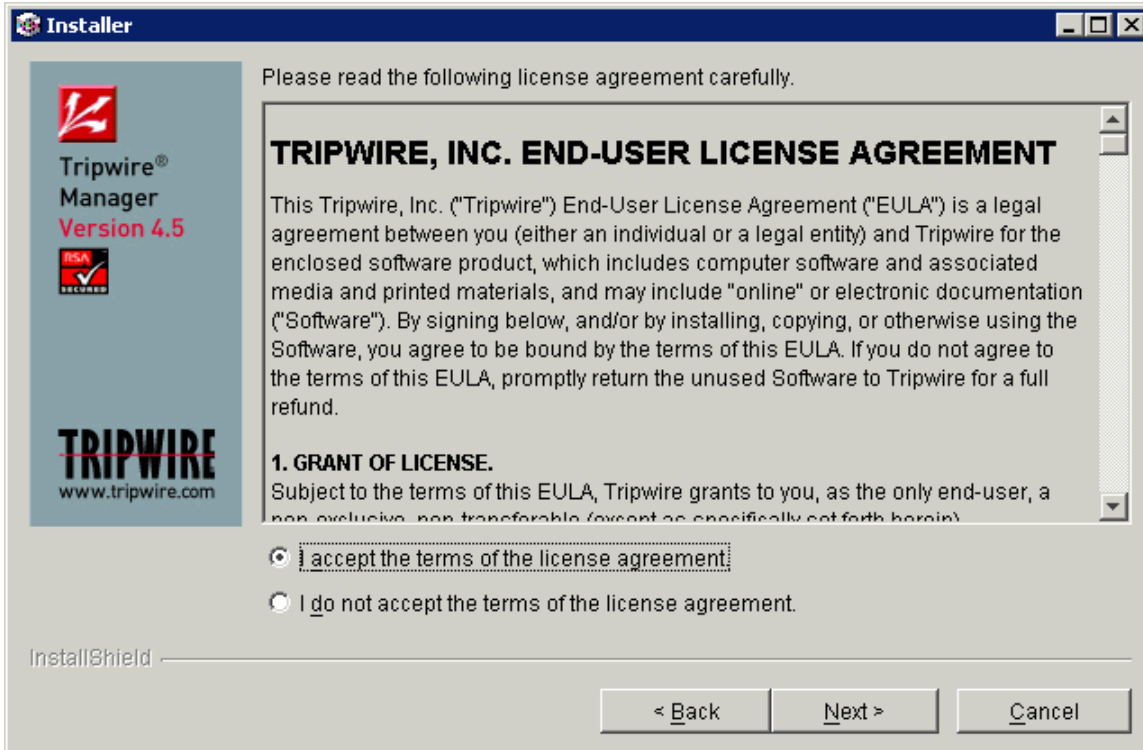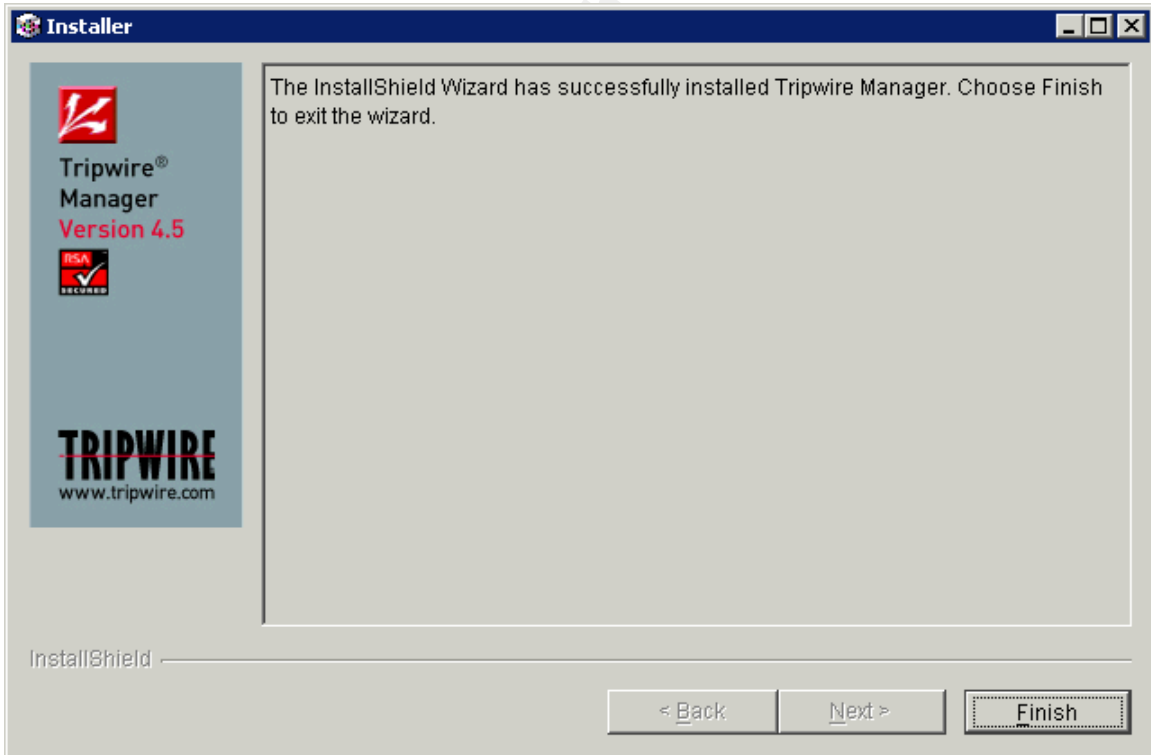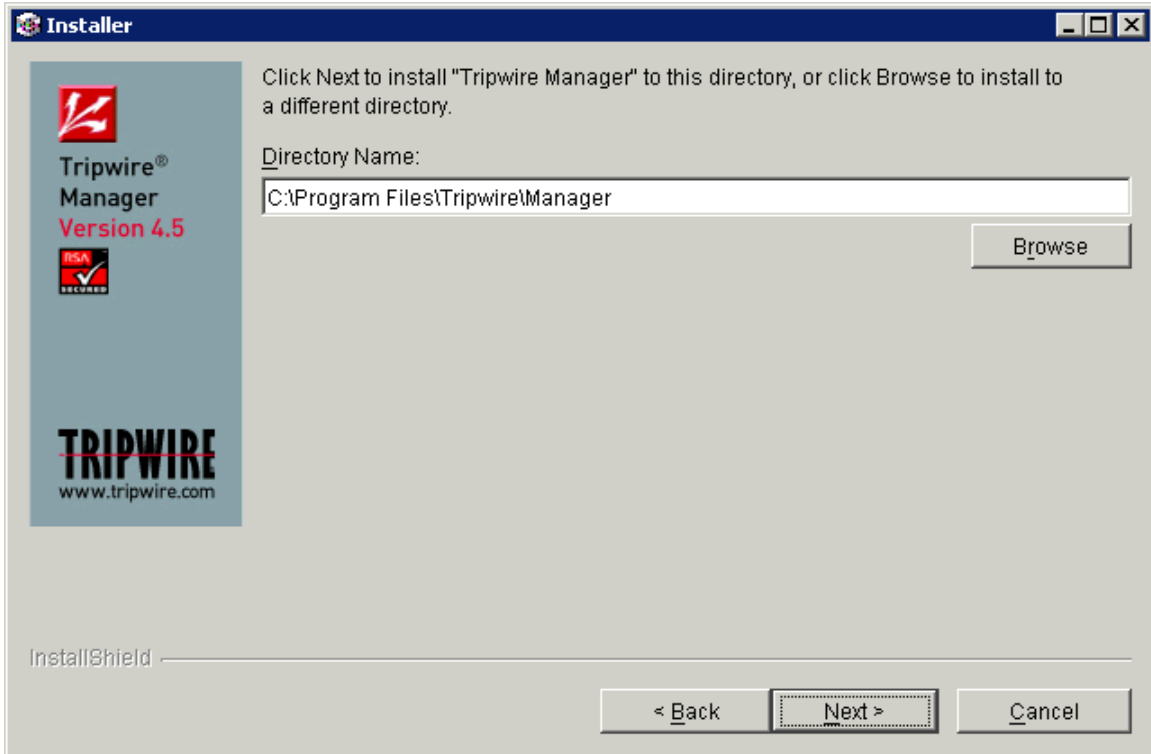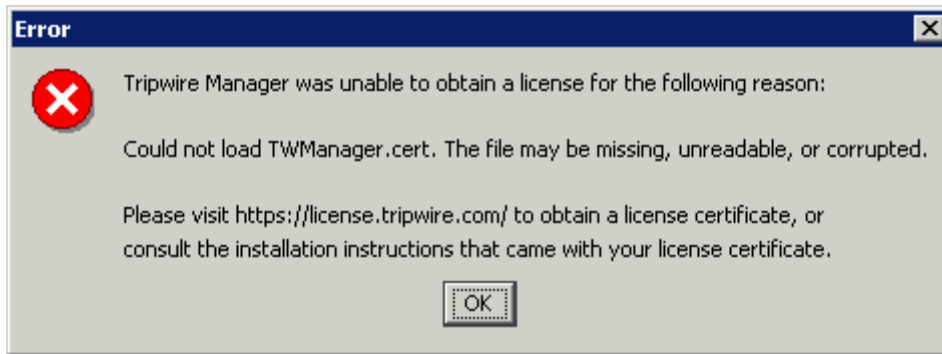
Hey, what's that? (oops, I didn't read the instructions clearly). I need to copy the certificate file into the "Manager" directory and then:
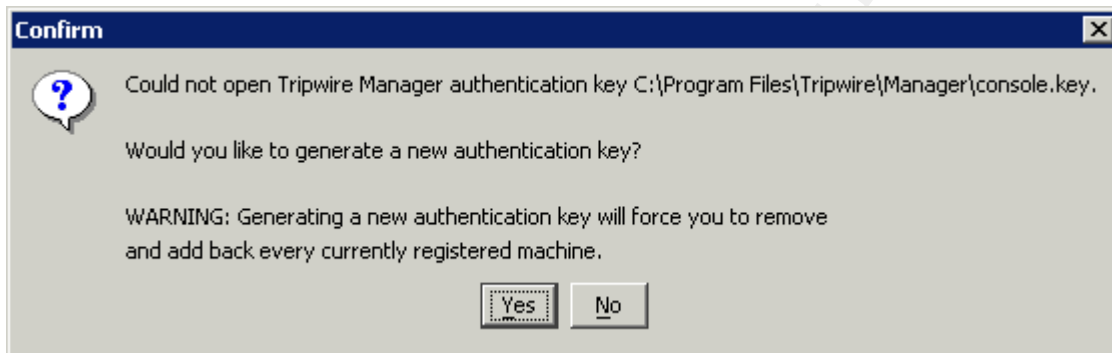
Tripwire Manager generated it's key based on a "Tripwire Manager passphrase" which must be "8 characters long, contain at least one letter, one number and one special character." This one allows connection to all your servers, so pick a good one!

Since there are no installed servers (that it knows about) it looks rather unimpressive at the moment, and we need to add the servers.

Connecting Tripwire Servers to the Manager:

The installation is essentially identical to the one shown in the evaluation section, except at the point where you select "Tripwire Operation":
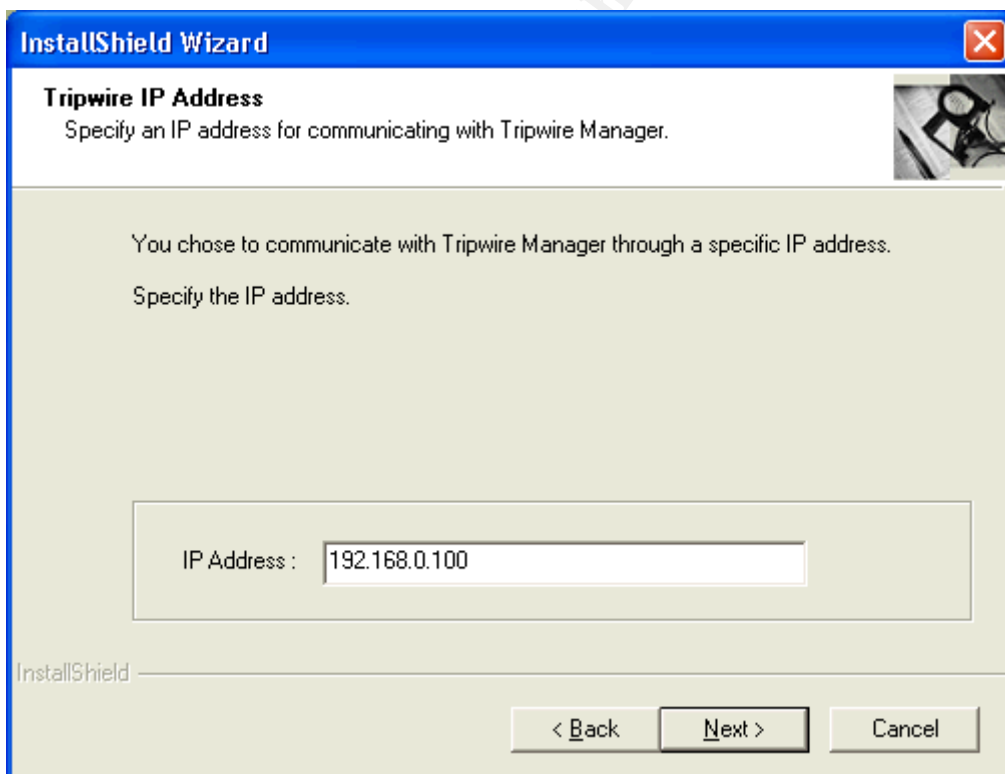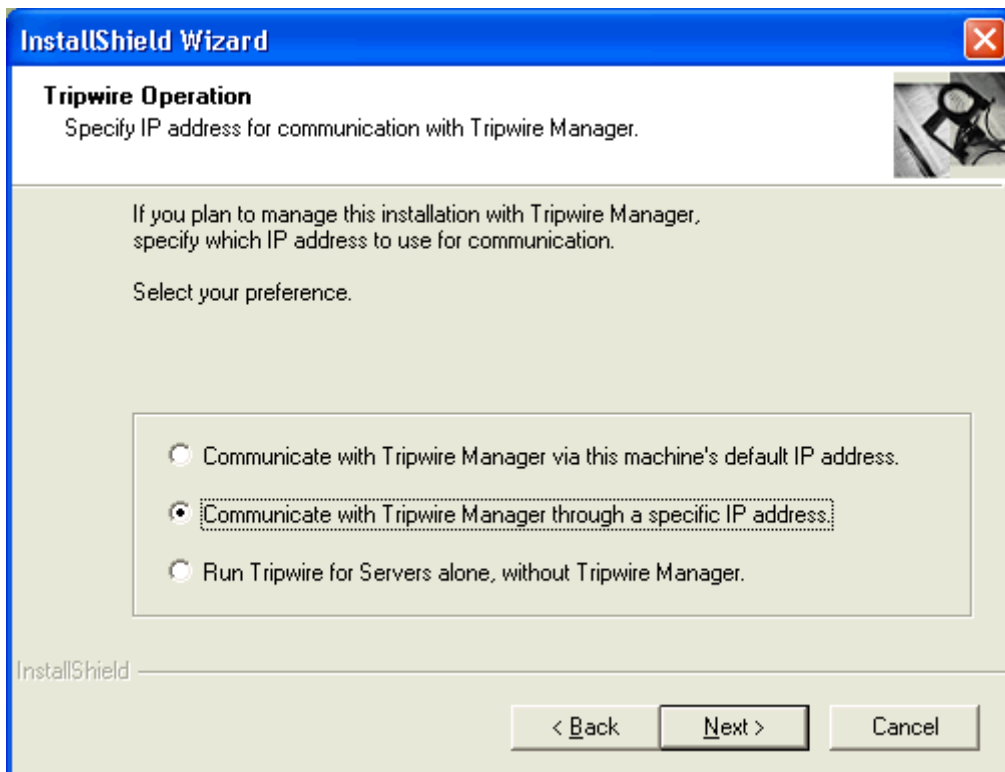
Note, the default was "this machine", which I changed since it looked (to me) as if it wanted the address of the Tripwire Manager. As it turned out, this was an error, as I will show later. It really wants the address that the server will *listen* on. However, the troubleshooting process also is an example of the support

information available at Tripwire, so I left it in this paper.

We are then led to specify the TCP port and account that Tripwire Agent (the component responsible for communicating with the manager) uses. In the following two screen shots. Note that this agent does run as the highly privileged SYSTEM account. It should be possible to run as a different user, but I did not investigate that option.
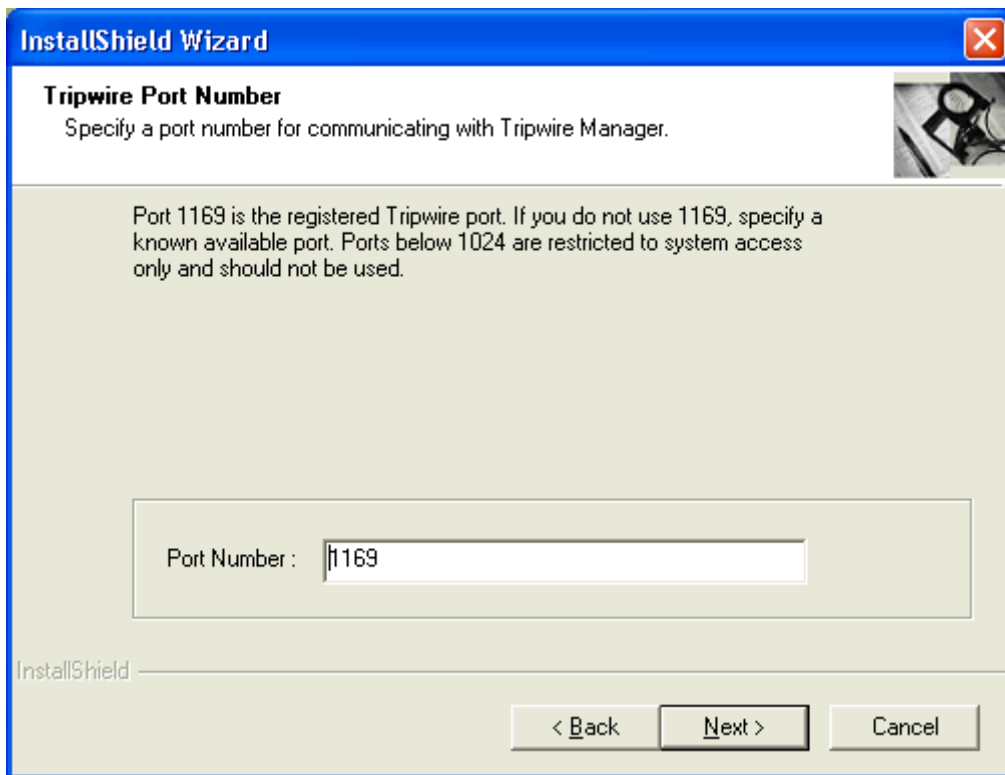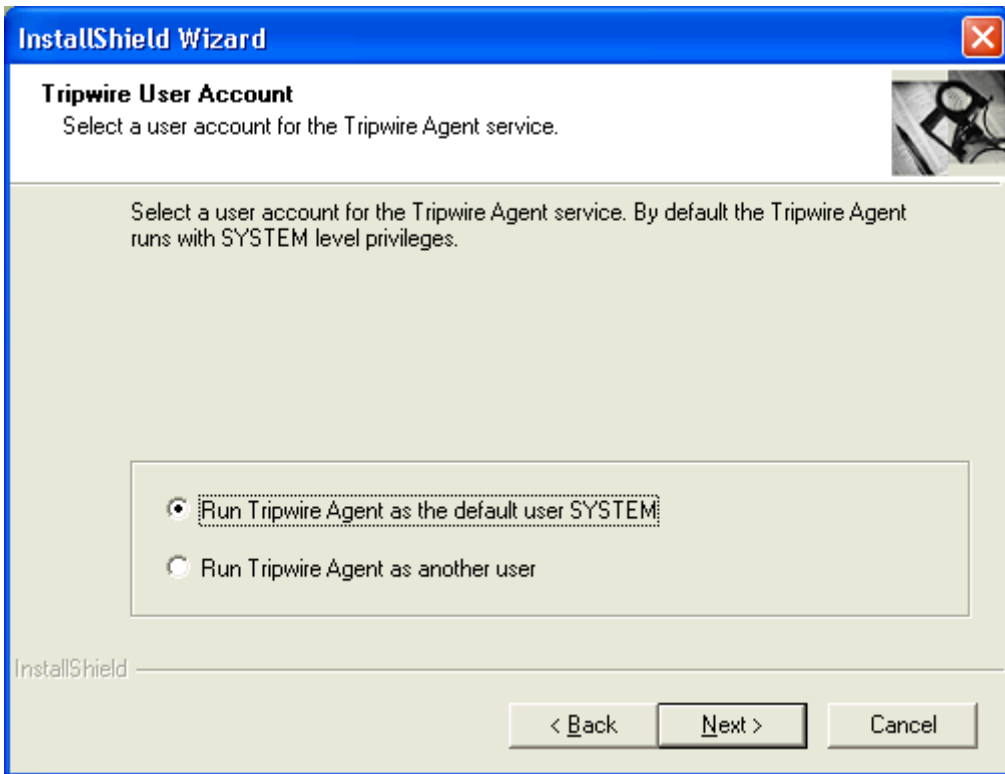
Once you select a Site and Local key passphrase, keys are generated and the configuration and policy text files are created.
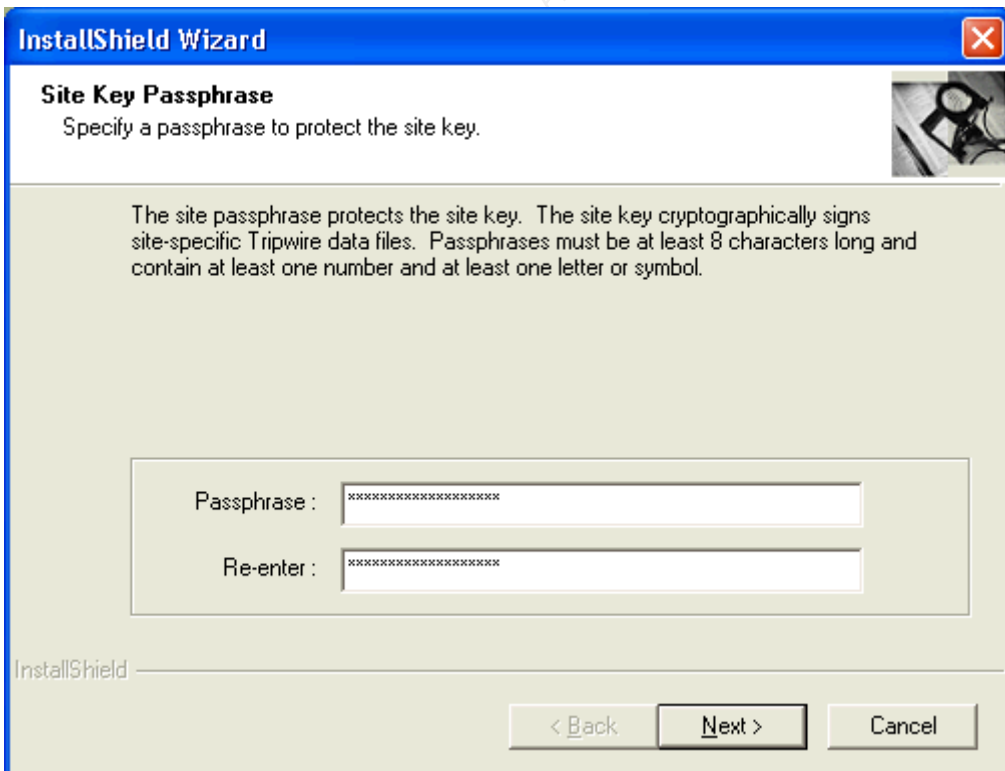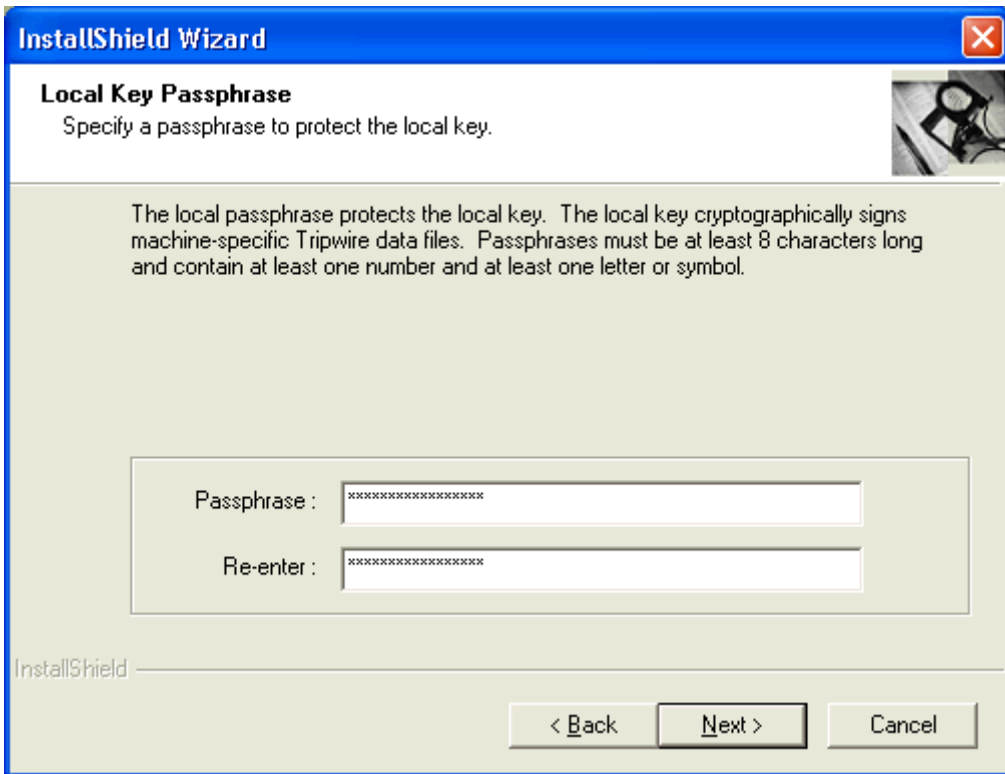
At this point, you are ready to start the Agent as a service.
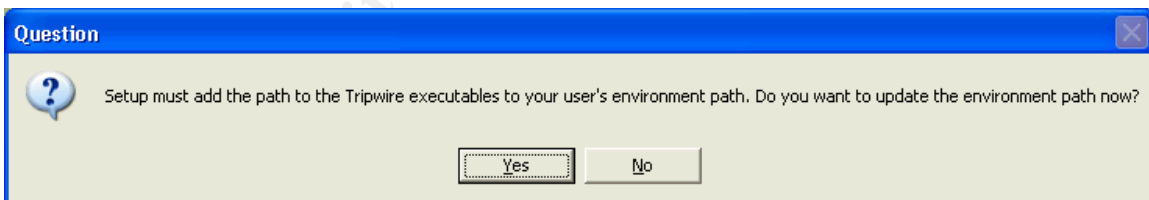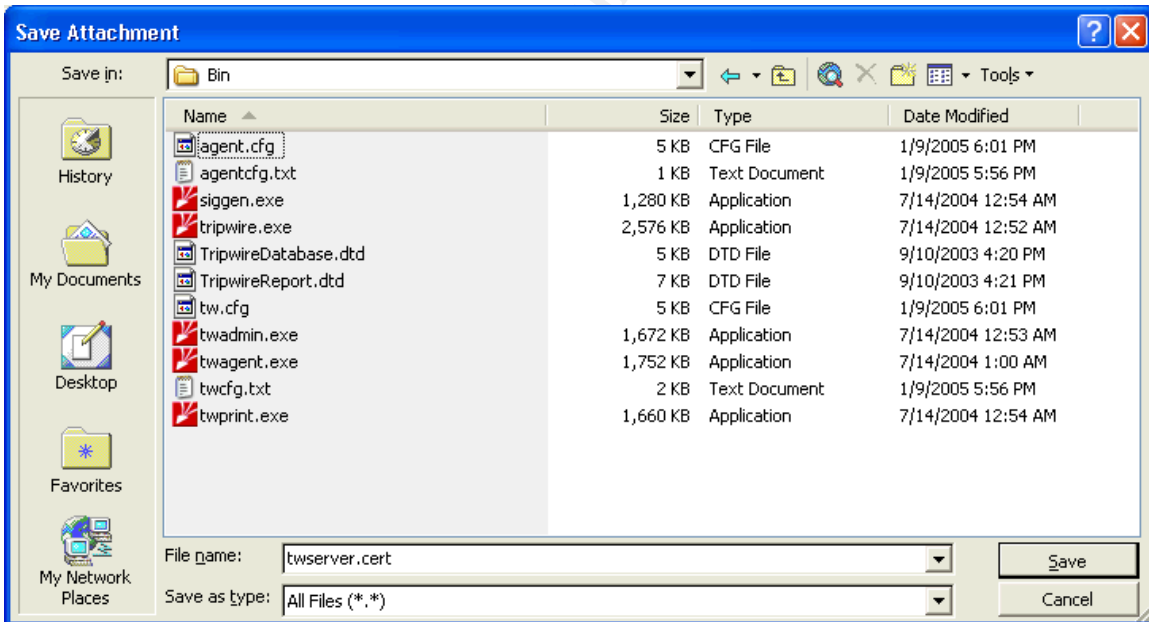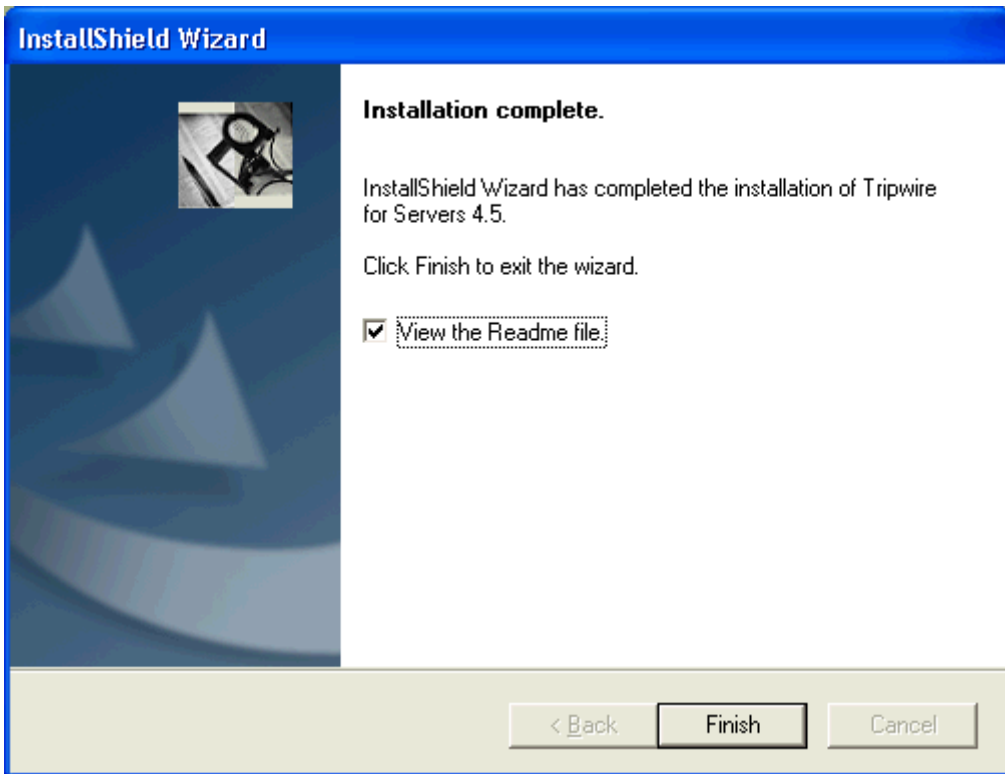
And (per instructions), copy the emailed license into the "Bin" directory. Note also the service properties (below).
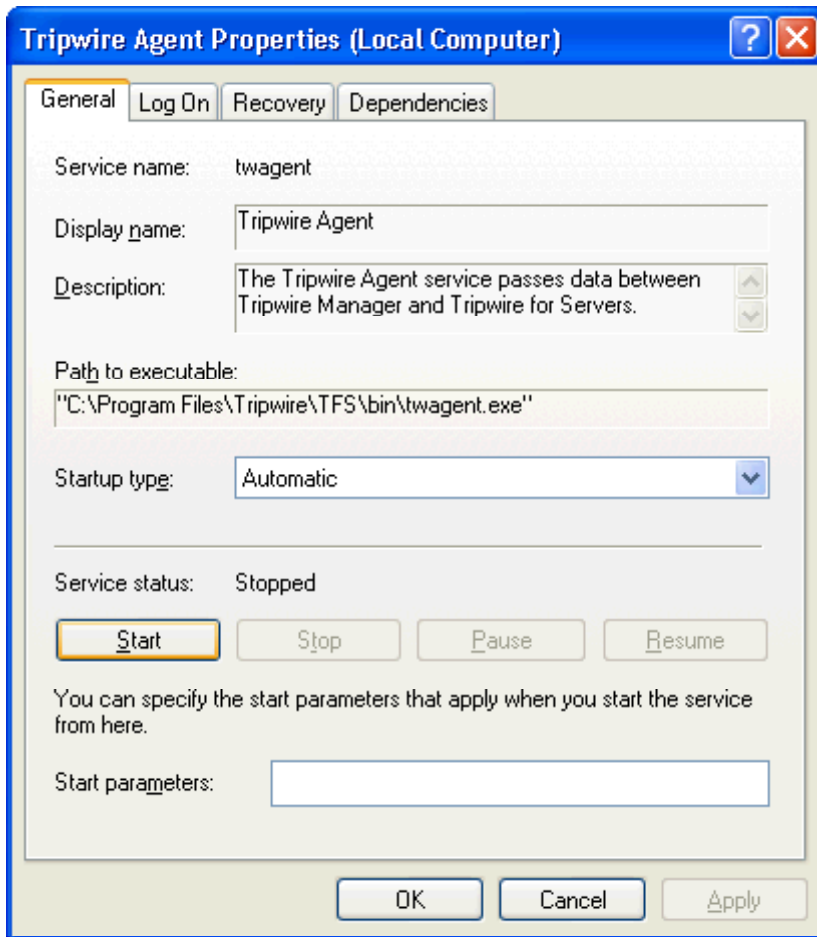
Connecting Machines to the Tripwire Manager

Unlike standalone, at this point the remaining configuration steps are done from the manager. From the (running) manager, select "Manager → Add Machines"
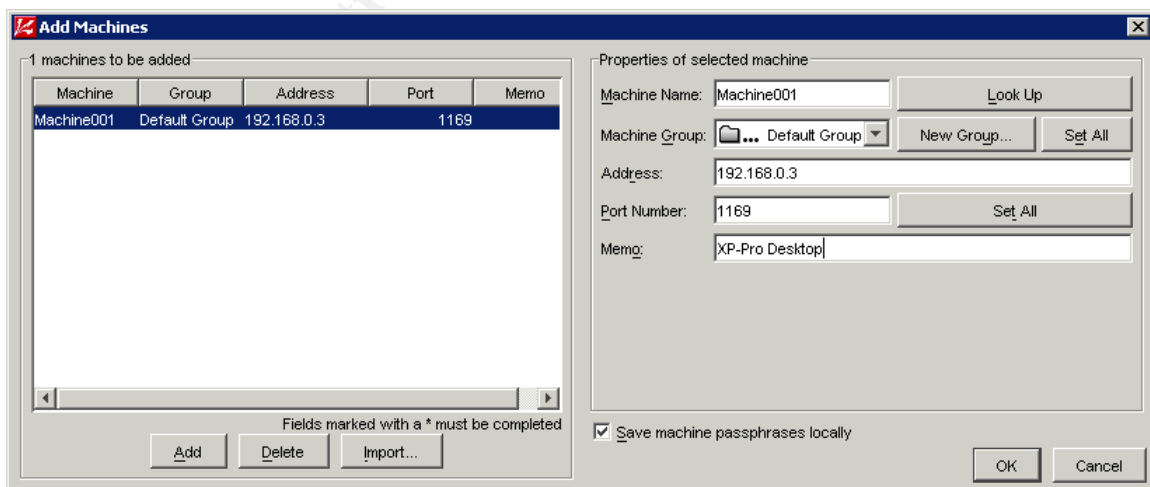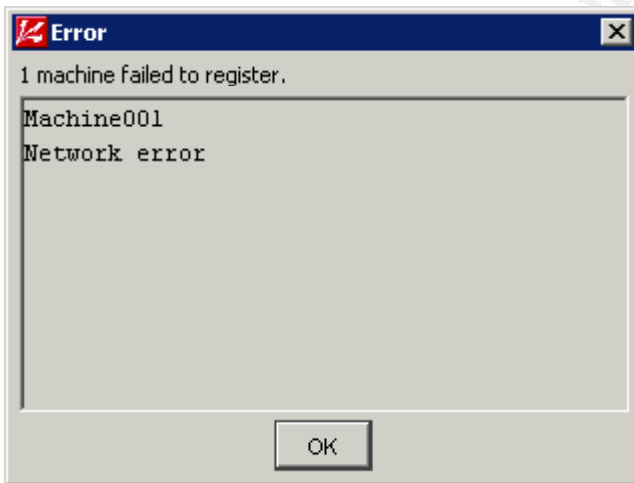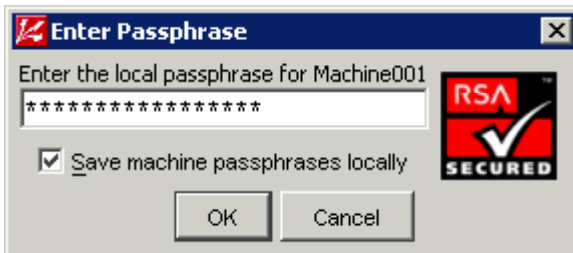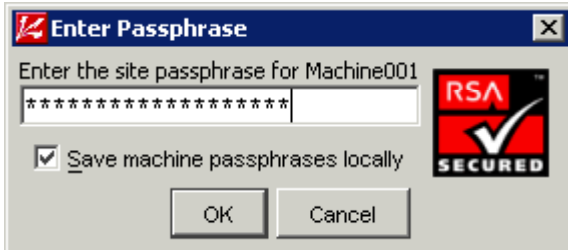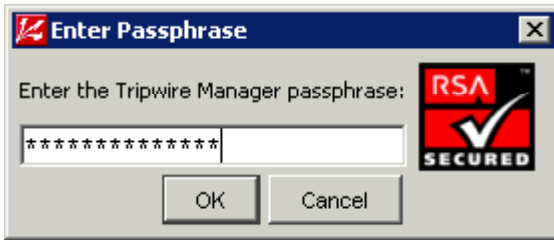
Network Error?!?! (Recall my note during installation when I specified the IP
address of the Tripwire Manager? This is the error it caused.) So, what does a
good network geek do? Check it out!!
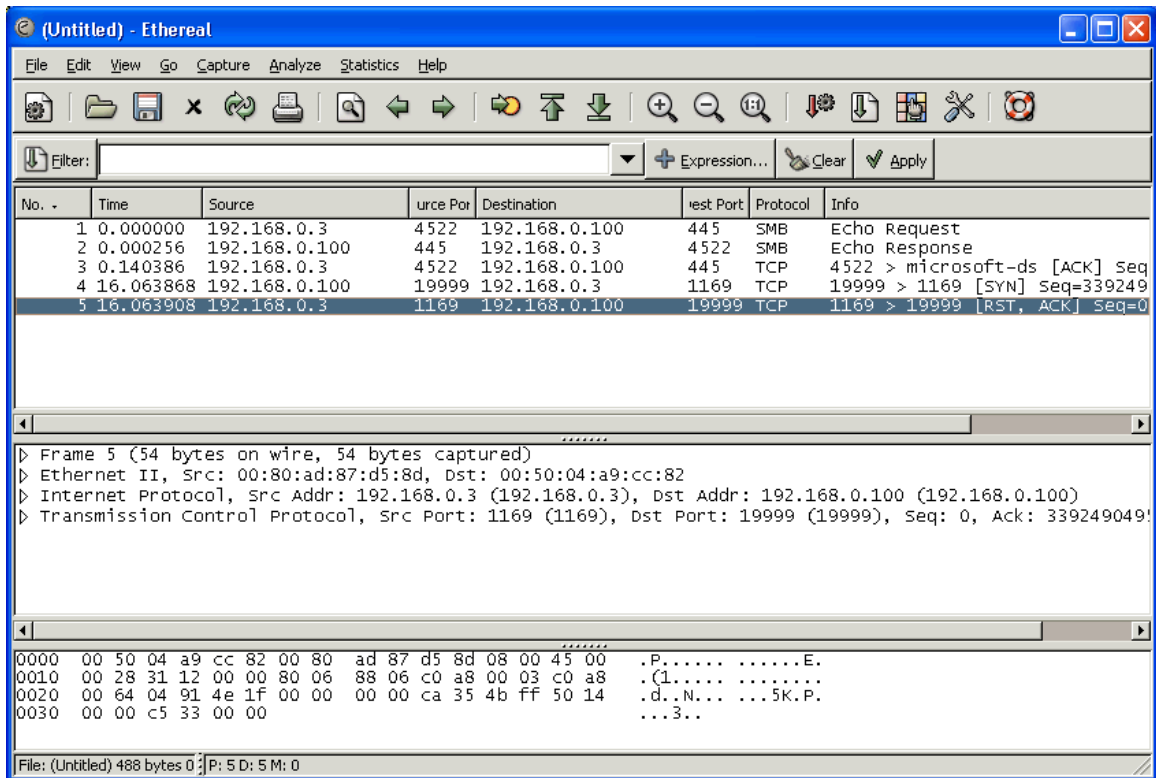
Hmmm…the tripwire manager can ping the server, but the server is simply sending a RST to the connection attempt.

This would indicate that the service is not listening. I checked the service again, and found it was starting, then immediately stopping. The Event Viewer said: "Tripwire Agent service terminated unexpectedly" and when I "googled" a bit, I found two references and a solution.[13]

"Uninstall the software via Add/Remove programs and reinstall, however if necessary choose a specific IP address that is valid for that machine. Selecting "Communicate with Tripwire Manager through the default IP address" will use the default of the server. "Communicate with Tripwire Manager through a specific IP address" may be useful if the server has 2 or more NIC's, and you need to select the correct one. Keep in mind, Tripwire is asking for the IP of the server, NOT the IP of the Tripwire Manager machine.

OR

Go to the Tripwire bin directory, and open the agentcfg.txt file with a standard text editor (ie. notepad). You may also use a text version of the current config file, extracted by using:

twagent -m f ../key/site.key agentcfg.txt (Windows)
./twagent -m f ../key/site.key agentcfg.txt (UNIX/Linux)

---

[13] http://www.tripwire.com/kb/view.cfm?aid=36 described the problem perfectly and included a link to the solution at: http://www.tripwire.com/kb/view.cfm?aid=12

Modify the last line (IPADDRESS=) to be a valid IP address for that machine, save and exit the text editor.
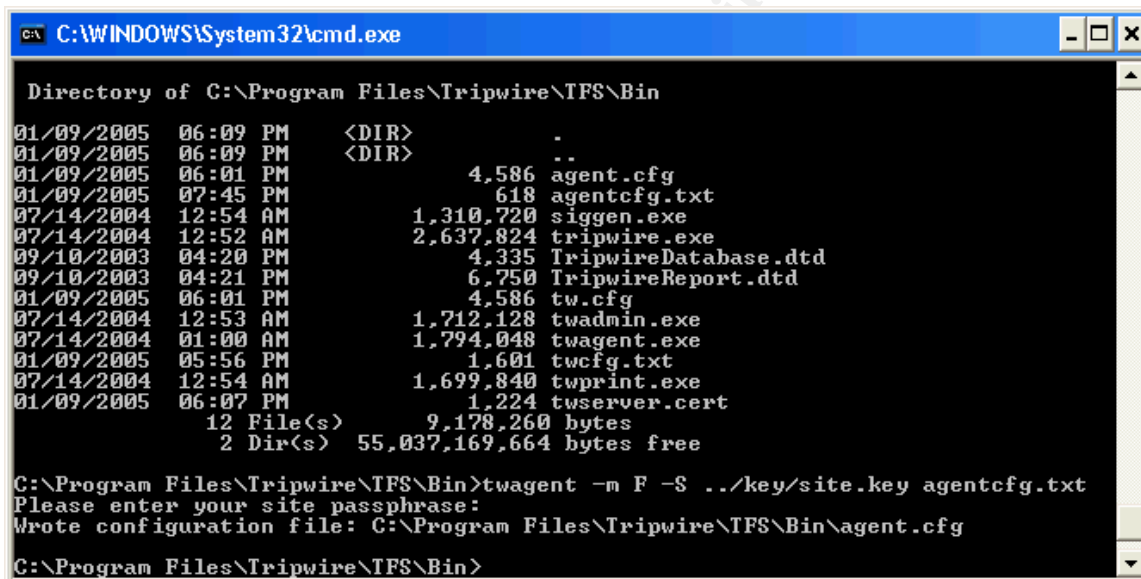
Open a command prompt and naviagte to the Tripwire bin directory, run the following command to create a new Tripwire agent configuration file.

twagent -m F -S ../key/site.key agentcfg.txt (Windows)
./twagent -m F -S ../key/site.key agentcfg.txt (UNIX/Linux)

You will be prompted to enter the site passphrase. Once you see the message the the agent configuration file has successfully been created run the following command from the bin directory start the Tripwire agent service.

net start twagent (Windows)
./rc.twagent start (UNIX/Linux)"

So I did what it said and…

then checked to make sure the service was running and the port was open:

Looked good on the Tripwire Server end, but I *still* got a network error from tripwire manager when adding the server…

Aha! I remembered I was running the XP firewall, (Note that the system will still report the service LISTENING even with the firewall blocking connections to it.)

I disabled the firewall and was then able to connect to TCP 1169 from another system. Now to mitigate the security problem of leaving the firewall off, you need to add the service, as follows:
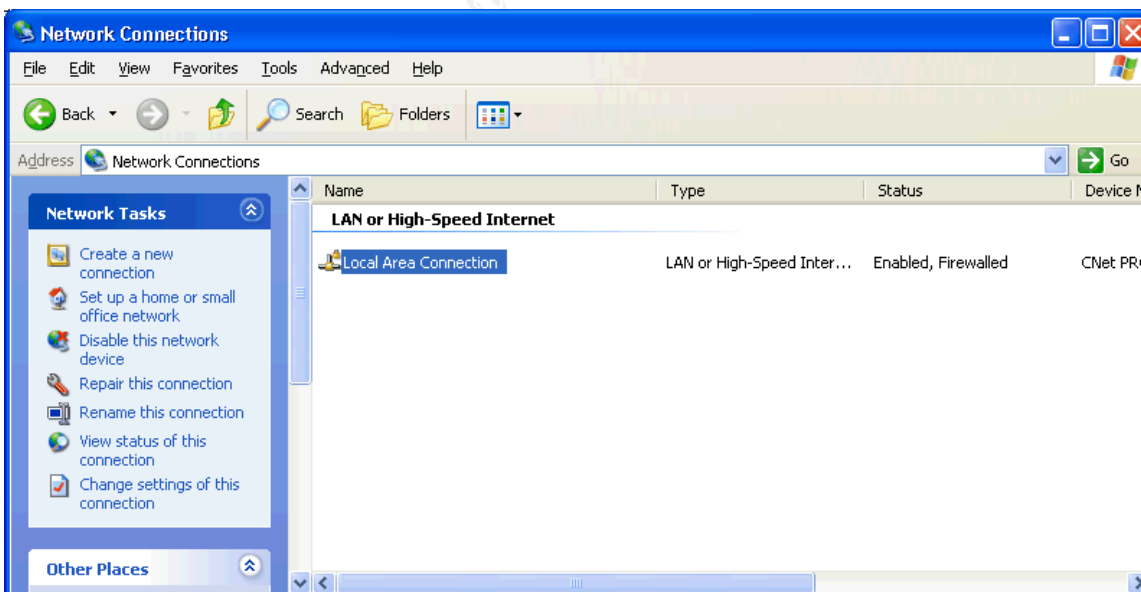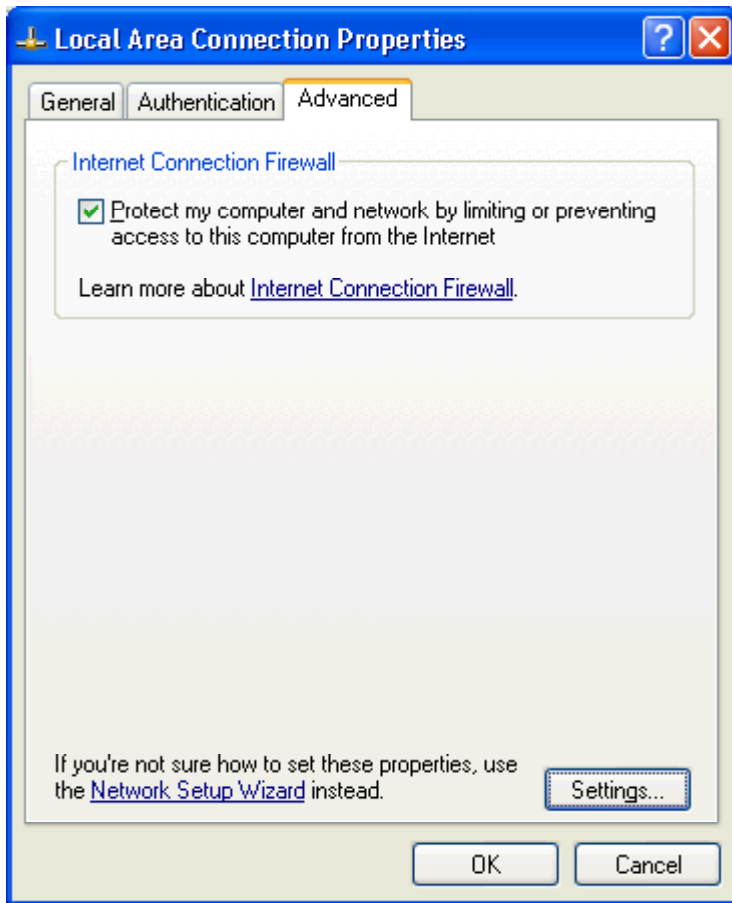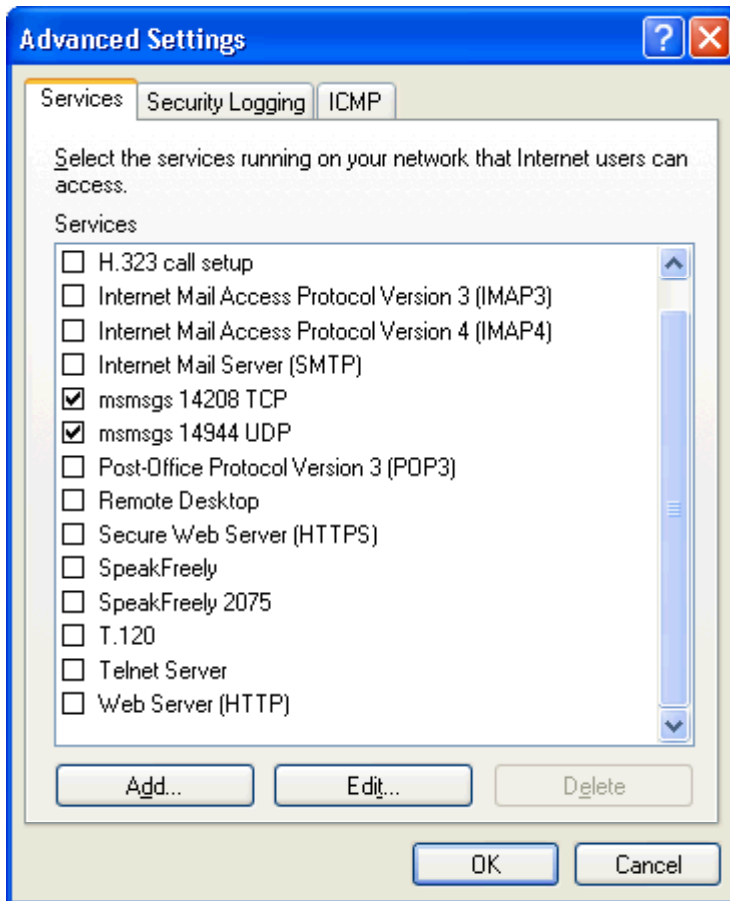
Select "Add" and enter the appropriate info.
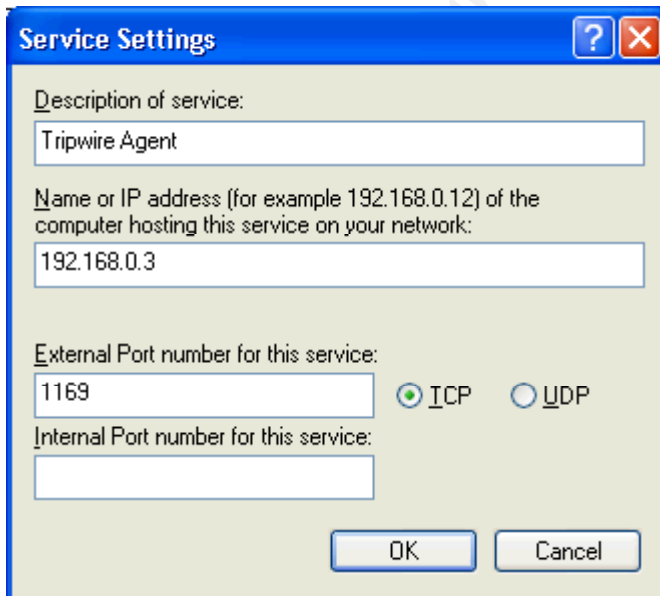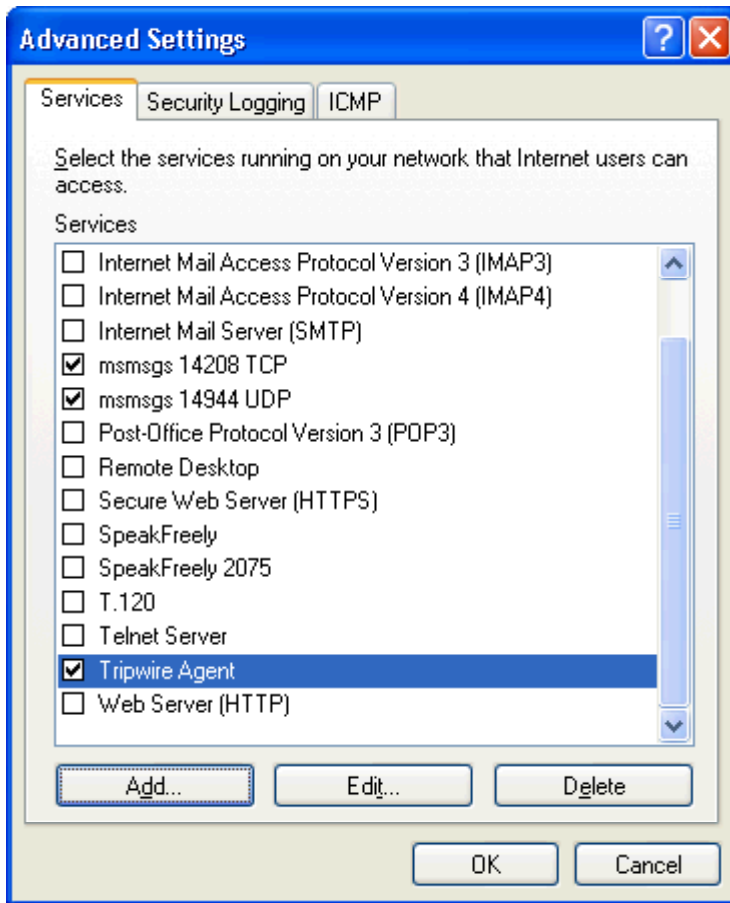
Note that this has now been added. OK, let's try adding the server to tripwire manager again.
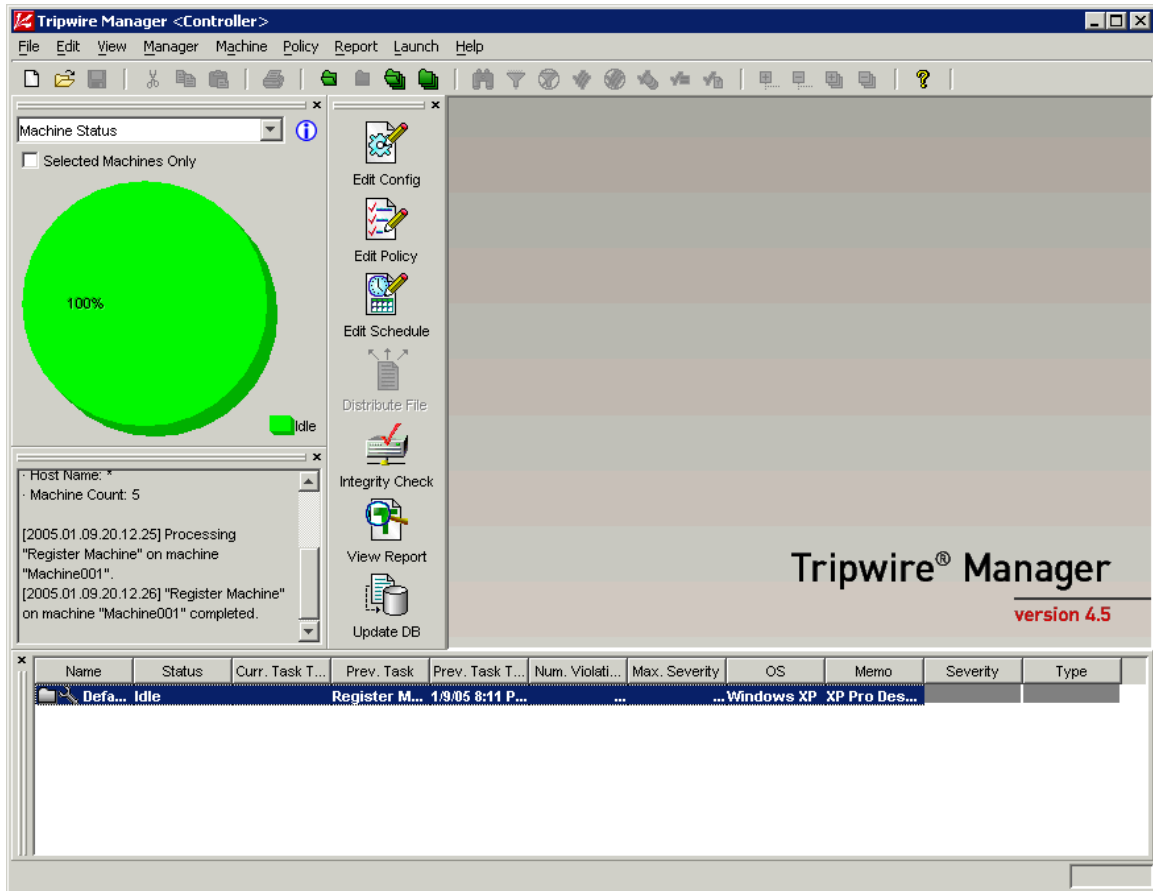
Success!

Please note that this problem will not be unique to the XP firewall. Any system firewall that blocks inbound requests will cause the same problem. This also includes servers where the Security Policy has been used to lock down the network (in the IP Security policy section)
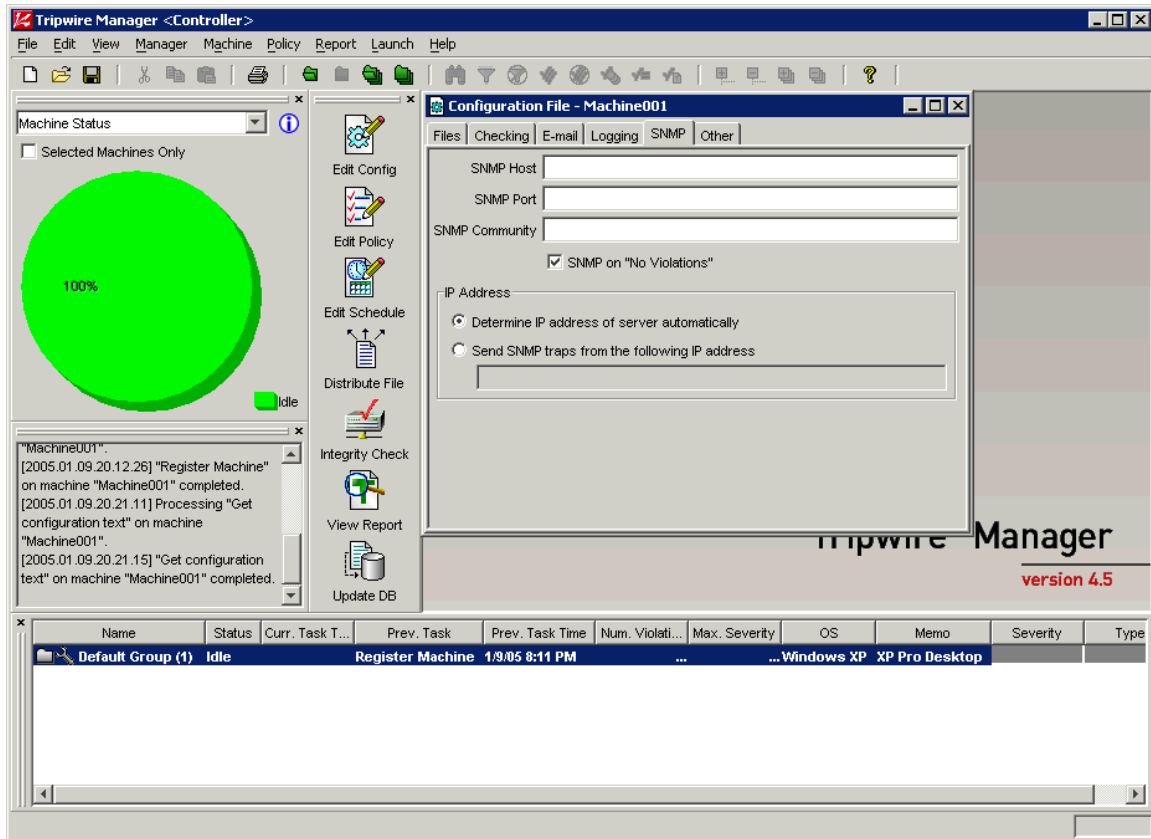
Once the Server has connected to the Manager (visible in the window and the "Machine Status" shows bright green) the Manager has the ability to control the settings on the server that were entered on installation (such as Logging to syslog or SNMP).

The next step is to create a policy file. However, unlike the standalone configuration, this is done through the manager now.
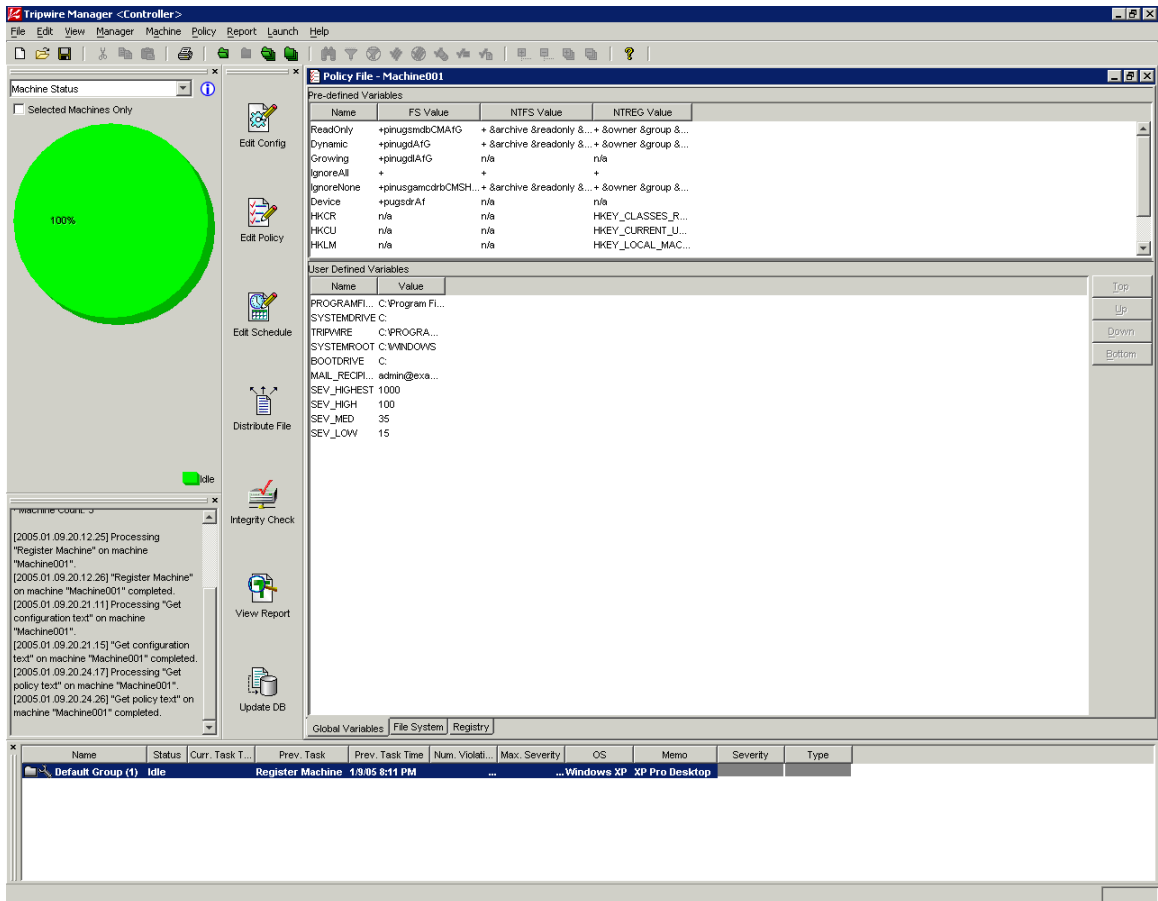
Selecting "Edit Policy" opens the graphical policy file editor window with tabs for Global Variables (shown),
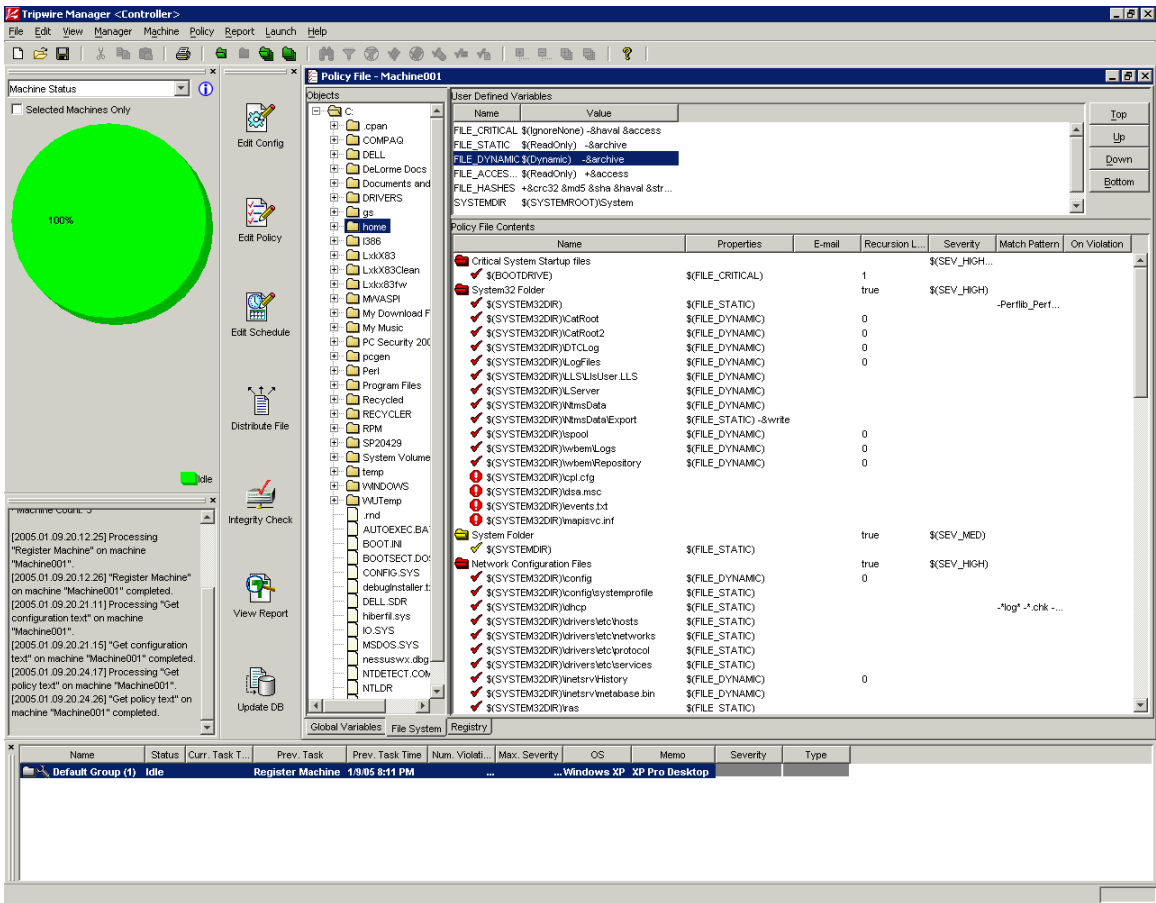
File System
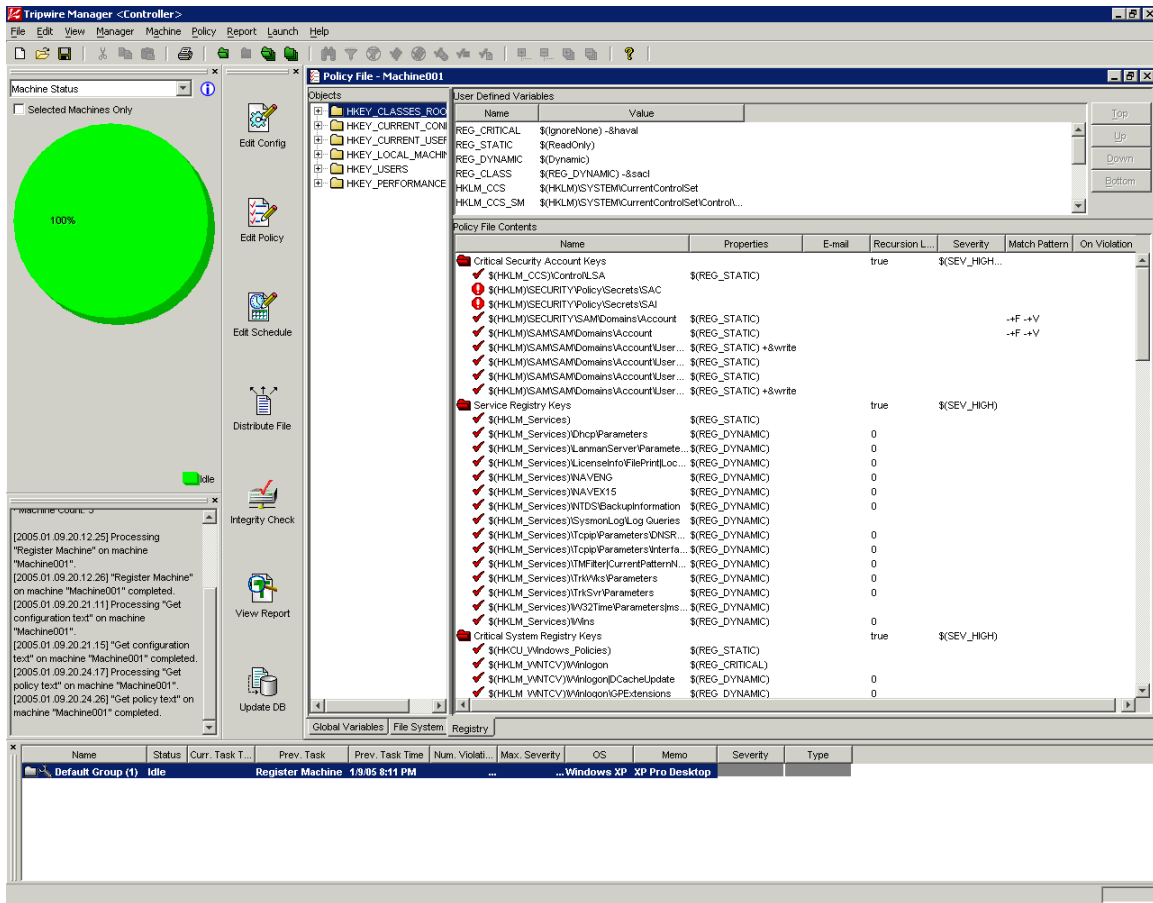
And individual registry settings!!!

Note that even though the manager provides a graphical interface to edit the policy, it *does not* make it completely intuitive to do so. I recommend that you begin with the default to get the feel of it, and that is what I did here.

Once satisfied with the policy you select "Machines, Initialize Database":
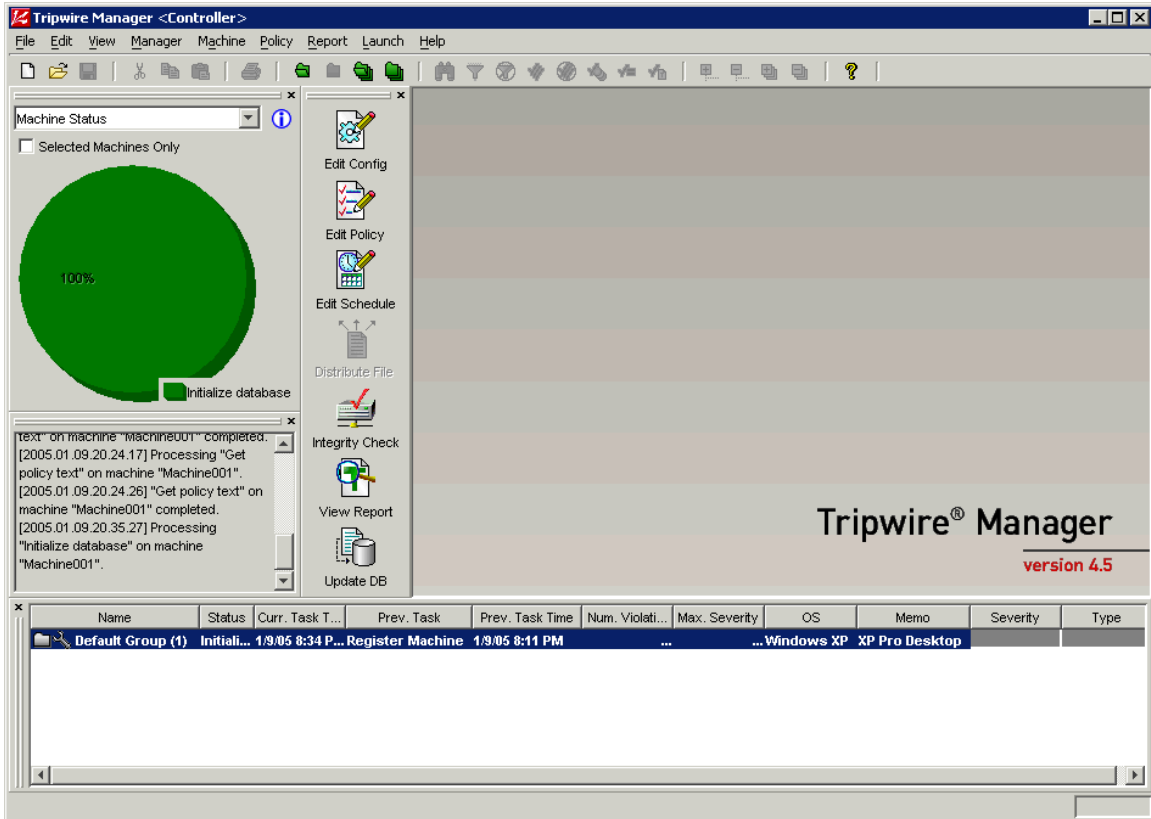
The "Machine Status" immediately shows "Initializing Database" and when it finishes, the icon changes back to bright green with the summary output in the text window.
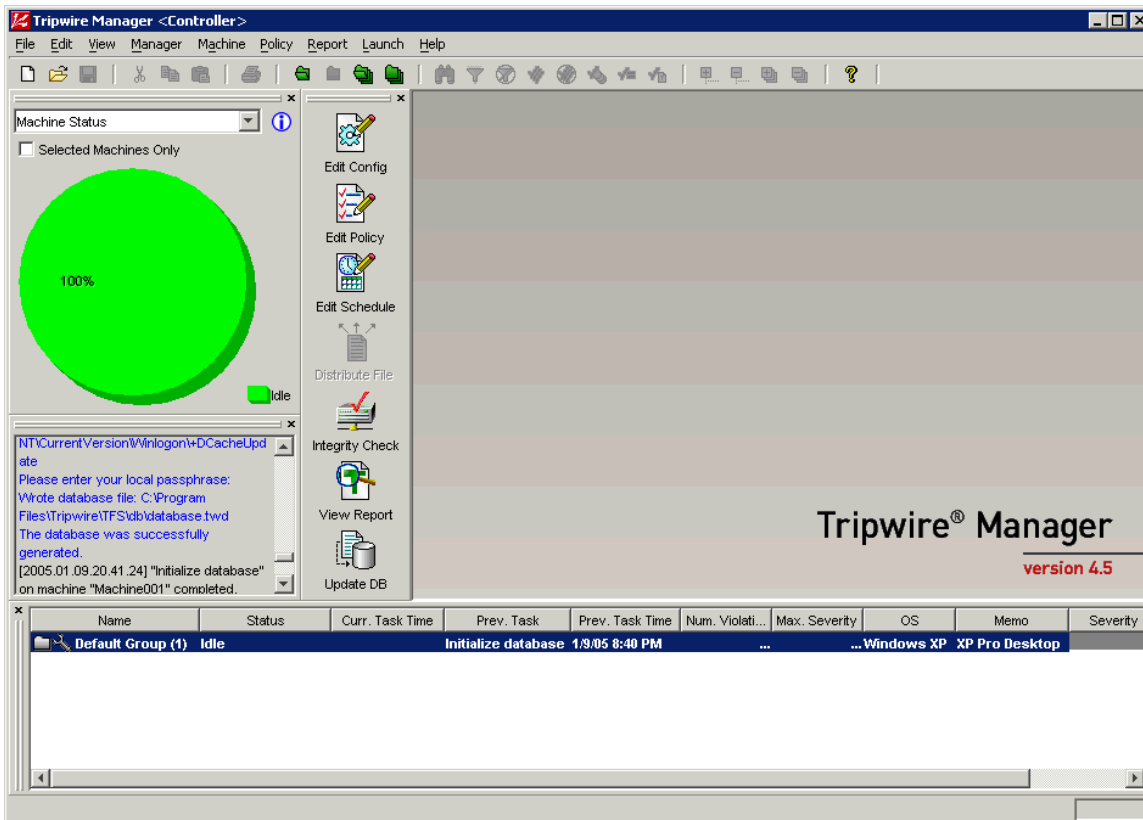
I then installed some software and ran an integrity scan . Once it completed, it
showed (lower right) red and yellow bars indicating various severity changes:
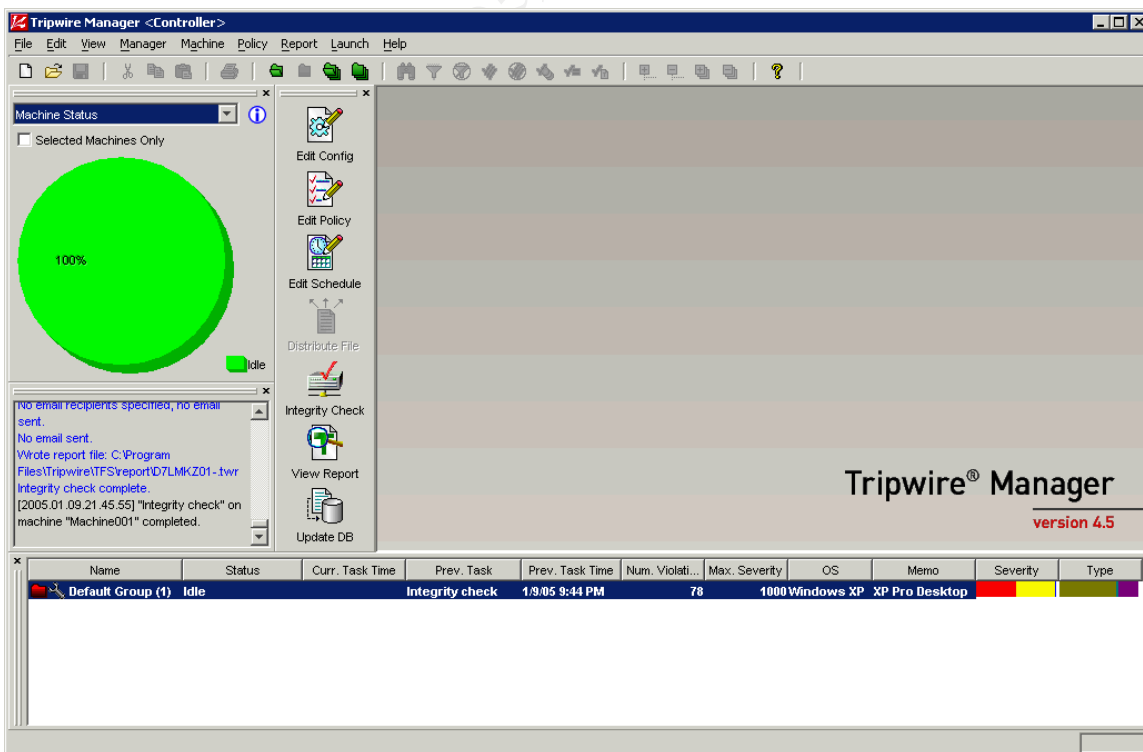
Note that when the scan completed, the "status" showed bright green, but the

"severity" and "type" now indicate possible problems. This is shown by selecting the various drop downs for the "pie chart":
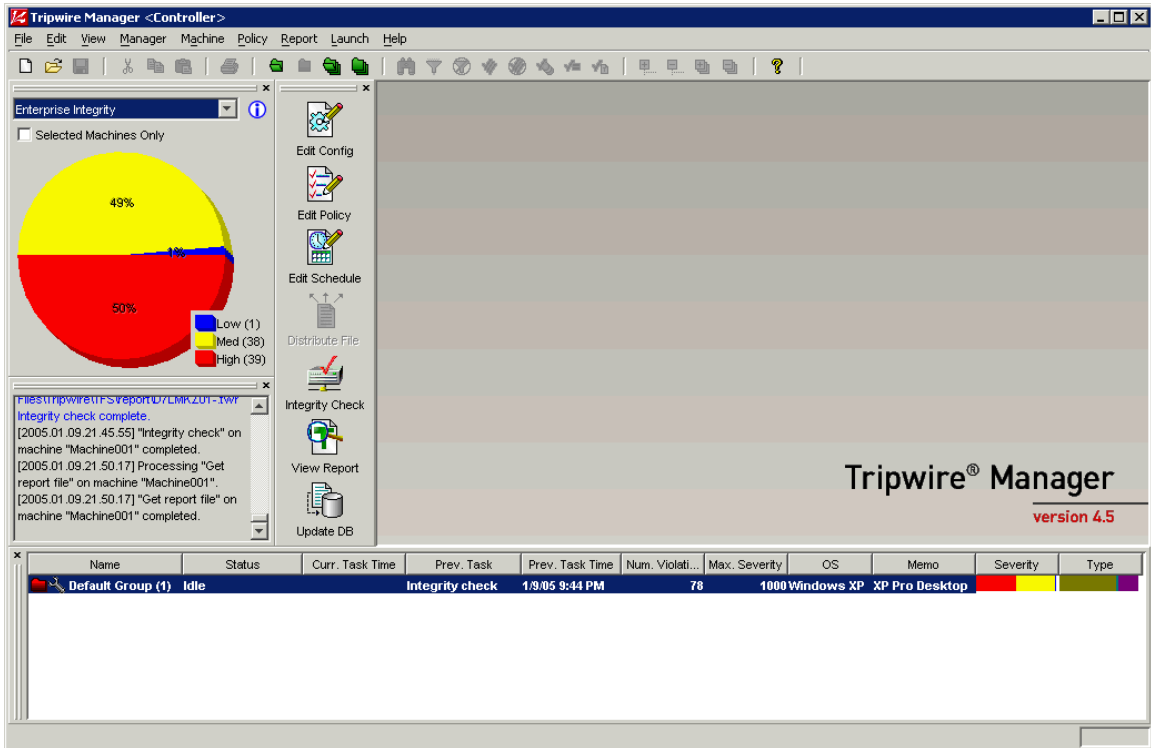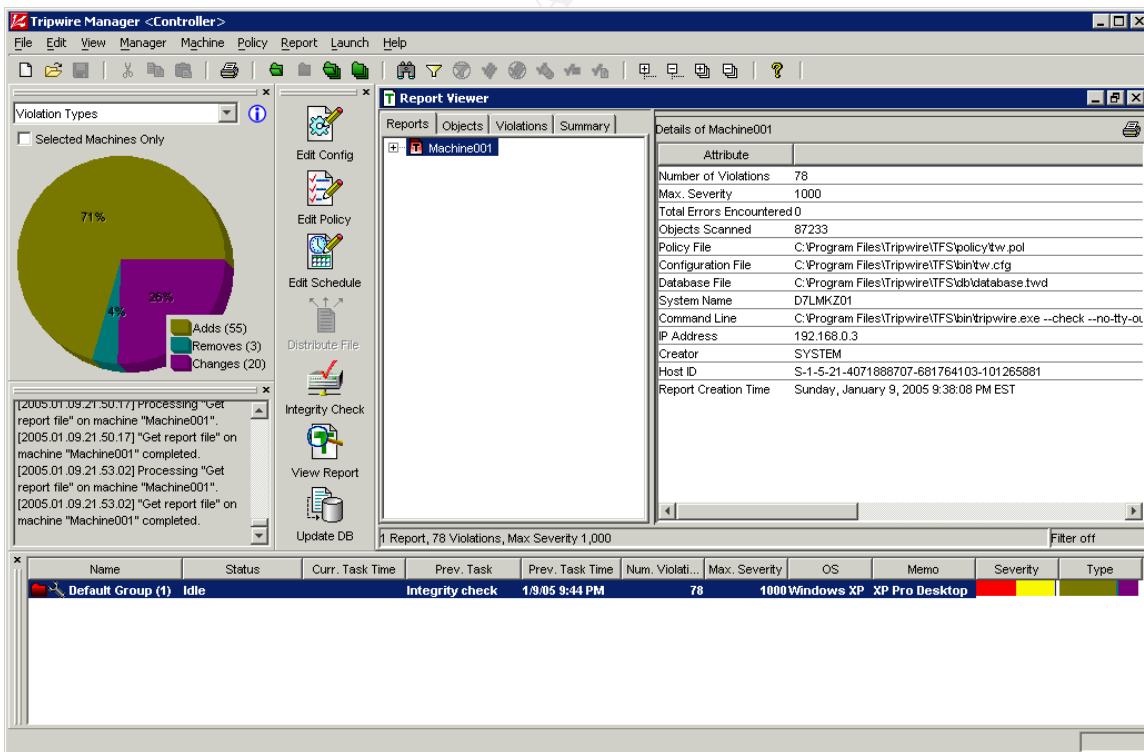

Enterprise integrity
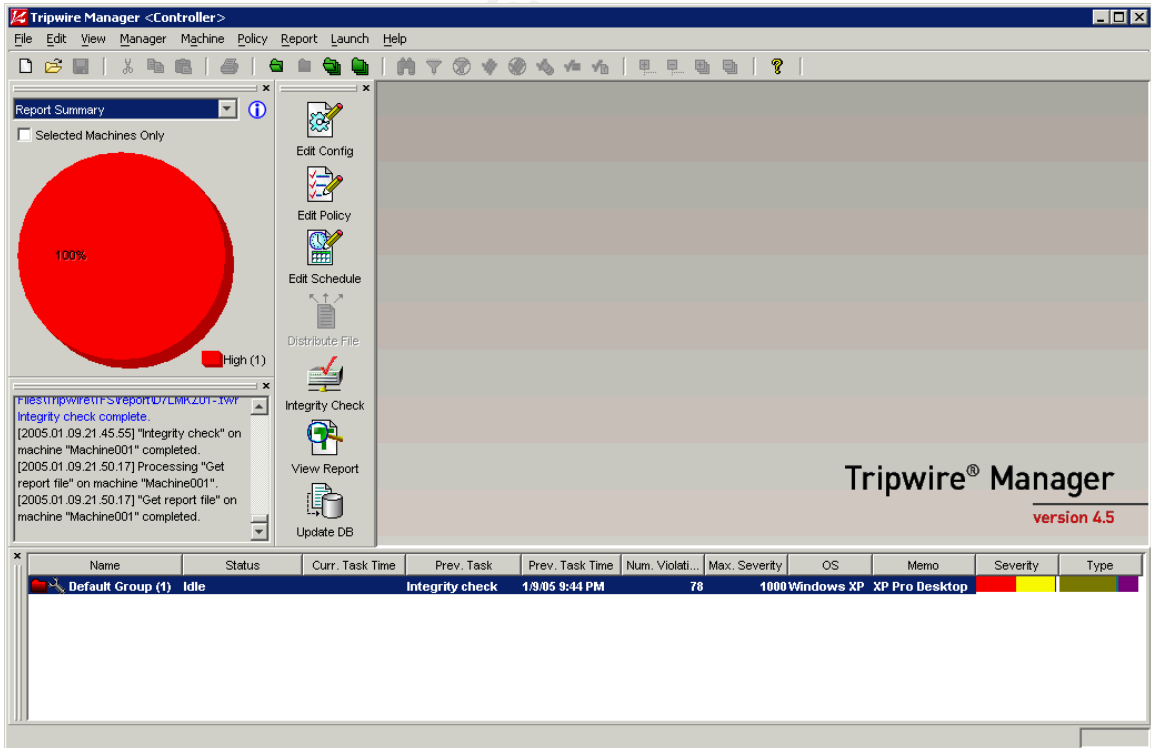
Report summary (for that machine)

and violation types (below)
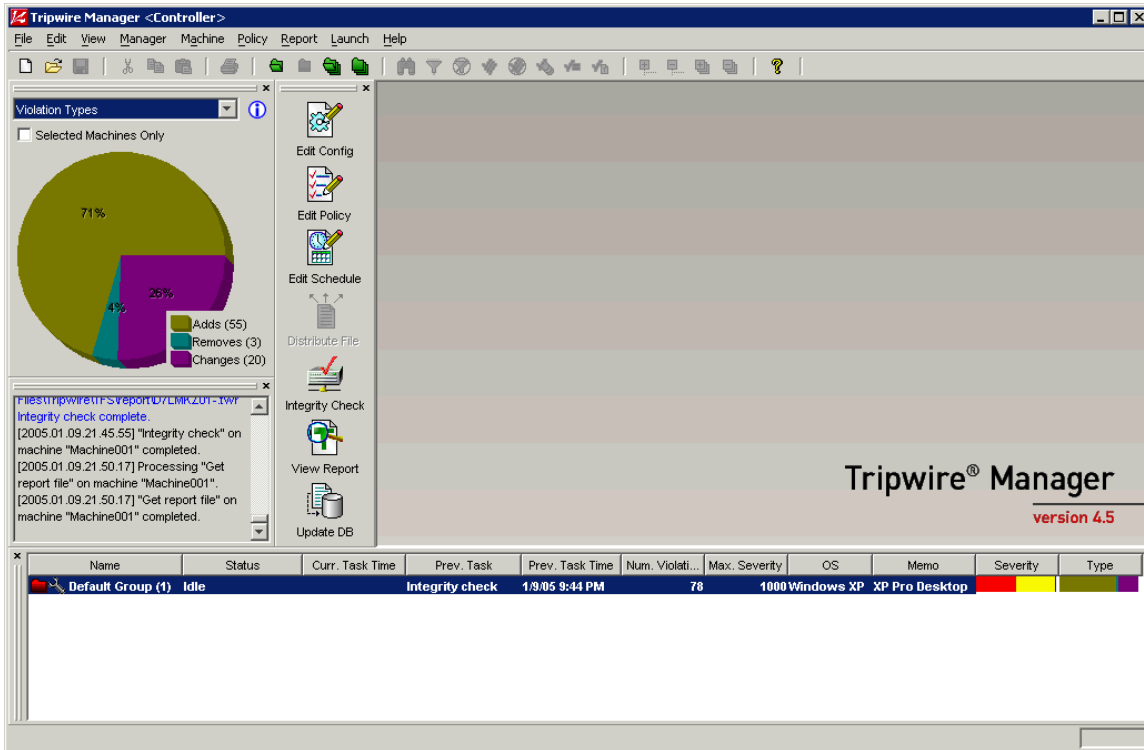
Choosing the "View report" icon opens the details in the main window:


Here you can drill down in various areas including Report (shown collapsed here since this is an enterprise tool, and only one machine shows violations at this

point), Objects, Violations, and a Summary tab also are available. The number of items available are huge, and I cannot include screen short displaying even a small sample. What I will do is run a more "enterprise" test using the additional machines once they are installed.

Tuning Policy:

Adding files to be monitored

Before proceeding to resolve the violations discovered on my desktop, I want to demonstrate how to customize the policy file for a different type of system. After installing TFS on my (W2K3) IIS server and adding it to the manager, I looked at the default policy file (Edit Policy) and discovered that there were critical directories that were not included. This server runs a PHP bulletin board, and neither the PHP directory, nor any part of the Inetpub area were covered. How did I know this? You can see this by opening the policy file, navigating to the directory "Inetpub" a "right click" gave the options: "Add Rule", "Add Exclusion" and "Find Elements". Selecting "Find Elements" brought up this window saying there were no rules or exclusions that covered C:\Inetpub.
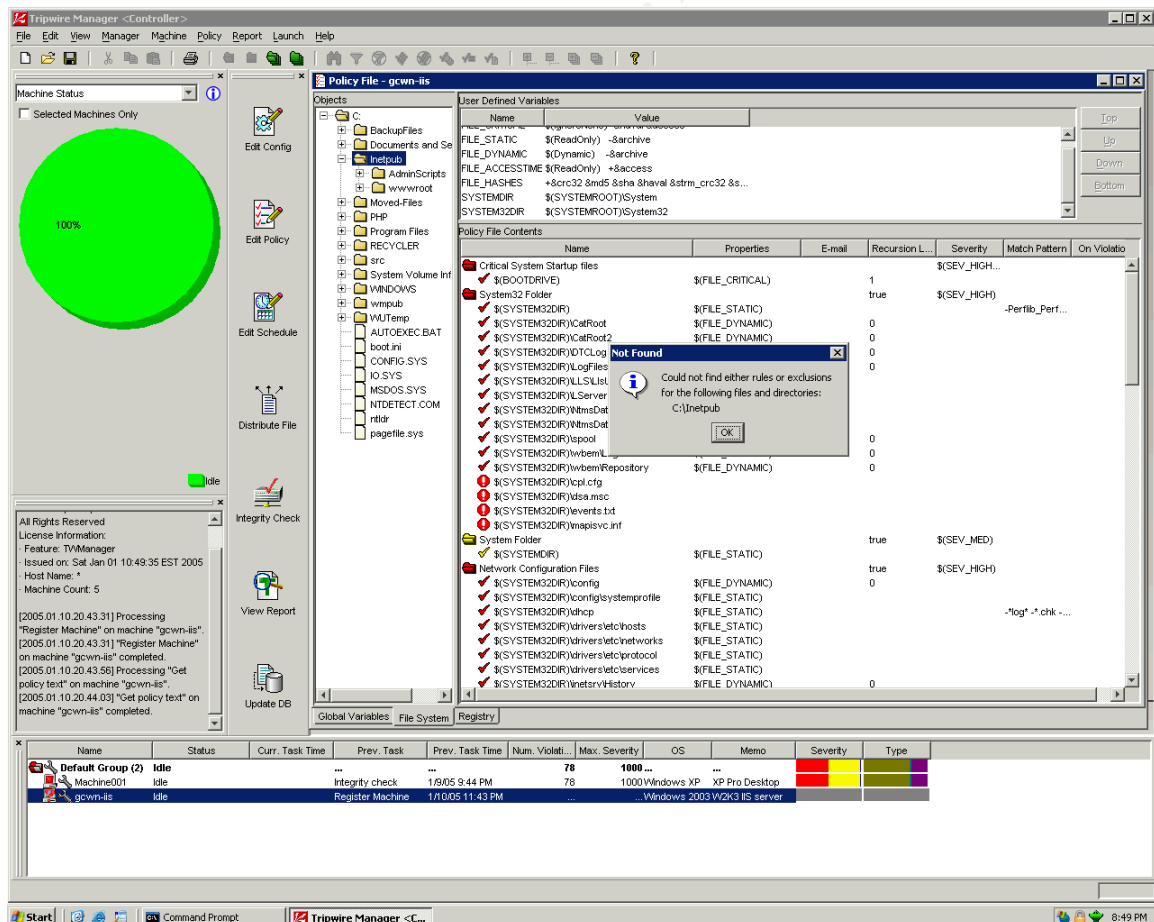
Let's see if we can fix both that and the (also not monitored) PHP area.
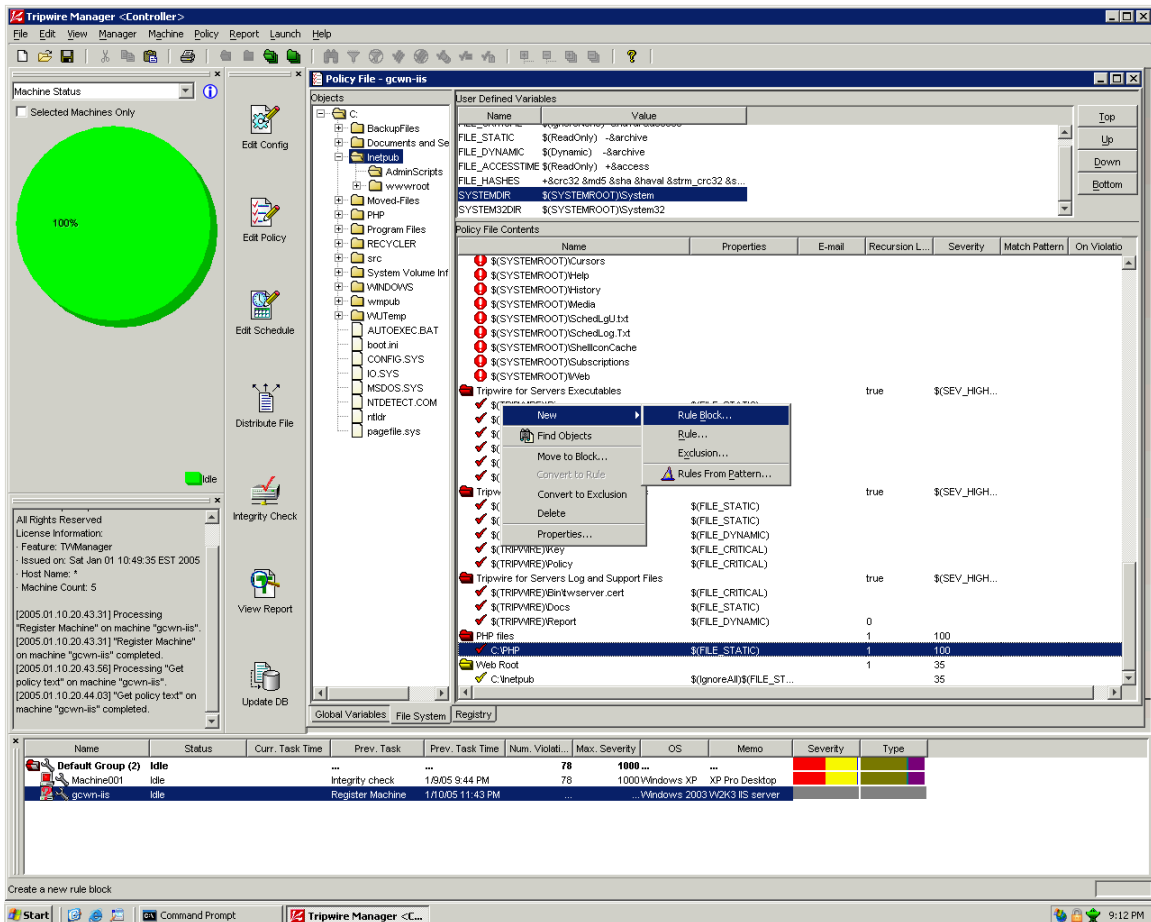
Here you can see, I have already created two new "Rule Blocks". The first is a "Critical" one (hence the red color) "PHP files" containing the object "C:\PHP" and the second (less critical, hence the yellow) "Web Root" containing the object "C:\Inetpub". The screen shot shows that right-clicking in the rules pane allows you to create new Rule Blocks and then within those, new "Rules". Here are the "Properties" of C:\Inetpub:
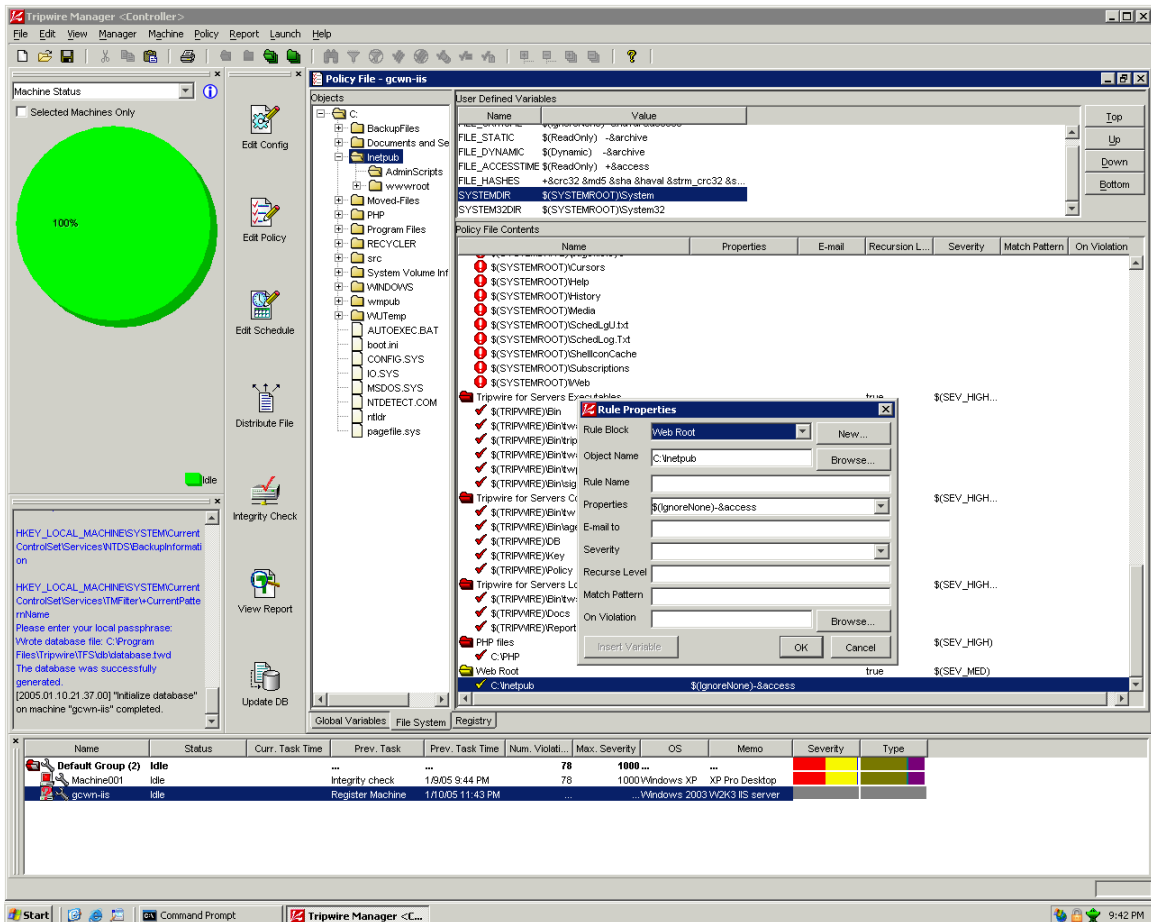
Once the "Rule Block" was created, I made a "New Rule" and with the window shown was able to "Browse" to find C:\Inetpub, use the drop-down list to select "$(Ignore None)-&access" for the "Properties". The attributes of "$(SEV_MED)" and recurse "true" are inherited from the rule block.

Note that for a web area, this will be much too scrict (especially since it hosts a bulletin board, but I will use this to demonstrate tuning the policy file.

Had I wanted to make this more general, I could have defined some user variables for where the "PHP files" objects were, so that the same rule could be used on another IIS server that might have them somewhere other than "C:\PHP", or that had multiple "Web Root" areas for a server hosting different sites.

I then initialized the database for the entire enterprise.
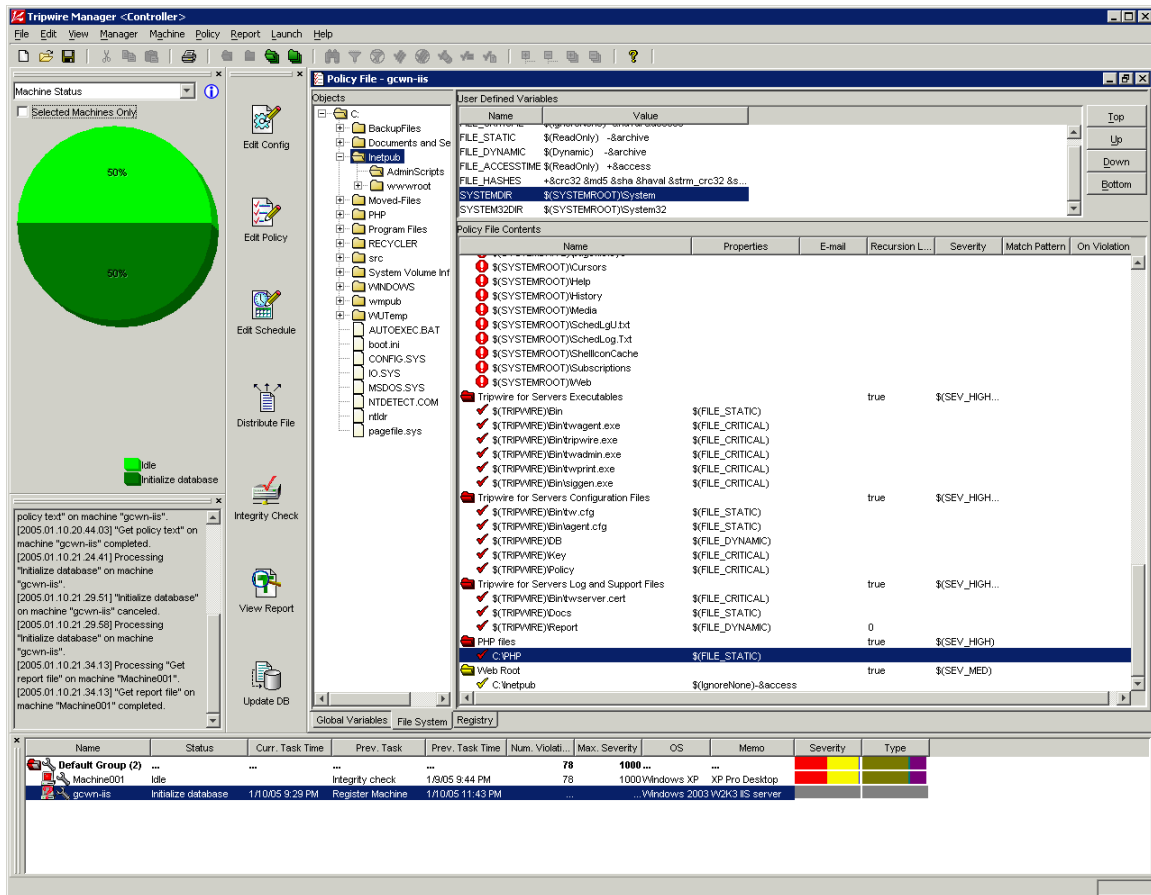
Notice the "Machine Status" indicator shows that 50% of the "enterprise" machines are idle, and 50% are doing "Initialize Database". The TWM window allows you to do multiple things concurrently, so you can be editing policy, initializing databases, running integrity checks, reviewing reports, and approving changes all at the same time. All the windows in use will be within the main "work area" of the app, able to be maximized, minimized, or resized as needed.

Once the initialization is complete, I ran an integrity scan (within minutes of initialization, and no-one accessing the web site). It should be clean.
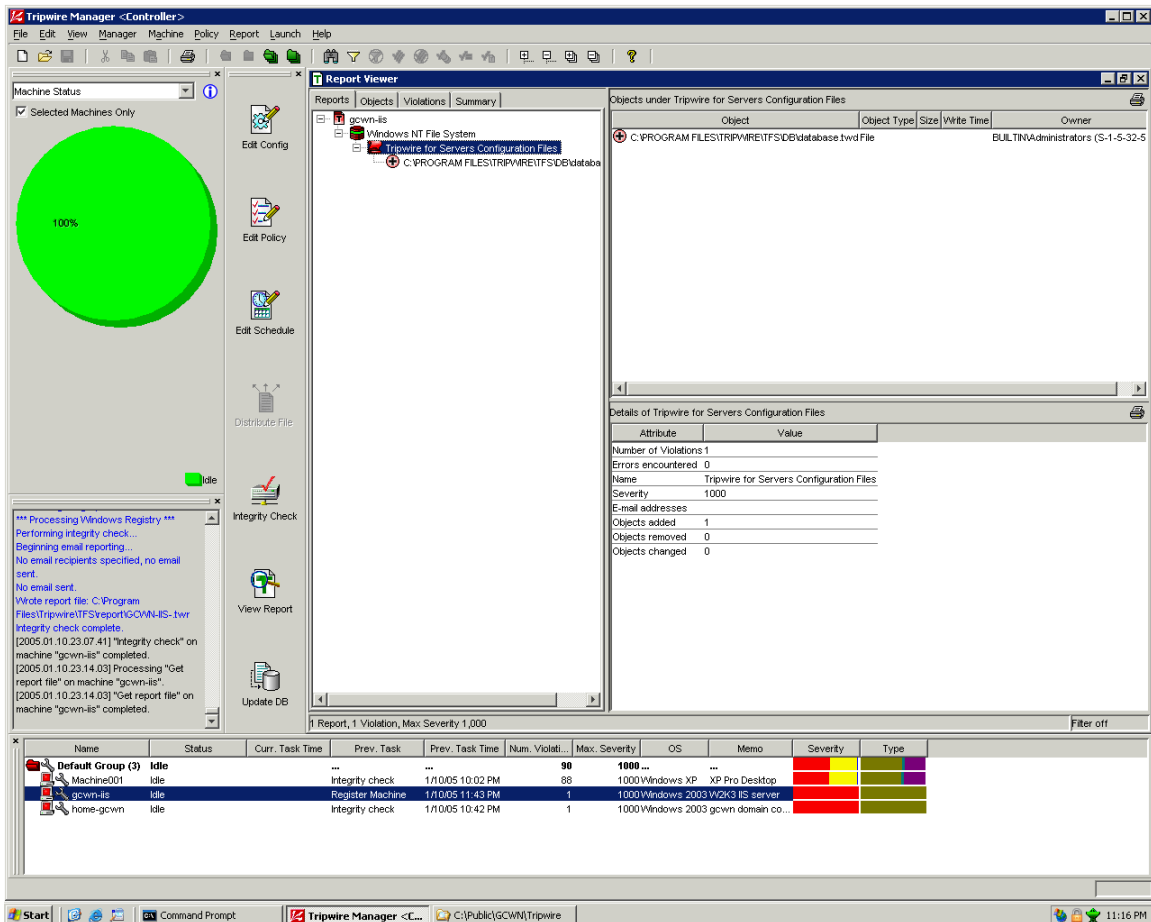
OK, what's that? One violation? Ahh, it's the fact that a new file was added. The "database.twb" indicating that a database file was created when the integrity scan was run.

Now, as this shows, I have a number of "violations" identified now on the two W2K3 servers, and the XP desktop. The next section of this paper will describe how to go about "tuning" tripwire to your system.

Dealing with violations
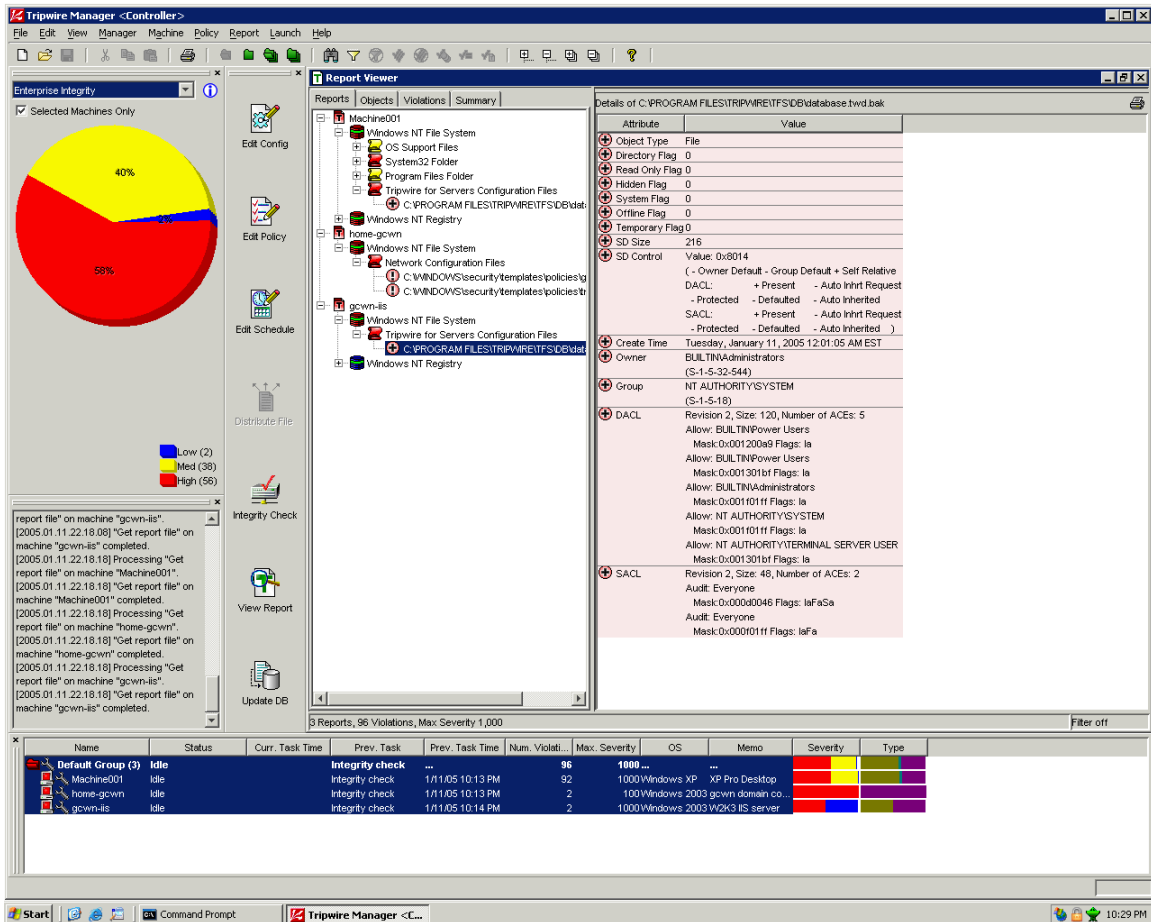
With the "Enterprise Integrity" pie chart showing a number of high priority changes, it might appear that we are in bad shape. But let's look a bit further.

Selecting "Violation types" for the pie chart, and selecting the machine with the large number of violations we find:
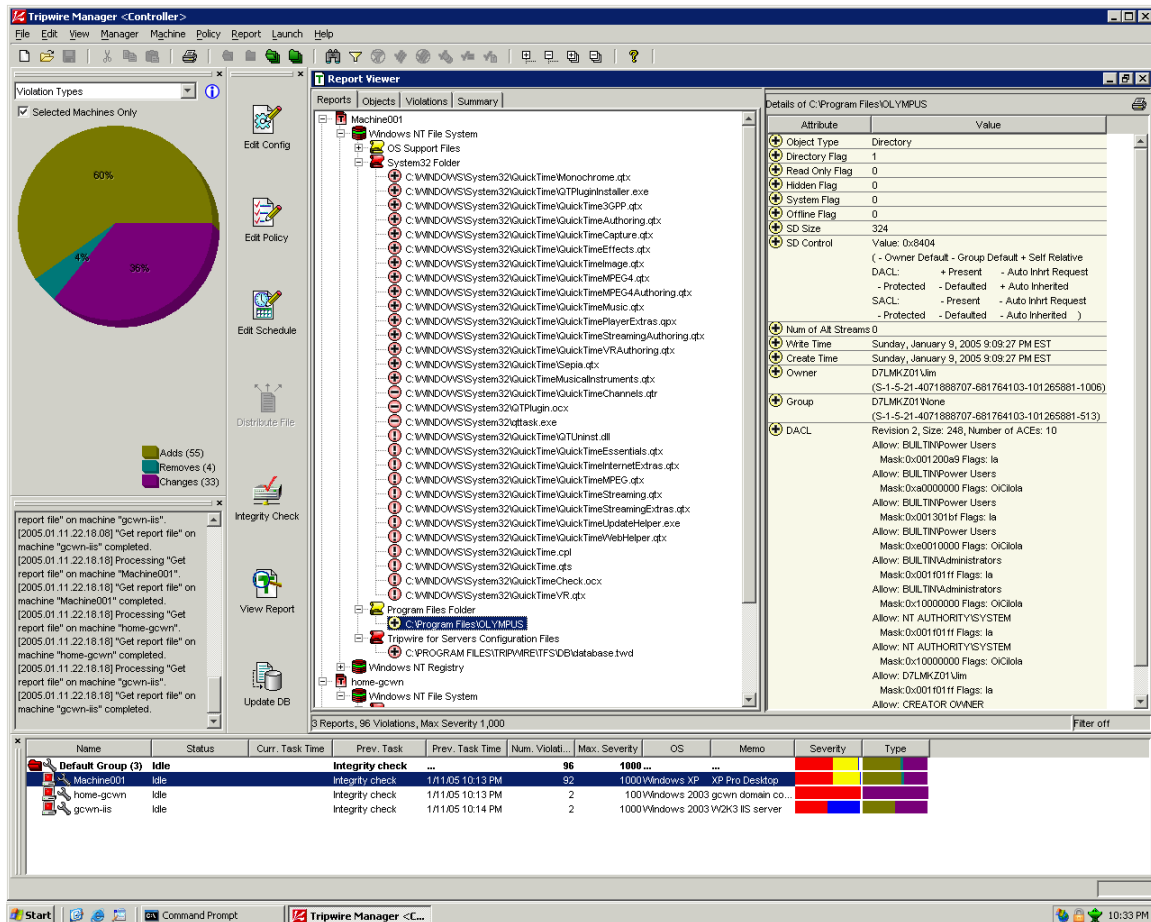
Most of these are added files. And the right hand pane shows that many of them are part of the "QuickTime" package, with one new folder (in yellow) named "C:\Program Files\OLYMPUS\" I happen to know that the "owner" of that system has just purchased a digital camera, and in fact, this report shows the result of my installing that software.

So, let's clean it up. Following the procedures in the documentation, the process is straightforward:

1. Run an integrity check.

2.   Examine the report, editing the policy rules to eliminate "false positives". You can do this within the Tripwire Manager by:

    a.   opening both the report and the policy file at the same time.

    b.   Go through the violations one by one and "right click" on the flagged file or registry value, select "Find rule in policy". (This is shown below)

    c.   Decide if you simply need to accept the change (as in this case), or whether the rule itself needs to be modified, or if you simply need to "exclude" the offending object from further scans.

3.   Update the policy file, rinse and repeat.
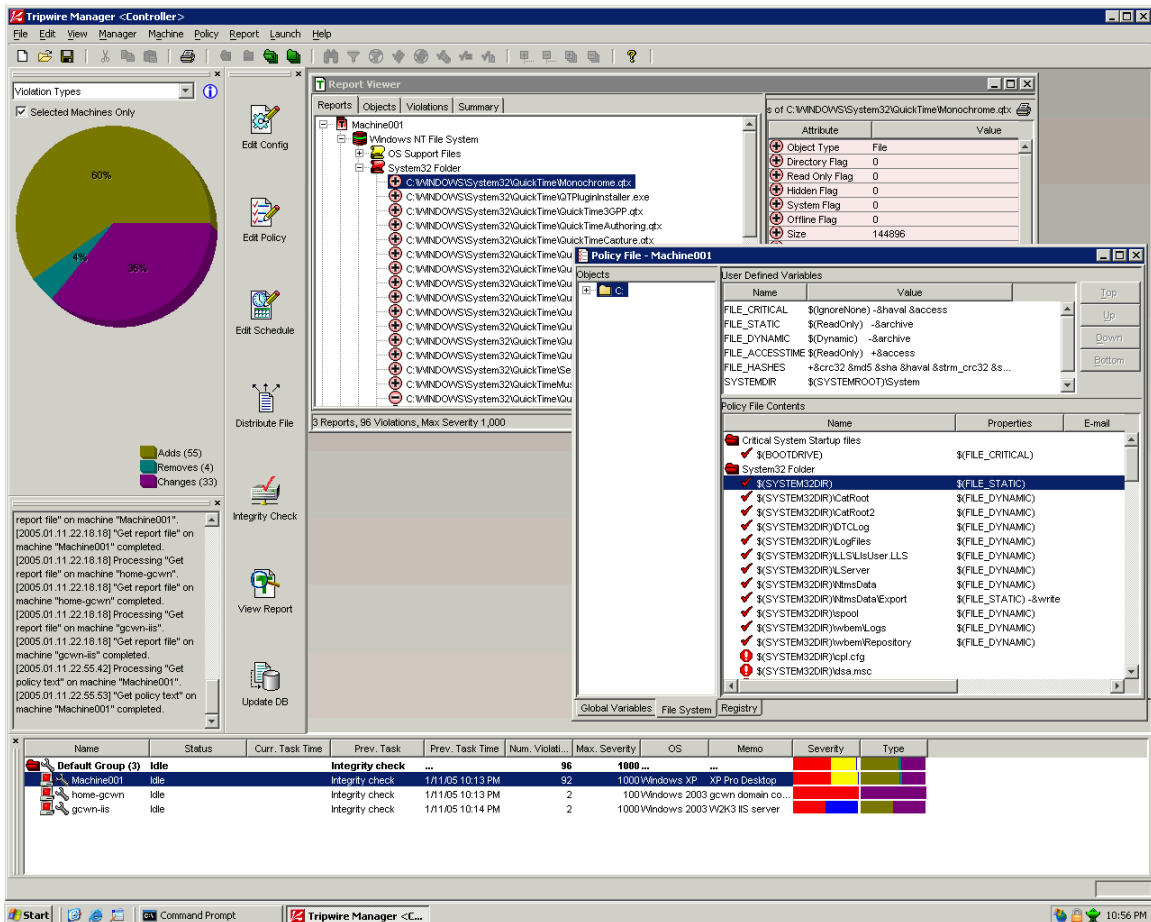
In this case, since I know I installed new software, before I mess with the policy, I will simply update the database with the new files "accepted".
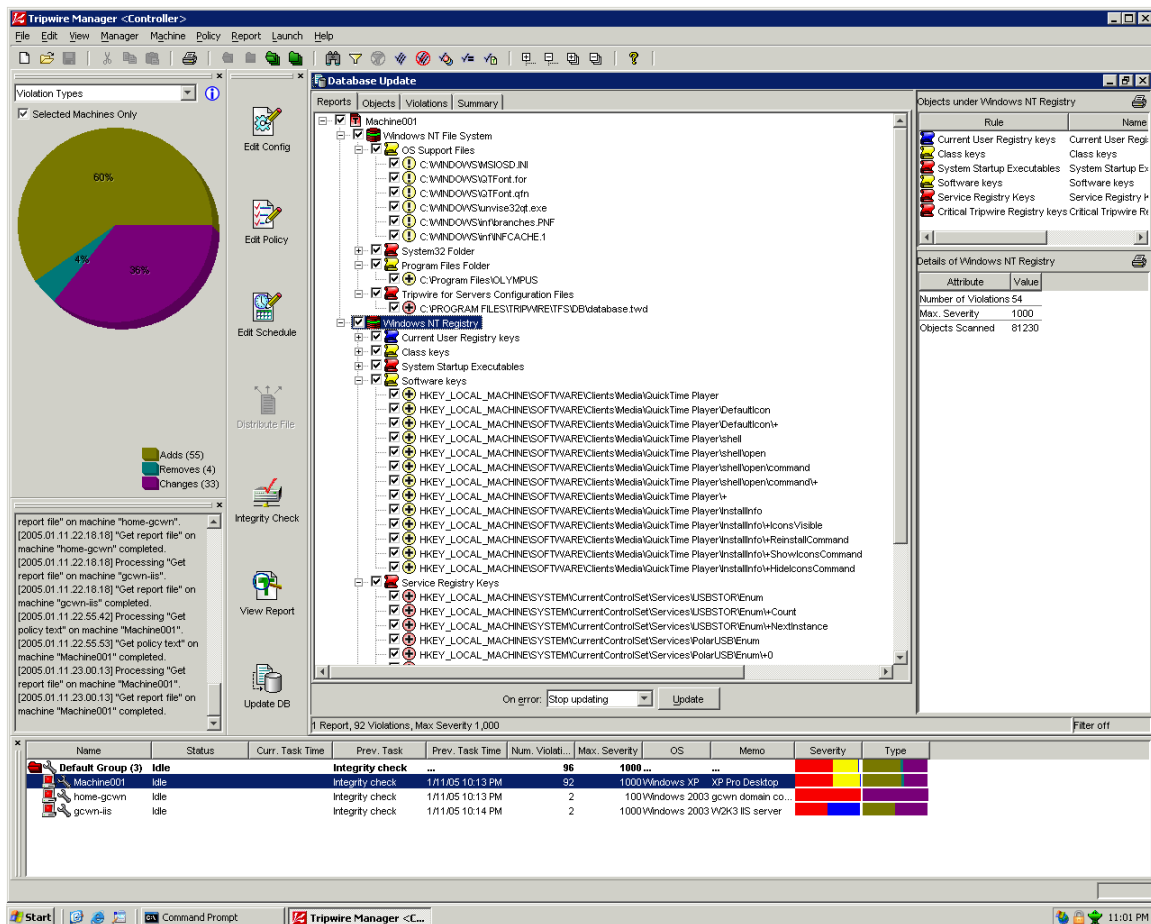
Here you can see the list of violations, and choose which ones you should accept. Once doing this, click "Update" and allow tripwire to re-generate a new database. Once this is finished, run another integrity check and review the report again.

This process will take time. Please be patient. Again, you are probably not going to know what changes you should be concerned about and what ones you shouldn't for some time. This is a learning process for you about your systems. I would recommend that anyone considering using this (or any other) host based intrusion detection software plan on dedicating time each day for several weeks simply "playing" with it. Try several systems. Run some with no changes, install or remove software on others. Apply patches. And tune, tune, tune the policies. And when you think you have it fairly clean, reboot the systems and run it again. (You will probably be surprised at how many things change on a normal reboot).

This process of tuning, re-running integrity scans, monitoring, adding software, patching, re-tuning continues essentially forever, although after a time, you will find that the alerts are fewer and more meaningful and it will take less and less time to correct for changes. In general, Tripwire Manager has brought all the difficult components of the original product into a user interface that is as easy to use as I any I have seen. They also have not (to their great credit) "broken" the ability to continue to edit configuration files and run tripwire from the command

line (if you are that sort).

Summary:

Host based intrusion detection is a difficult problem to solve for many reasons. To recap but a few:

- To truly know whether or not your systems have been compromised, you must monitor (nearly) every file and folder on the system.

- Each object (file or folder) has multiple characteristics or attributes that need to be considered.

- Objects that serve different purposes need to have different sets of these attributes monitored for change to allow for "normal" system operations to occur without causing false positive alerts.

- Aside from normal operating changes, the legitimate changes that are made to systems by users and administrators must be somehow accounted for.

- Knowing how to respond to various types of events needs careful thought or you can easily overload yourself with information, risk unintended consequences from automated responses, or miss something important.

As I have worked more and more with computer and network security, I have come to accept as true that the single most generally important and at the same time single most difficult task in any aspect of your security architecture is to gain a thorough understanding of your specific situation.

- In designing and running a firewall, you need to set the policy to reflect the needs of the site. What may be fine to allow through for one may be completely unacceptable at another. You need to understand if you are protecting a network of college students, or a financial system at a bank.

- Network intrusion detection requires that you understand your normal network traffic. If you see attempted access to SQL server on your LAN, is it OK or not? What about Windows IIS servers? If you see an attempt to access "default.asp" is that a normal HTTP access or a scan?

- Host based security is no different. When files or registry settings change on a system, do you need to panic and "nuke from high orbit" and reinstall? Or is it a normal change that happens as part of daily system operations?

These tools are simply part of a toolkit that helps you to better "Know thyself" … where have I heard that before?[14]

---

[14] A famous Greek maxim, often attributed to Socrates.

References:

"Alternate Data Streams: Out of the Shadows and into the Light", Ryan L.
Means, 2003
http://www.giac.org/practical/GCWN/Ryan_Means_GCWN.pdf

"Crypto Foundations" David Aspinall, School of Informatics, University of
Edinburgh, 13th January, 2004.
 http://www.inf.ed.ac.uk/teaching/courses/cs/0304/lecs/cryptoI-6up.pdf

"Description of the Microsoft Windows Registry", Article ID 256986, December
21, 2004
 http://support.microsoft.com/default.aspx?scid=kb;EN-US;256986

Tripwire home page
http://www.tripwire.com/

"Datasheet – Tripwire for Servers" Tripwire, Inc. 2004
http://www.tripwire.com/files/literature/product_info/Tripwire_for_Servers.pdf

Ethereal – "The world's most popular network protocol analyzer" – used to
troubleshoot the connectivity problem between Tripwire Manager and the
Tripwire for Servers machine
http://www.ethereal.com/

Scene 23, "The bridge of death", Monty Python and the Holy Grail
(Gratuitous Monty Python reference...)
http://www.rit.edu/~smo4215/monty.txt

Tripwire for Servers Evaluation Request
https://www.tripwire.com/downloads/tmtfs/request.cfm

"The Design and Implementation of Tripwire: A Filesystem Integrity Checker"
Purdue Technical Report CSD-TR-93-071, Gene H. Kim and Eugene H.
Spafford, Nov 19, 1993 (requires postscript viewer)
http://www.cerias.purdue.edu/homes/spaf/tech-reps/9371.ps

Ghostscript & Ghostview – Utilities to view Postscript files on Windows.
http://www.cs.wisc.edu/~ghost/gsview/index.htm
(the download link for the 8.50 release)
http://www.cs.wisc.edu/~ghost/doc/AFPL/get850.htm

GFI Languard System Integrity Monitor: Features and Download page
http://www.gfi.com/lansim/lansimfeatures.htm
http://www.gfi.com/downloads/downloads.aspx?pid=LANSIM&lid=en

GFI Languard System Event Log Monitor (features)
http://www.gfi.com/lanselm/lanselmfeatures.htm

Osiris – A Host Integrity Monitoring System
http://osiris.shmoo.com/index.html

Windows File Protection – MSDN Library, general information and reference
http://msdn.microsoft.com/library/default.asp?url=/library/en-
us/wfp/setup/windows_file_protection_start_page.asp