

No Stone Unturned, Part Four

H. Carvey 2002-05-27

No Stone Unturned, Part Four

by *H. Carvey*

last updated May 27, 2002

Introduction

This is the fourth installment of a [five-part series](#) describing the (mis)adventures of a sysadmin named Eliot and his haphazard journey in discovering "the Way" of incident response. As we left off [last time](#), Eliot had managed to resolve a nagging minor incident, one which illustrated the need to have specific incident response procedures in place

Part IV

Eliot sat quietly in front of his workstation. The steaming porcelain cup he clutched in his lap offset the cool glow of the screen. It had been a couple of weeks since he'd worked with Dave to determine the cause of the "suspicious" activity that Dave had detected. Since then, Eliot had read more closely the messages from the public list servers he subscribed to. He also started digging through the archives. One thing he'd learned is that NT and 2K systems were "chatty", often generating copious amounts of traffic that administrators considered "suspicious". He'd also learned that while there was a great deal of information available on conducting incident response and forensics investigations on Linux, Solaris, and other flavors of Unix, there wasn't much of the same available for NT/2K. While there were a couple of Web sites that talked about using freeware tools like 'dd' to make disk images, and a few that had Linux-based tools for examining those images, the majority of sites concerned commercial products. There just weren't many Web sites that covered collecting information from NT and 2K systems that had been 'hacked'.

That being the case, Eliot had decided to take it upon himself to develop his own incident response toolkit. He looked at it as something of a challenge...he couldn't find a comprehensive set of tools and utilities for investigating NT/2K systems, so he'd make one. With his familiarity with Linux and Solaris systems, he knew he could simply identify the functionality he needed and then hunt down the necessary tools. Eliot decided that he could even write some simple tools of his own, if he needed to. From there, he'd simply have to put together a methodology

for using the toolkit. This meant that the time he had spent in front of the computer over the past couple of weeks had been tedious at times, but engaging at other times. He had found and tested quite a few of very useful tools, in several cases finding two or three tools that did pretty much the same thing. He'd decided that redundancy was advisable, as using multiple tools allowed him to verify the results.

One thing that was particular to Eliot's environment was that several of the systems he administered were file, Web, and e-mail servers. Eliot had talked to his supervisor, and had found that the primary concern of senior management was to keep the servers running. There were some servers that had their downtime measured in thousands of dollars per minute. There weren't many spare servers available, so Eliot had to be sure that he kept the systems up and running. Downtime just wasn't an option. Eliot decided that his goal should be to create a toolkit that allowed him to quickly and efficiently examine the systems in order to decide whether or not to recommend taking them down. Not only would this end up saving time and money in the long run, but the same toolkit and techniques could be used to solve problems that came up with other systems.

As he went about his search, Eliot began finding some interesting sites. At the [SysInternals](#) Web site, he'd found quite a few very useful tools that allowed him to look at the information available on running processes. [FileMon](#) and [RegMon](#) allows him to monitor processes and see what files and Registry keys they accessed. [Pslist.exe](#) shows the process information, similar to what is shown in the Task Manager. [Listdlls.exe](#) not only shows which modules, or DLLs, a process is using, but also lists the full path to the process image and the full command line used to launch the process. He found that [pulist.exe](#), from the [Microsoft](#) Resource Kit, would show the owner of the process, and when combined with the tools he'd already found, [netstat.exe](#) and [FoundStone's fport.exe](#) (displays process-to-port mappings), provides a fairly comprehensive snapshot of activity on the system.

Eliot had already had the opportunity to try out his toolkit. He'd gotten a call from a system administrator one afternoon about an employee's workstation that had something "odd" going on. The admin had left a message on his voice-mail, so Eliot hadn't been able to ask her for specifics about the situation. After several attempts at phone tag, Eliot simply grabbed a couple of 3 ½ inch diskettes and copied the utilities he'd found. He did so quickly, not being concerned about packing the tools onto as few disks as possible. Once done, he'd headed down the hall

the elevator, and up to Jill's floor. He ran into Jill there.

"Jill, I got your message."

"Oh, Eliot! Thanks for coming." Jill had been hurrying past the elevators when Eliot had stepped out. She'd looked as if she was heading to a fire - but then, that was pretty much how things were for her. She provided support for one of the smaller offices, but the employees in this office were an unusual bunch. They liked to download and share all sorts of software they found on the Internet: games, animated cartoons, screensavers, just about everything. Someone had figured out how to disable the anti-virus software whenever they liked, even with the security mechanisms Jill had put in place. So Jill was kept on her toes. In fact, she described it as "herding cats down a beach."

"Well, can you tell me what's up?"

Jill sighed. "I'm not sure. One of the users said something weird was going on with his machine, but I couldn't get any specifics from him. I took a look at the system, but there wasn't anything in the EventLogs. I saw some stuff in the Task Manager, so I made sure the anti-virus was updated and ran a scan, but that didn't turn up anything. I figured I'd call you and see if you could find anything."

"Sure, I'll see what I can do."

Jill led Eliot into the cube farm, where they met and spoke with the user. He was still a little vague, but after several questions, Eliot was able to determine that he'd had trouble opening some documents and e-mail attachments, and even opening Outlook seemed to be taking longer than normal. He'd said that in the past, large documents would take a couple of seconds to load, but there was always a lot of disk activity. Now, the documents opened slower, but there wasn't any accompanying disk activity.

Eliot sat down at the user's desk and put his hands on the keyboard...and stopped. Where to begin? He had his disks, so he put the first one into the drive, opened a command prompt on the system, and changed the prompt to the A:\ drive. He confirmed that the disk had the first utility he wanted, and began to type the first command, pslist. But before he hit the Enter key, a thought occurred to him. How would he get the information he collected back to his workstation? He couldn't simply analyze it here, he had to get the data back to his system to he

could go through it and figure out what it meant. He needed to be able to analyze the data and decide what to do next, and he couldn't do that here, there was too much going on, too many distractions. Besides, he felt he had some time, as nothing had really happened. So far, the user hadn't said anything about files being deleted, or the age-old shenanigans of pop-up messages and the CD-ROM tray opening and closing. Eliot thought at first that he could save the output of the commands he ran in a file, and then either copy the file to a diskette, print it, or copy the file to a mapped drive on his workstation. But he couldn't do that, could he? If he did, what he would be doing would be tantamount to altering a possible "crime scene". By writing the files to the hard drive, he'd be making changes to the system he was investigating.

Eliot sat back and thought for a moment. He then re-ran the 'dir' command and found that he had plenty of space on the diskette itself. So he ran the following command: `a:\pslist > a:\pslist.log`

This saved the output of the command by creating a file on the diskette, rather than the hard drive. Eliot repeated this with each of the utilities he'd copied to the diskettes, and found that he had plenty of space left over on several of them. After all, the output of each command was basically a text file, and text files didn't take up a lot of space. Eliot decided that the chances of any programs on the system had been "trojaned" were pretty slim. He remembered that this happened quite often on compromised Linux systems, particularly when "rootkits" were used. But after all, he was running a command prompt from the system, so he ran several commands that were native to the system. He also decided to see if he could collect more information about network connections and activity. He ran `arp.exe`, `nbtstat.exe` (with the '-S' and '-c' switches to view sessions and the name cache, respectively), and several variations of the `net.exe` command, including "net use", "net sessions", "net file", and "net view". He hoped these "net" commands would reveal some information about network connections on the system.

Once Eliot had collected all the information he could think of, he thanked the user, and went by Jill's office to give her an update. As he suspected, she wasn't around, so he decided to leave her a message on a sticky note and then follow it up with an e-mail once he was back at the office.

When he got back to his office, Eliot went straight to his workstation and copied all of the information he'd collected from the diskettes to a new directory on his system. He started going

through the myriad of information he'd collected, but didn't see anything unusual. He saw a process called "WinWord" for example, in the output from pslist.exe. He looked for the process identifier (PID) in the output of listdlls.exe, and found by looking at the command line that it was indeed Microsoft Word. He'd remembered seeing that application listed on the taskbar. He even went so far as to check the output of fport.exe to be sure that the process wasn't using a network port.

As he was working, Eliot thought that this process of analyzing the data by hand was tedious and time consuming, but he couldn't see any other way. He knew there are [lists of well-known ports](#) available, mapping major and commonly used services to the ports they listened on, awaiting connections. He also knew that there were many, many lists of default Trojan ports that listed the vast array of Trojans and the default ports they used. However, Eliot was also well aware that most, if not all, Trojans provided the user with the ability to change the default port, so trying to track down a particular Trojan by port scanning the system seemed to him to be a great waste of time. Eliot felt that there had to be a better, perhaps more efficient way of handling this, but he hadn't found it until he'd located fport.exe at the FoundStone site. This tool was unique: it's only drawback was that it had to be run locally on the system being examined.

Finally, after spending almost an hour going back and forth between the various output files, he'd finally decided to just print them out. Eliot thought he'd found something. While he couldn't say that this is what was causing the problem, he did find two unusually named processes. The output of listdlls.exe for each showed that the executable files were in the same directory. Eliot opened up a browser, and headed to [Google](#), his favorite search engine. Once there, he typed in the names of one of the processes, and was almost immediately rewarded with a complete page of hits. He read a couple of summaries, and the second one said something about "spyware". Eliot knew that spyware was that little extra code or program that got sent along with some other, legitimate, program in order to collect information about the system or user, and send that information back to a central location. He'd read that marketing firms often used this kind of software.

Interestingly enough, the site was fairly detailed and explained not only what each of the files did, but also that they were part of the package that was downloaded with an ostensibly legitimate program. Eliot quickly confirmed that there was information that referred to the program in the data he'd collected, and decided that he'd figure out what a problem was, if not the problem. He went back and took a look at the information he'd collected regarding the two

processes. The output of pslist.exe showed that they were both pretty busy, and the output of fport.exe showed that one of them was using a network port. He took a look at the output of netstat.exe, and found that the system was connected on that port to a remote system.

Eliot put together a quick e-mail to Jill, with instructions on what to do. He let her know that there really wasn't much to be concerned about other than deleting a couple of files and Registry entries. By the end of the day, Jill returned his e-mail and said that she'd followed his instructions and the user wasn't experiencing the delays opening files any longer.

That evening, while at home, Eliot thought about the events of the day. He felt that he had a reasonably good toolkit put together, but he also felt that he had more work to do. He was glad that the incident hadn't turned out to be anything criminal, as his puttering around might have damaged or even destroyed evidence. Eliot decided that what he needed to do was get his entire toolkit together, and burn it to a CD. That way, he'd have it all together, and it would be protected if a virus had infected the "victim" system. He could also prove that his tools hadn't been affected or modified in any way. In fact, he could go so far as to copy the command interpreter itself (cmd.exe) to the CD, and run his commands from this version. He even considered writing a batch file to automate collecting the information he needed, knowing that by doing so, he sped up the collection process and minimized the chance of making mistakes.

That was great, but what about his tools? He could copy them to the CD, but he was wondering if he was fully prepared...did he have everything he needed? Using the tools he had already found, he had collected a great deal of what he referred to as "volatile data". This was information about the system that existed in memory, and was gone when the system was turned off or rebooted. Volatile data included things like processes, network connections...but what else was there? Had he left anything out? What about the ClipBoard? Eliot found himself copying things to and from the ClipBoard all the time, sometimes copying chat sessions from his [AOL](#) Instant Messenger to be pasted into a Word document or even another chat window.

It occurred to Eliot that he'd used the command prompt quite a bit. In fact, he did that all the time. He felt more comfortable using the command prompt than having to navigate through a maze and a myriad of clicks in the user interface. A fleeting spark skirted across his brain, and he reached for an old copy of an MS-DOS 5.0 command reference he kept around. As he thought, if he typed the command "doskey /history", he'd see a list of commands typed at the

prompt, much like the history file in a Unix shell.

It had been a long day, and Eliot felt he was on the right track. He decided to keep his eyes open for other tools, but what he really needed to do was find a way to get data off of the system he was examining without altering the system itself.

Next Time

Stay tuned for the fifth and final installment of the “No Stone Unturned” series. In the final installment, Eliot completes his toolkit, answers some final questions, and tries to discover the purpose behind a file he found on a system.

To read Part Five of this series, please click [here](#).

Relevant Links

[No Stone Unturned: Part One](#)

H. Carvey, SecurityFocus

[No Stone Unturned, Part Two](#)

H. Carvey, SecurityFocus

[No Stone Unturned, Part Three](#)

H. Carvey, SecurityFocus

[No Stone Unturned, Part Five](#)

H. Carvey, SecurityFocus

[Privacy Statement](#)

Copyright 2006, SecurityFocus