

Technical White Paper

July 24, 2005 (updated)

Christopher L. T. Brown, CISSP

clbrown@techpathways.com

www.TechPathways.com

Technology
Pathways

Suspect Host Incident *Verification* in Incident Response (IR)

Abstract

Given the frequency of alerts from today's Intrusion Detection Systems, most system administrators have experienced that dreaded thought "My system has been hacked!" followed by hours of poking around the suspect file system and log files trying to confirm their suspicions. Let's face it; our security fears in information technology are reinforced on a daily basis by our own experience as well as reports in the media. Based on the 2003 Computer Crime and Security Survey published by the Computer Security Institute, only 70% of the companies surveyed indicated they experience unauthorized use of their computer systems over the past 12 months. Moreover, the CERT Coordination Center at Carnegie Mellon shows that over 135,000 incidents were reported in 2003. No one really knows how many went unreported. Given the real threats that exist today, a measured approach to investigating suspicious hosts can help administrators maintain their sanity.

This white paper discusses technical incident response methods that can help you quickly evaluate the status of a suspected host. While examples used in this paper focus on Windows hosts, much of the information outlined in this paper will pertain to all operating systems.

Background

For years people have talked about incident response planning and the need for a detailed incident response plan to guide you through difficult procedures in times of confusion. Many of today's incident response guides fail to address steps to assist administrators to adequately verify that an incident has occurred. Furthermore many plans neglect to outline procedures for evaluating an incident in a manner that will properly maintain and preserve evidence for possible future civil or criminal litigation.

You see it over and over again; an administrator suspects a machine has been hacked and they start searching through the file system looking for anything out of the ordinary. Next they sift through local system logs. Unfortunately, the system administrator can't trust what they see because the system may have been hacked... but they do it anyway. It's the distrust in what they are looking at that causes the administrator to continue to delve deeper into the suspected system trying to find anything that will be conclusive, but the trust issue is there preventing conclusive findings.

Over the years savvy system administrators have developed two methods to help resolve trust issues:

1. Create cryptographic hashes of important files on the file system. In this approach the administrator who suspects a compromised host they can create new hash values and compare the new hash values to a set of "known good" values. (see sidebar)
2. Use a set of known good applications, sometimes referred to as "trusted binaries" to investigate the suspected host running from a CDROM, or remote disk.

Did you know? A Cryptographic hash is an algorithm used to produce fixed-length character sequences based on input of arbitrary length. Any given input always produces the same output, called a hash. If any input bit changes, the output hash will change significantly and in a random manner. Additionally there is no way the original input can be derived from the hash. Two of the most commonly used hashing algorithms are MD5 and SHA 1.

Unfortunately today's hackers can easily affect a host at a much deeper level than merely replacing files to cover their tracks and set up services. Hackers achieve this deeper infection by installing one of the widely available "Kernel Mode Rootkits". These rootkits are implemented as device drivers in Windows platforms and LKM's (Loadable Kernel Modules) in Linux.

To better understand “Kernel Mode Rootkits” let’s take a look at the basic principles of the security kernel architecture used in Windows NT/2000/XP platform design. Microsoft divides the operating system into two modes;

User Mode - This is where all general applications operate. General applications and subsystems for Win32, Win16 and POSIX (Portable Operating System Interface) all run in this mode.

Kernel Mode – This mode is a trusted mode of operation for system services and device operations or access. All requests by user mode applications are brokered through Windows NT Executive Services within the kernel mode. This includes checking security ACL’s (Access Control List) and allowing access to file I/O and attached devices.

Did you know? Early rootkits only replaced user mode applications such as “netstat”, “dir”, etc. By replacing “dir” a hacker could control the “dir” application output (set to not display certain files), but “dir” would still need to request all file I/O from a protected source in the kernel mode. It was these early Rootkits that hashing and trusted binary schemes were designed to overcome.

The current approach to “Kernel Mode Rootkits” is simple. If the goal is to hide a file or process, rather than replace “dir” or “netstat”, why not replace the command user mode applications would call for information from in the kernel? In the case of file I/O we need to replace the kernel mode I/O routine “ZWQUERYDIRECTORYFILE” In this approach not only will “dir” be able to hide hacker’s files, but any other application which makes a call to the kernel mode I/O routine “ZWQUERYDIRECTORYFILE” will receive compromised information. Hackers accomplish this by writing a Windows device driver that through a process called “Hooking” replaces trusted kernel mode I/O routine with their own. Of course the hacker’s routine only provides information they want users to see.

The implication of these relatively new hacking techniques is that comparing hash values of files on the system is useless because any hashes created on the system cannot be trusted. The newly created local hashes would use local system I/O and the user mode files most likely didn’t change anyway. Using trusted binaries running locally will not help for the same reasons.

Another important issue to consider is all that the investigation on the suspect system, even when using trusted binaries from a CDRom, change almost every file’s last accessed time. If it turns out there has been an incident, tracking hacker’s actions becomes more difficult as you lose valuable time stamp information and these changes can raise authenticity issues in legal proceedings.

Resolving Trust in Incident Identification

One accepted way to detect a kernel mode rootkit is to reboot the suspected system in “Safe Mode”, then look around for anything that’s been hiding. Another way is to connect to the suspect system file shares from a trusted remote system (using the trusted remote system’s I/O and trusted binaries) then explore as before. In the first case taking the server offline for mere suspicion is rarely an option. In both cases files last access times will be changed and the question may still remain “are the trusted binaries truly trusted?”

How do you read files from a live systems disks with trusted I/O and not change any last accessed times?

To answer this need Technology Pathways has introduced a new network enabled version of its computer forensics product, ProDiscover® *Incident Response* which reads the live disks sector-by-sector then implements a read-only file system on a trusted system for analysis of the suspect system. See sidebar for a discussion of possible attacks. ProDiscover® IR’s core features provide the system administrator the ability to investigate suspected systems in a *least-intrusive* manor, seeing all the files on the suspect system while preserving evidence for possible criminal or civil litigation. ProDiscover IR has three ways to locate malware on a suspect system. First is by searching based on hash signatures. Investigators may compare all files on the suspect system to known hash signatures of malware. ProDiscover provides over 1200 hash signatures of known malware files. Investigators may also use a hash signature search to check all system files against known good signatures and identify any file which has been altered. The second method is a changed based method. Here the system administrator can create a baseline of all the system files at initial setup and then use this baseline to compare to the system files at a later time to identify any adds, deletes or changes. Third is a unique behavioral based method in which ProDiscover will identify any files or processes which are being cloaked by a kernel mode rootkit. Among the other techniques the investigator may use in ProDiscover IR include imaging and searching the suspect system’s RAM memory, searching

the suspect systems registry, capturing volatile system state information, exploring all processes running on the suspect system with their dependent dll's, recovering deleted files, or searching files and disks for keywords. ProDiscover® IR is intended to be a key part of a comprehensive IR investigation including monitoring Intrusion Detection Systems, logging and auditing.

Did you know? Some administrators will suppose that if a rootkit could hook (replace) File I/O request they could simply hook the sector level read commands and foil the approach that applications such as ProDiscover® IR use. While this is true on the most basic level, hooking kernel sector read commands would have a trickle-down effect on all other kernel level file system operations and require a large amount of real-to-Trojaned sector mapping and/or specific sector placement for the rootkit and supporting files. This undertaking would not be a trivial task even for the most accomplished kernel mode rootkit author.

For quick and reliable investigations the administrator need only utilize a PDServer™ CDROM or floppy and run “pdserver.exe” on the suspected system. This will load into memory only and will not alter the disk iage. Once PDServer™ is running, the administrator connects to the suspect system from the ProDiscover® Incident Response client and adds the suspected disk to a current project. By using ProDiscover in this way, the suspect system can remain up, running, and on-line while the system administrator conducts the examination. The system administrator will see all the files on the suspect system, even files cloaked by Trojans or rootkits, and can access them without altering any valuable data or metadata.

If it turns out after investigation that the server had been compromised, full incident response procedures can be implemented that may included creating a bit-stream image of the suspect hard disk. In this case ProDiscover® IR will allow the administrator to create the image through any TCP/IP LAN or WAN in a secure channel encrypted with the 256 bit AES or TwoFish encryption algorithm.

Conclusion

Incident response is a highly procedural process in which identifying an incident alone can be time consuming and require taking critical resources out of service. This impacts overall productivity and if not done quickly, makes it impossible to capture the data needed to catch the criminal. Also, the processes utilized in identifying an incident and capturing “evidentiary quality” data can be critical to successful criminal or civil litigation. ProDiscover® IR provides administrators with solutions to quickly identify incidents and properly manage the technical aspects to the corporate Incident Response process.

Detailed Steps to Live Analysis Using ProDiscover® IR

At program launch ProDiscover® IR will present a dialog asking the user to create a new project or select an existing project as seen in figure 1.

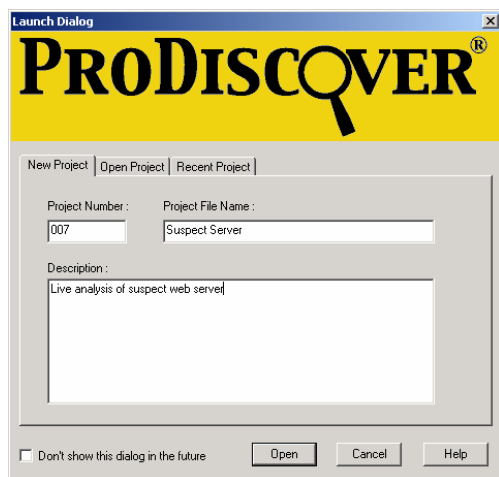


Figure 1

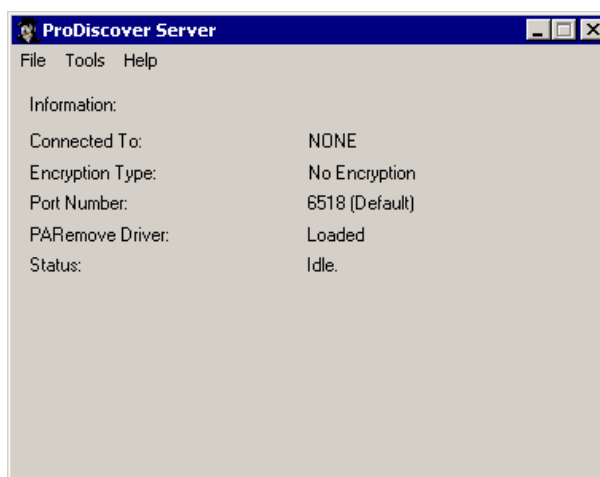


Figure 2

The user has the option to enter a project number, name, and project description in the new project tab option, then click the **Open** button to create the project. (Note all items are optional with the exception of “Project File Name”.)

From the suspect host place a **PDServer™** CDROM in the CDROM drive and execute **pdserver.exe**. Once running, the PDServer™ program window will be displayed as seen in figure 2.

Now that **PDServer™** is running on the remote system, from within an active project on the ProDiscover® IR client system use the **"Network"** menu to select **"Connect to..."** and the dialog box seen in figure 3 will appear.

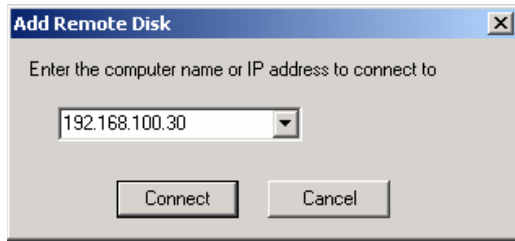


Figure 3

If the user is connecting to a PDServer™ running in Stealth Mode with a password set the ProDiscover® client will display a dialog box asking for the password prior to connection (see sidebar). Even if encryption mode is not set, TwoFish encryption is used to create a secure channel for all communications setup to prevent password sniffing and man-in-the-middle attacks.

Did you know? "Stealth Mode" is helpful to HR and Policy Compliance Officers desiring to conduct ongoing investigations. "Stealth Mode" prevents the user from knowing the system is being forensically examined. More information on "Stealth Mode" is available in ProDiscover® IR documentation.

Once the connection is established, users can create a secure channel if desired by selecting **"Encryption..."** from the ProDiscover® clients **Network** menu. If a seed key is not provided, ProDiscover® will provide its own discrete key. (see figure 4)

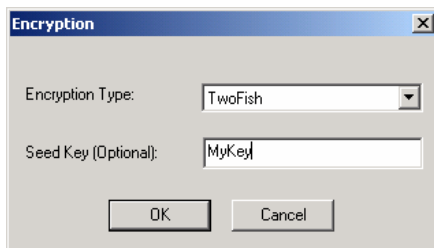


Figure 4

PDServer™ on the remote server will now show the encryption type as seen in figure 5. While encryption does cause some overhead, there is rarely a noticeable difference in performance in most LAN environments.

Once a connection is established the user will choose **"Add / Disk"** or right-click over **"Remote Drives"** in the tree-view and choose the desired remote disk in the dialog that appears. (see figure 6)



Figure 5

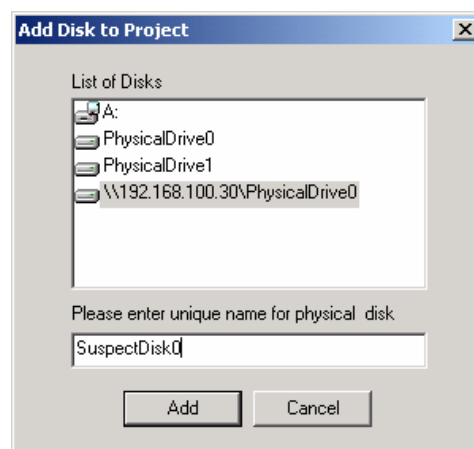


Figure 5

With the remote disk added to the project the user can now browse the file system using the ProDiscover® clients content-view item from the left hand program area. (see figure 7)

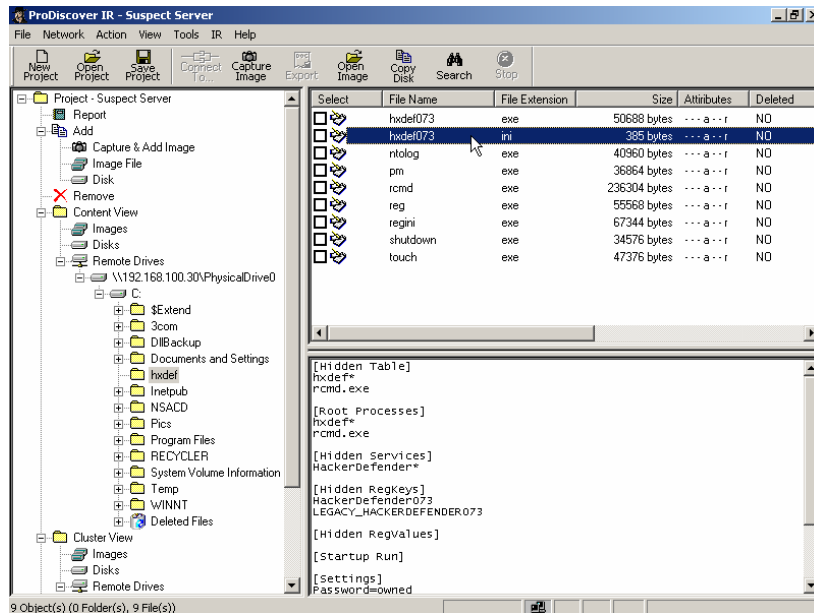


Figure 6

In the case shown in figure 7 the system administrator has found a directory “Hxdef” containing a few suspect files. When selecting the file “hxdef073.ini” the data view area shows the file contents to be alarming. None of these files were visible on the remote system locally or using a trusted binary CD because a RootKit had been installed.

When using ProDiscover IR the administrator can quickly find files hidden from the remote system’s users using the “Find Unseen Files” feature available from the IR menu. This process will compare what the file system’s low-level file tables contain against what standard file I/O system calls return. To utilize this feature the administrator first selects the directory structure they want to scan along with the types of hidden files they desire to find. (see fig. 8) Individual directories can be selected for comparison by right-clicking on the directory in content view.

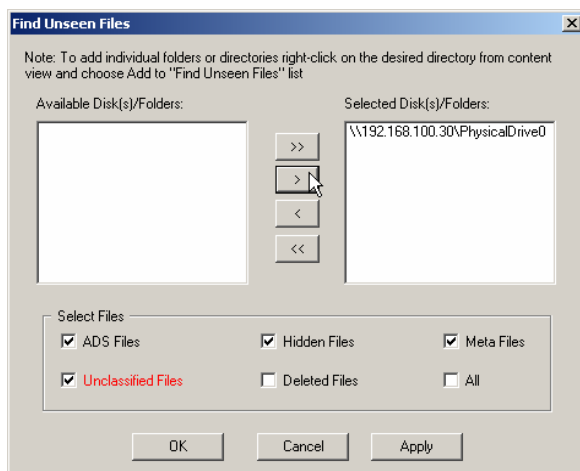


Figure 7

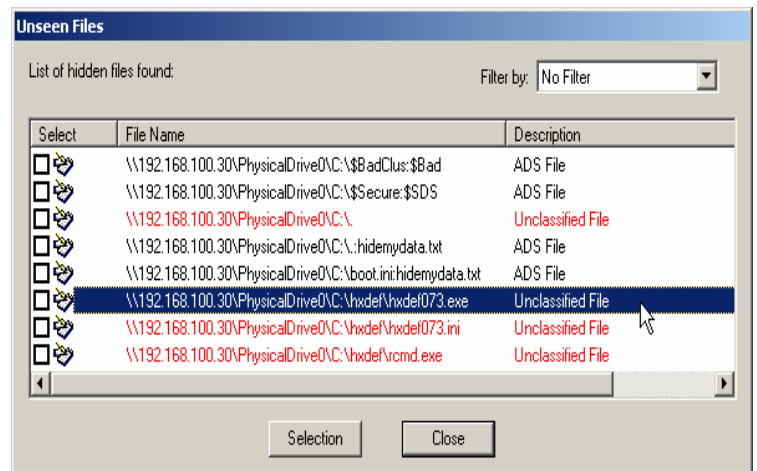


Figure 9

Unclassified Files is where Rootkits normally appear, but administrators may be interested in finding all files marked system hidden as well as ADS files. The “Find Unseen Files” process is often completed in under ten minutes as seen in figure 9.

The administrator can select any of the files found in “Find Unseen Files” as evidence of interest and may at this point want to utilize the IR Menu option to “Find Suspect Files” to conduct a more comprehensive file system search. The “Find Suspect Files” feature offers administrators a method to conduct a comprehensive file hash value comparison against a suspect files list. Technology Pathways provides hash databases for over 1200 known bad or suspect files. As when using the “Find Unseen Files” feature, the “Find Suspect Files” feature allows administrators to select an entire disk to scan using the dialog box or individual directories with a right-click from the content-view (figure 10).

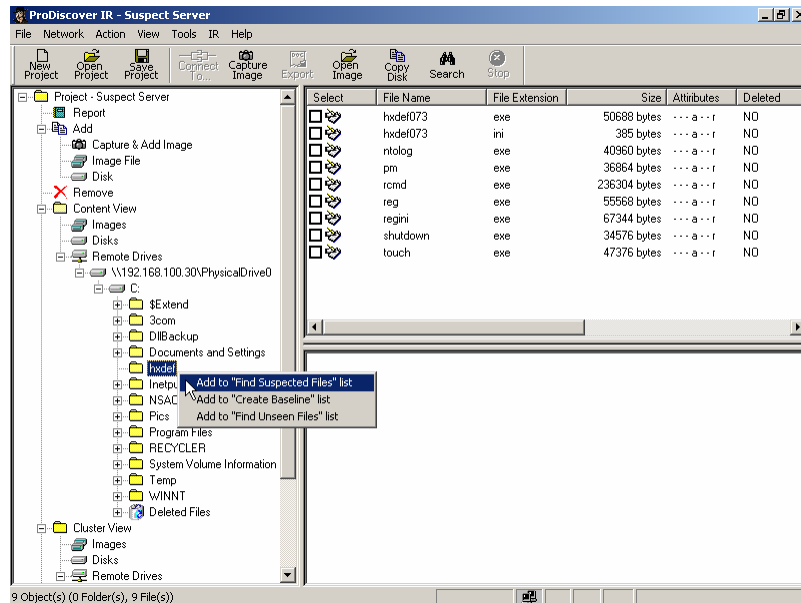


Figure 10

Once the directory or disk is selected the administrator selects any hash database (in Hashkeeper format) to scan and compare against as seen in figure 11.

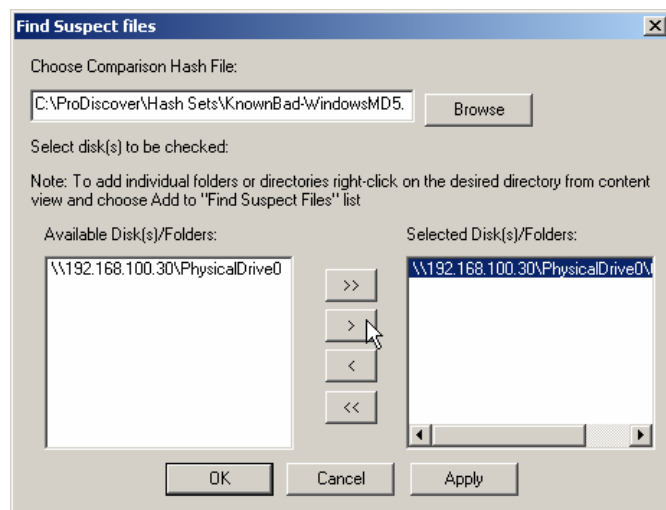


Figure 11

Once the administrator chooses “OK” ProDiscover IR will conduct hashes of all files in the selected directory path using its read-only, disk-level up file system and then compare the results file-for-file to the selected hash database. The resulting positive matches are then highlighted (blue by default), automatically selected as evidence of interest and added to the project report as seen in figure 12.

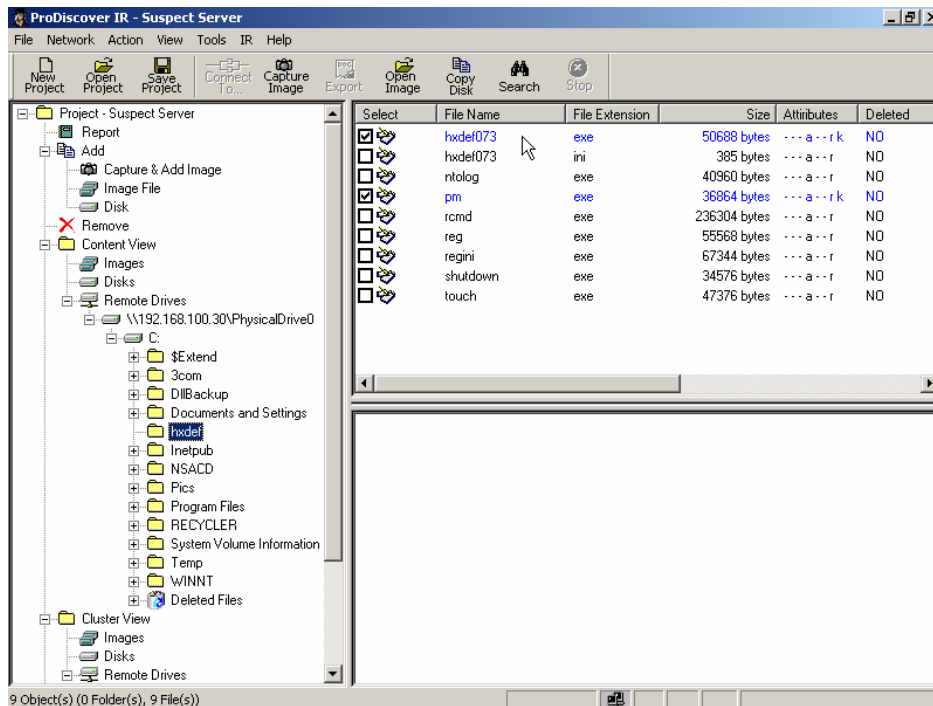


Figure 12

Tip: Administrators who wish to incorporate ProDiscover IR into a comprehensive system integrity verification strategy can use the “IR” Menu options for “Create baseline” and “Compare baseline”. Unlike tripwire these features will create a file system baseline hash database from the bottom up (disk bit level) in its read-only file system. From this point on, administrators can conduct an integrity check using the original baseline.

If necessary, the administrator can also image the physical memory of the suspect system (see figure 13) and search this for signs of memory resident only malware (see figure 14).

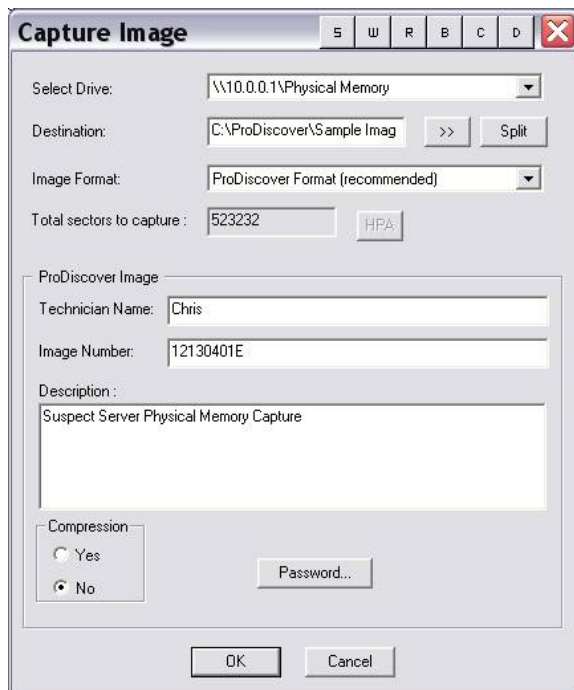


Figure 13

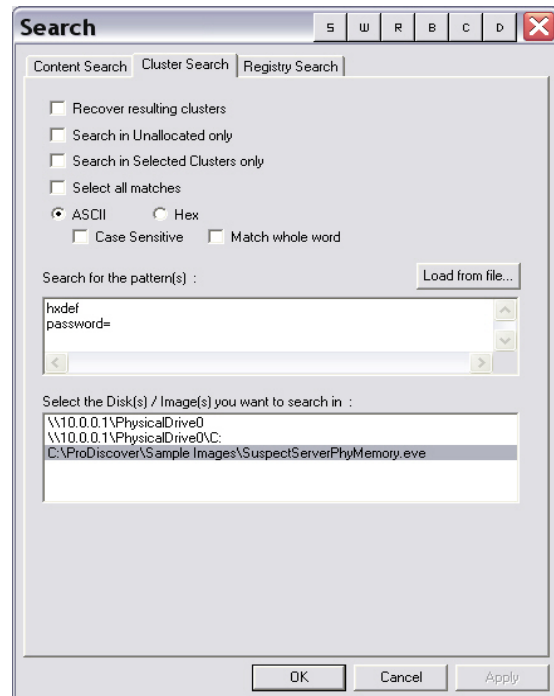


Figure 14

Additionally the administrator may search the suspect system registry for evidence of malware (see figure 15).

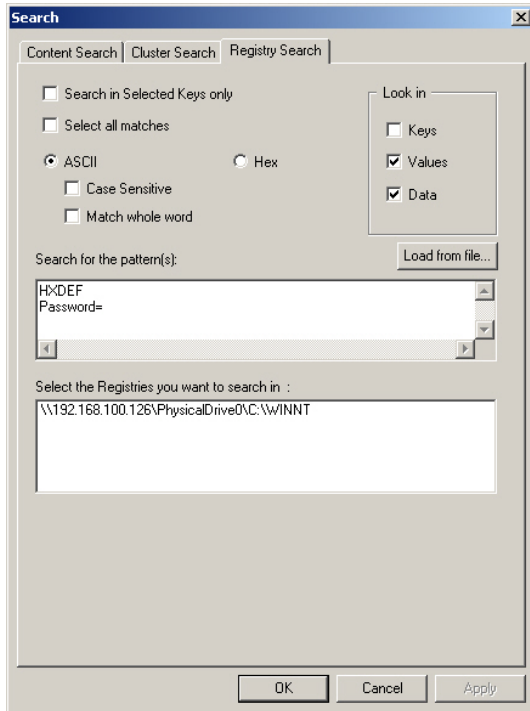


Figure 15

At this point the administrator knows they have been hacked and can fully implement a incident response plan which most likely includes creating a bit-stream image of the remote system. With ProDiscover® the administrator can create the image with a single click on the “Capture Image” icon from the button bar. (see figure 16)

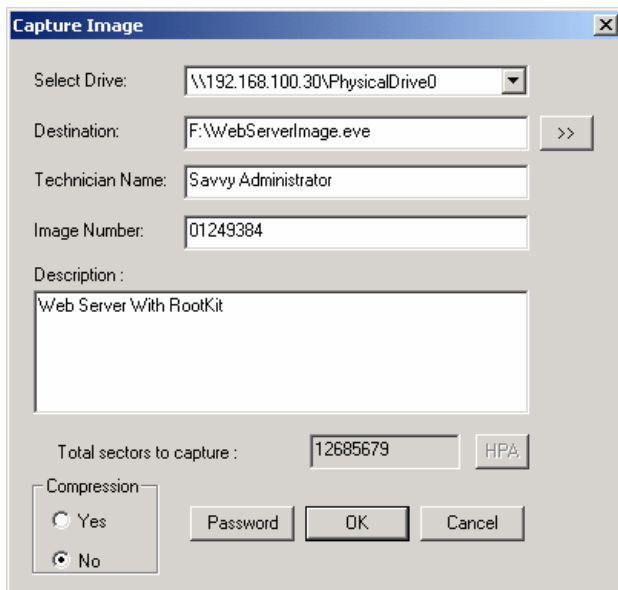


Figure 16

ProDiscover® IR provides a variety of other features which fully support accepted computer forensics methodologies as well as the complete incident response process.