PestPatrol in a Corporate Environment.

A Case Study In Information Security.

Author: Tim Strong
Assignment: GSEC 1.4b – Option 2
Date: November 19th, 2002.

Table of Contents

Abstract

Viruses are no longer the only source of malware on corporate networks. There are many other types of malware, otherwise known as pests, to now contend with. These pests range from cookies to keyloggers, from spyware to sniffers. In a study performed in October 2001 by National Software Testing Labs (http://www.pestpatrol.com/NSTL/NSTL_Report.pdf), virus scanners do not detect all pests.[i] Enter PestPatrol by PestPatrol Inc. PestPatrol is designed to detect malware beyond that of viruses. While some pests that PestPatrol detects are relatively harmless, many others are not.

This paper discusses the PestPatrol version 3.2 product as deployed in a corporate environment. It covers planning the deployment, the actual deployment itself and a post deployment analysis. Within those sections, it will specifically focus on why PestPatrol was chosen, overcoming some limitations of the product and keeping PestPatrol in an operational state once it has been implemented.

Pre-deployment situation

The Situation In General

Information security incidents are in an ever increasing trend from when CERT started tracking them in 1988. A quick look at the CERT statistics (http://www.cert.org/stats/cert_stats.html) shows that the number of incidents reported has increased every year since 1988.[ii] The only exception to this was in 1997. These incidents are no longer the mischief incidents that were so popular five to ten years ago. Back then, they mostly comprised of deleting or modifying files and individual (PC) DoS attacks. Their principal mode of propagation was 3.5" floppies that had been infected with the virus. Other than these sneaker-net propagated viruses, there were not a lot of pests in widespread use.

Recently though, pests have encroached into just about every aspect of computing. Pests are no longer limited to viruses, but now envelope an entire plethora of different types of software. Even software which is destined for legitimate use can be used maliciously in the wrong hands.

The internet has become the largest purveyor of pests. Many web site cookies are now considered pests due to the fact that they track your web browsing usage across many sites. Not only has the internet helped the aspect of pests in propagation, but it has also helped in the area of information retrieval and control. When PCs were standalone or limited to LANs, an outsider had no way of easily retrieving information or controlling some aspect of that PC or network. With the advent of the internet, pest developers had new motivation for creating pests that would do more than delete or modify files and/or stop individual PCs from working. They now had a medium for surreptitiously retrieving information or taking control of a PC without the user's knowledge.

This had huge ramifications for corporations that were particularly sensitive to information falling into the wrong hands. An entire new set of tools was now available to anyone wanting to conduct corporate espionage. Information warfare aside, an article by Pete Cafarchio in the TISC Insight Newsletter classifies these items as pests[iii]: Keyloggers, Remote Administration Trojans (RATs), Commercial Remote Administration Tools, Hacker tools, DDoS zombie agents, Spyware and Adware. In addition to this list, our corporation also classified any legitimate administrator tool as a pest if it was found to be in unauthorized use or unauthorized possession.

Taking all this information into account we decided to take a look at how it would apply to our company and the possible impacts it could have.

The Situation At Our Company

Our company is a leader in satellite broadcast technology. As in any highly competitive industry, every effort has to be made to protect information, minimize downtime and reduce TCO. A substantial portion of our network is dedicated to customer service. This was an influencing factor in deciding to implement a pest control system.

Our customer service representatives (CSRs) need fairly unrestricted (but still monitored) access to the internet. This is so they can respond to questions from customers about the competition, various products and the programming coming from our satellite. Unfortunately this “internet friendly” environment can produce a lot of pests in the form of cookies and adware. No doubt it also produced an entire slew of other pests that were not as readily detectable as cookies and adware at the time. The CSRs are not assigned permanent seating when it comes to their job functions. As a result, they are configured with Windows NT roaming profiles. Many pests reside in the roaming areas of the profile. These profiles in turn are stored on network servers. This can severely affect logon and logoff time as the profile containing the pests are copied back and forth between the client workstation and the profile server.

Despite the fact that the customer service department weighed heavily on the solution, other departments such as sales, marketing, IT/IS, etc. were taken into consideration as well. In addition to departmental roles, our current infrastructure also came into play.

As far as virus protection goes, we were in a migratory phase from Innoculan to NAV when we started looking at pests on our network. While we were confident that our current setup could block all major viruses, we wanted more protection.

The net result of our pre-deployment environment was not a good one. We had thousands of unrestricted, internet connected PCs that could easily be infested with pests. As a result of an infestation, a large portion of those machines and users could have severe performance impacts due to the number of pests on the machine or in their roaming profile. In addition to performance impacts, there was also the concern that sensitive information could be transmitted out to the internet by some of the more malicious pests.

Deployment

Choosing A Product

Before deploying a pest control solution some limited research was done on various products. However, there were two main factors in our decision to choose PestPatrol as the choice product. The first was the recommendation from our main software supplier. We had been dealing with our software supplier long enough that we felt confident that they knew our environment and carried a product that would fill the void. From our experience, our supplier typically carries only the top or near top products from across the industry. While this has some drawbacks, such as an ever changing lineup of products, it also has some time saving benefits when it comes to evaluating and finding solutions. The second factor that affected our decision was a report produced by the NSTL in 2001. The highlights of the report mentioned:

> - The product with the highest overall detection rate was PestPatrol, detecting 36% more pests than the next best product, PC-Cillin 2000. PestPatrol detects nearly twice as many as McAfee, Norton AntiVirus and Trojan Remover, and at least three times as many as the remaining competitors.
> - Aside from Pestpatrol, no product detected more than half of the available pests except for PC-Cillin
> - Of the 13 products compared, PestPatrol was, by far, the most effective tool for detecting pests in each of the categories tested.
> - Of the category of pests used in the test, only PestPatrol was able to detect spyware tools. [iv]

With a product decided upon, we set about determining the best method for scanning our environment. The PestPatrol network documentation for v3.2 mentions two methods for scanning a networked environment: (1) starting PestPatrol Command Line (CL) version from a logon script and (2) scheduling a scan on each individual PC.[v] In addition to these two methods, PestPatrol Inc also has instructions posted on their website on how to scan shares.[vi] We evaluated all these possibilities and selected the most appropriate one for our environment: PestPatrol CL initiated via the logon script.

Determining a Scan Method

None of PestPatrol's network solutions are ideal and each one has it's own shortcomings. Starting PestPatrol from the logon script has the huge shortcoming in that it runs in the user context on the workstation. This means that it will not scan any folders that the user does not have access to. The result could be a pest that is hiding somewhere on the PC in an area that has not been scanned. For us, this was a reasonably acceptable risk when we examined it in more detail. In order for a pest to install itself properly, it would have to be installed in an area on the PC that the user has access to. In order for the pest to run properly on subsequent boots with a different user, it would need to be in area where all users have access. If a pest could only run for a specific user, it would be counter productive to the goal of that pest. So the upshot of this is that logon script scanning could conceivably miss some pests, but the risk is worth it.

Moving on to the second option for scanning a network, we find a huge administrative burden. This method involves scheduling a scan on each PC. The burden comes from having to manage scheduled tasks on over 1500 PCs. While we are sure that there is software that exists that will let you do that easily, it was outside the scope of the PestPatrol project. In addition to managing the scheduled task for existing PCs, there was also the consideration of changing the PC build images for new installs.

The last option which PestPatrol Inc lists is share scanning. While this solves some of the problems of the previous two solutions, it places a heavy demand on the network. Since you can scan administrative shares, you can scan the entire drive of the workstation. This avoids any part of the disk being missed. As with the second solution, the administrative demand could also be significantly heavy. In one sub-scenario, you would need to maintain a list of all UNC names to scan… quite a task in itself. A second sub-scenario would be to build the UNC list at scan time. This would involve some scripting toil, but could very well be worth it if you have the capacity and opportunity window to conduct the scan on your network. “Capacity and opportunity window” is mentioned because of heavy demand this type of scan places on the network. Since PestPatrol CL is executing on a single server and scanning shares, it is pulling every file from that share across the network to be scanned locally. Scanning one PC at a time reduces the load on the network, but would take an unfeasible amount of time in an environment as large as ours. Scanning multiple PCs at once would reduce this time, but place a huge load on the network. Further testing would need to be done in this area to ensure you could scan all your PCs in a reasonable timeframe if this method was chosen. At the time of writing, PestPatrol had just released v4 which includes the ability to automatically detect and scan the administrative shares of W2K machines on the network.[vii] This feature was not available for our deployment. Even with this new feature, the rules for network load still apply.

In summary of the scanning options, we chose the one that had the least administrative & network burden, but could potentially miss something. However, this might not be the best option for other environments where due to the size of the network there would be less administrative and/or network burden.

Testing

Once we had decided on a scanning method, it was time to test. The test environment consisted of 7 desktop PCs running W2K professional and 2 PCs running NT4. One of the W2K machines acted as the PestPatrol server (\\pesttest). The rest of the PCs were typical workstations within our department.

The goals of our initial tests were to determine what PestPatrol would find, how much of the PC resources the scan would consume and what logging information it would yield.

PestPatrol was installed on the \\pesttest server. The PestPatrol directory was shared as pp$ to give the logon script access to execute the PestPatrolCL program and allow access to the logging folder. The logging folder is subfolder in the PestPatrol directory.

With a server in place and the correct permissions set up for executing and logging, the next step was to decide on the numerous switches and options that PestPatrolCL permits. The PestPatrol v3.2 documentation says that the command line options permit control of where to scan, what to scan, when to scan, logging options, detection options and notification options.[viii] Once we had evaluated and tested various options, we built our command line as follows:

```
start \\pesttest\pp$\pestpatrolcl.exe /delete /wait=600 /nosound /nopause /spycookienoalert
/nologafter /log=\\pesttest\pp$\logs\%computername%.log
```

Start \\pesttest\pp$\pestpatrolcl.exe actually initiates the PestPatrolCL program. Start is used to start the program in a separate process than the logon script window. This allows the logon script to finish processing.
/delete specifies that we want to delete any pests found.
/wait=600 tells the PestPatrolCL program to sleep 600 seconds before starting to scan. Our environment has a relatively heavy startup sequence. By delaying the scan for 10 minutes, it ensures that all the other startup applications have finished running and the PC is relatively idle.
/nosound specifies that no sound should be made after a pest is found.
/nopause specifies that no pause or popup windows should occur after a pest is found.
/spycookienoalert tells PestPatrol to detect all adware and spyware cookies, but not to send any alerts when it does so.
/nologafter prevents the log from being displayed when PestPatrolCL has finished scanning.
/log=\\pesttest\pp$\logs\%computername%.log tells pest patrol to create an individual log file for each PC scanned.

The most interesting aspect of these options is the decision to create a separate log file for each PC scanned. The PestPatrol program creates 5 lines of text for each pest found in addition to an 18 line header at the beginning of the file. With this, initial scans could easily run into 50kb. With thousands of PCs to scan, we quickly came to the conclusion that one giant log file would get very cumbersome. However with many small files, the task could become very tedious for reviewing the logs.

Manually inspecting the logs we noticed that in our test machines, the bulk of the information were spyware cookies. Not wanting to be too concerned with spyware cookies, we opted for automatically scanning the logs for anything else. While there are many third party flat file scanning tools available, we decided that the file format was simple enough we could extract the information we required by using native NT4/W2K commands. The following batch file executed once a night on our test server.

```
@ echo off
rem date & time stamp master file.
date /t > c:\program files\pestpatrol\logs\masterlog.tmp
time /t >> c:\program files\pestpatrol\logs\masterlog.tmp

rem parse all *.log files looking for "Found" string for a pest.  Write file name to log and copy file for
inspection.
for %%f in (c:\program files\pestpatrol\logs\*.log) do for /f "tokens=1,2" %%i in (%%f) do if
%%i==Found if not %%j==0 @echo %%f =^> %%j >> c:\program
files\pestpatrol\logs\Masterlog.tmp & copy %%f c:\program files\pestpatrol\logs\pest\*.*

rem cleanup *.txt files and rename tmp log file to txt.
del *.txt
ren masterlog.tmp *.txt
```

The first section simply creates and timestamps a master log file. The core of the batch file are the embedded “for” commands in the second section. The goal of this paper is not to teach the NT “for” command, so the explanation will be a high level overview. A good explanation of the command can be found in the Tips & Tricks section of the JSI Inc website (http://www.jsifaq.com/subg/tip3200/rh3243.htm).[ix] NT “for” permits stepping through a folder or file and extracting tokens. Each token is determined by the delimiter specified in the command. Our command uses a “for” to step through every file in the log folder and another nested “for” command to determine if any pests were found in each log file. If the pest count for the log file does not equal zero, then the file is copied to another folder and the filename (same as the computer name) is added to a master log file. The end of batch file simply removes old files and cleans up from the batch file execution.

We now had a test environment that was stable and met all of immediate goals for controlling pests within our enterprise. The last step was to scale it up to a production environment with over 1500 PCs.

Scaling Up To Production

The first item to do was find a suitable distribution point. PestPatrol inc recommends installing the software on each NT domain controller.[x] However, in an faq on the PestPatrol inc website, it says it is possible to relocate PestPatrol by simply moving any pest*.* file.[xi] Further experimenting along these lines indicated that it was possible to run PestPatrolCL with only four files: PestInfo.dat, PestPatrol.bin, PestPatrol.dat and PestPatrolCL.exe. Simply moving these files to our distribution points rather than installing the entire product seemed like a much more efficient solution. To make the distribution even easier, we decided to create a PestPatrol folder under the logon scripts folder on our primary domain controller. Placing the four required PestPatrol files in this folder would have the result of automatically distributing PestPatrolCL to our geographically separated backup domain controllers.

The next major thing to change in scaling up our test environment was the location of the logs. Instead of using a subfolder of the PestPatrol directory, we setup a new share on a file server exclusively for PestPatrol logs.

Both of these changes required some small modifications to the command line we had developed for the test environment. The final command line looked like:

```
start %logonserver%\netlogon\pestpatrol\pestpatrolcl.exe /delete /wait=600 /nosound /nopause
/spycookienoalert /nologafter /log=\\serverA\pplogs$\%computername%.log
```

The only differences between this and our test environment command line are the path to start PestPatrolCL and the path for the logs. There were also some slight path changes to the batch file that scans the log files looking for pests.

The actual implementation to the users was done in phases. We have several logon scripts that cover a large percentage of users within the enterprise. The PestPatrol command line was added to one script every few days for about a month. The result was a very smooth, incremented, controlled deployment. The only minor surprise was the size of the logs and the logs folder whenever PestPatrolCL scanned PCs for the first time. We did speculate that there could be a lot of data during the initial scans, but it surprised us none the less as to how much there actually was.

It didn’t take PestPatrol very long to find some interesting things on our network. Apart from the massive amounts of spyware that people accumulate from regular web browsing, there were a few things that really caught our attention. The most surprising was the sheer number of RATs… Remote Access Trojans. The surprise came from both the quantity and diversity. Trojans are by far one of the most prolific and potentially damaging forms of malware on the internet today. A paper by Dancho Danchev on The Security Writers Guild website states that Trojans will continue to become more complex with many more features available to their makers and users.[xii] Because our internal LAN is using NAT behind a firewall, the chances of an inbound connection actually being able to connect to a RAT are low. However, the chances of exploiting a RAT internally are much greater. A simple port scanner and a list of Trojan ports, such as the one by Joakim von Braun on the SANS website[xiii], can go a long way in aiding an internal exploit.

At the end of the deployment we were detecting and deleting pests for all the PCs within the company. Even though the bulk of the work was finished, PestPatrol, like most other security products, requires a certain amount of maintenance after installation. Items to consider can be areas such as up to date definition files, reviewing the logs, future direction of the product within the company.

Post Deployment

Dealing With The Logs

The most complex issue after the deployment was dealing with the logs. This can be broken down into three issues we were experiencing: (1) Filtering the logs, (2) Reviewing the logs and (3) Securing the logs.

Even though every precaution was taken to minimize the visibility of the logs, it soon became apparent that with the type and quantity of data collected by PestPatrol it would be a prudent idea to secure the logs further. Executing PestPatrol in the user context means that the user must have certain NTFS permissions to the log folder. We experimented with different types of write and append permissions, but did not find a way where they could only append data to the end of the file without being able to read it. Having read ability on the folder & files could mean that someone could exploit the information we were collecting for there own means and/or overwrite information about pests found on the network.

We also found that even reviewing the non-cookie logs was taking more time than anticipated. We wanted to filter it down to only find the really bad stuff on our network. The problem was that once we had filtered out “cookie-logs” there were a lot of logs that contained adware. While we do have a concern for adware pests, it is nowhere near the level of concern we have for trojans.

We addressed all three of these issues with a couple of quick fixes. The security issue was fixed by executing our “cookie-filter” script every 30 seconds and moving the pest infected files to a secure area. What was left in the normal log folder was basically a list of cookies. The reviewing and filtering was vastly improved by using a second filter script utilizing once again the “for” command. By searching for specific adware pests and moving them to another folder, we are left with a folder <u>not</u> containing the pests we specify. Here is the “for” command that moves files that have Gain adware in them.

```
for %%f in (d:\pplogs\pest\*.log) do for /f "tokens=1,2" %%i in (%%f) do if %%j==GAIN @echo %%f
=^> %%j >> d:\pplogs\pest\gainlog.txt & move %%f d:\pplogs\pest\gain\
```

If someone wanted to they could even expand on this idea to neatly classify their logs per pest.

The Future Of The Deployment

As we become more familiar with the product and as our environment changes we will constantly evaluate new ways of integrating PestPatrol into our company. One of the greatest drivers of this will be the currently unused features and/or new releases of the PestPatrol software. With version 3.2, which we currently have deployed, the only currently unused feature is the memory scanner.

The PestPatrol memory scanner works by detecting the pest signature at runtime and terminating the process with that signature. While it is highly unlikely we will implement this feature within out current v3.2, we might consider it if we move to v4. One of the largest problems that affects the disk scanner also affects the memory scanner. That is the fact that in a corporate environment, the most cost effective way to deploy it is via the logon script. As with any process that runs from the logon script, it is possible for the user to terminate it. While most users welcome added privacy and security, anyone with the intention to use malware could very well be aware of a product like PestPatrol and defeat it.

Even though there doesn't appear to be any significant install or deployment methodology changes in v4, a couple of interesting components have been added. These would be CookiePatrol and KeyPatrol. CookiePatrol eliminates spyware cookies before they are placed onto the computer and would probably be a welcome addition for anyone. KeyPatrol is a key logger detection application that supposedly detects key loggers which do not have signatures in the PestPatrol database.

Other than these two components, more options have been added to PestPatrolCL. Most notably are the /idle and /shares switch. The /idle switch would reduce the performance impact when PestPatrol scans a PC. Even though in our tests we found that a local PestPatrol scan had little noticeable impact on performance, we did notice that in production a performance hit was indeed noticeable. We have yet to determine the root cause, but we speculate that is because there is simply too much going on when the PC starts up… virus scanner, s/w inventory scanner, h/w inventory scanner, MS Findfast, etc. Even with a 600 second scan delay on PestPatrolCL, it doesn't guarantee that the PC is relatively idle at that point. Utilizing the /idle switch could smooth out a few performance issues.

The /shares switch is going to take some intensive testing. While the idea of this switch appears nice on the surface, the usefulness of it might be limited in large environments. The pros and cons of share scanning were covered earlier on. By scanning a few shares at a time, it could take an inordinate amount of time to complete an entire enterprise. By contrast, bulk scanning shares at the same time could create network and/or server congestion.

Other Ideas

A final thought on what could be done with PestPatrol is to use it to monitor the deployment of legitimate remote control tools. In a March 2002 press release by Peter Cafarchio of PestPatrol Inc, PestPatrol Inc decided to include the signatures of commercial remote administration products after a memo was released from the US Navy's Computer Incident Response Team (NAVCIRT).[xiv] Brian McWilliams of Computer User says that the unclassified memo mentions that the US Navy is currently in the midst of an investigation which aims to remove a commercially available product from their systems.[xv] This is of a great benefit since some users bypass security measures by simply installing a modem and a product like pcAnywhere.

Despite the fact that many reviews such as the one by Stu Craine of Compunotes[xvi] or Chad Todd of MCP Magazine Online[xvii] advocate that PestPatrol is a mature easy to install product, there is very little on PestPatrol as deployed in a large environment. Along those lines it would be nice if PestPatrol inc developed a stronger centrally managed corporate product. Ideas of this could be an administrator console which could remotely install a PestPatrol agent as a service on the workstations. As a result the agent could run in an administrative context which would be harder for a user or hacker to disable or bypass. Continuing along these lines, the agents could also report back to central location using a proprietary protocol. The PestPatrol "server" could even act as middleware when it comes to logging and put the logs into an ODBC backend. This could be extremely useful for creating your own reports and/or tying into an IDS data correlator.

Conclusion

PestPatrol has brought a greater level of security to our network. However, getting to this point has been a rather arduous process. There are many avenues left to explore with PestPatrol in securing our environment and as such, it will become an integral part of our infosec plan.

Any company large or small that has a high level of concern for information leaks should consider PestPatrol. We felt that its ability to stem unauthorized transmittal of information to outside the organization was one of its strongest assets. It doesn’t take much to get a VP to execute a trojanized file which will then send keystrokes and files to someone outside the company.

If you are an auditor, PestPatrol should be a part of your toolkit. It is simply a good practice to ensure that an environment is free from pests.

The other benefits of PestPatrol such as spyware cookie detection and adware detection are great for simply maintaining a clean environment. While it is unlikely that these will severely affect a corporate network over the short term, it is nice to know that you can control them.

Overall PestPatrol has been worth the time, effort and money invested. With plenty of testing and some original ideas we were able to integrate it into our environment without a great deal of impact to the users. We are looking forward to future versions of the product which will no doubt be even better for a secure corporate environment.

Endnotes

[i] "Test Report for Safersite, Inc." October 2001. URL:http://www.pestpatrol.com/NSTL/NSTL_Report.pdf (15 October 2002). p.8.

[ii] "CERT/CC Statistics 1988-2002.“ 4 October 2002. URL:http://www.cert.org/stats/cert_stats.html (15 October 2002).

[iii] Cafarchio, Pete. "The Challnge of Non-Viral Malware." TISC Insight. Volume 4, Issue 12. August 2 2002. URL:http://www.tisc2002.com/newsletters/412.html (15 October 2002).

[iv] "Test Report for Safersite, Inc." October 2001. URL:http://www.pestpatrol.com/NSTL/NSTL_Report.pdf (15 October 2002). p.1.

[v] "PestPatrol In A Networked Environment." Version 3.1.0423. 2002. URL:http://www.sunbelt-software.com/evaluation/911/web/documents/ppce_implementation_guide.pdf (17 October 2002). p.4, 7.

[vi] "How to Scan Shares." 2002. URL:http://www.pestpatrol.com/Support/HowTo/How_To_Scan_Shares.asp (October 17 2002).

[vii] "PestPatrol In A Networked Enviroment." Version 4.0.930. 2002. URL:http://www.sunbelt-software.com/evaluation/911/web/documents/ppce_implementation_guide_v4.pdf (October 18 2002) p.7.

[viii] "PestPatrol In A Networked Environment." Version 3.1.0423. 2002. URL:http://www.sunbelt-software.com/evaluation/911/web/documents/ppce_implementation_guide.pdf (18 October 2002). p.5-6.

[ix] "Windows 2000 FOR command enhancements." JSI FAQ. URL:http://www.jsifaq.com/subg/tip3200/rh3243.htm (October 18 2002)

[x] "PestPatrol In A Networked Environment." Version 3.1.0423. 2002. URL:http://www.sunbelt-software.com/evaluation/911/web/documents/ppce_implementation_guide.pdf (28 October 2002). p.4.

[xi] "Running PestPatrol On A LAN." URL:http://www.pestpatrol.com/support/faq/lan_faq.asp (28 October 2002).

[xii] Danchev, Dancho. "The Future of Windows Trojans." The Complete Windows Trojans Paper. 24 October 2002. URL:http://www.securitywriters.org/texts.php?op=display&id=58 (28 October 2002).

[xiii] Von Braun, Joakim. **"**What port numbers do well-known trojan horses use?" Intrusion Detection FAQ. 09 February 2001. URL:http://www.sans.org/newlook/resources/IDFAQ/oddports.htm (28 October 2002).

[xiv] Cafarchio, Pete. "PestPatrol Detects and Removes Threat from Misuse of Remote Administration Tools." 21 March 2002. URL:http://www.pestpatrol.com/News_Media/Releases/2002_3_21_Remote.asp (16 November 2002).

[xv] McWilliams, Brian. "U.S. military scours Windows systems for hacker back doors." Computer User. 18 March 2002. URL:http://www.computeruser.com/news/02/03/18/news1.html (16 November 2002).

[xvi] Craine, Stu. "Software Review of PestPatrol v3." CompuNotes. URL:http://www.compunotes.com/InternetReviews/pestpatrol3.htm (16 November 2002).

[xvii] Todd, Chad. "Keeping It Clean." Microsoft Certified Professional Magazine Online. October 2002. URL:http://www.mcpmag.com/reviews/products/article.asp?EditorialsID=339 (16 November 2002).