

No Stone Unturned, Part Two

H. Carvey 2002-03-27

No Stone Unturned, Part Two

by H. Carvey

last updated March 27, 2002

Part II

A lone figure sat in front of a computer monitor, silhouetted in its cold, blue glow. The dark, cave-like room hummed with the life of high-powered computer systems and their electrical lifeblood. The figure sat, seeming unmoving for minutes on end. The stillness was occasionally broken with movement as the figure raised a steaming cup to his lips and sipped.

Eliot had spent the early hours of the morning researching something Steve had told him about the other week...something about LastWrite times on Registry keys. He'd gotten better at searching the miasmic labyrinth that is the Microsoft Developer's Network site, and could finally at least get responses back that were in the ballpark of what he was looking for. Searches of the Internet in general revealed little of use, other than the general fact that these values played a role similar to the last modification times on files. From his experience with Unix systems, Eliot immediately saw the usefulness of this information.

After searching and reading for a while, Eliot had stumbled across a Perl script called keytime.pl that actually retrieved the LastWrite times from Registry keys and translated the 64-bit value into something readable. This script had several advantages for Eliot, the first of which was that it was written in Perl, a language he understood. Secondly, it was free. Third, because the script was written in Perl, it laid exactly what he needed to know with regards to the LastWrite time. Finally, the site had a lot of other scripts, many which looked very useful.

In his searching over the past week or so, one thing had become very clear. While there were many sites that talked about how to deal with Linux and the various flavors of Unix in the face of a security incident, there was almost nothing of a similar caliber available regarding the Microsoft operating systems. In fact, Eliot had just about arrived at the conclusion that the default behavior, if any of the public lists were to be believed, for NT administrators was to either ignore the incident, or simply reload the operating system and applications from clean media, without even figuring out what happened. He'd run across a couple of public threads in

which the administrator seemed to have made a copy of the victim system's hard drive prior to reloading the platform from the original CDs. In several cases, Eliot had read posts in which the administrator had suspected something was wrong with a particular NT system, and had port scanned it with nmap. While that sounded reasonable enough, the administrator then proceeded to compare the results of the port scan to a list of default trojan ports. He didn't see how it made much sense to do that, and thought of it as a waste of time. On various Linux systems he'd done basic troubleshooting on, he'd used commands to retrieve information from the system prior to deciding whether or not to unplug it from the network, or even just shut it down. A great deal of information exists in memory on a running system generally referred to as volatile system information, such as running processes and network connections. He'd used commands such as "fuser" or "lsof" to determine the port-to-process mapping, showing the processes that were listening on ports listed in the output of the netstat command. Using known good copies of these tools that had been burned to a CD proved to be an effective approach to incident response, and Eliot didn't see why he couldn't do something similar for NT and 2K.

Deciding to put together a list of tools was one thing. Finding the tools was quite another matter. Many of the tools he'd used on Linux systems had been part of the distribution. Eliot started by compiling a list of functionality he thought he'd need, and then tried to find one or two tools that would provide him with those capabilities. When working with Linux systems, he'd statically-compiled his tools, and then protected them by burning them to a CD. That way, he'd had a set of "known good" tools that he knew for sure hadn't been compromised or trojaned. After all, just about every rootkit that was available for Linux systems did just that. However, NT is an entirely different beast, thought Eliot. First off, there wasn't much in the way of rootkits available for NT systems. There had been one that was being developed, but it didn't seem to be in widespread use. Second, due to the NT architecture, statically compiling tools was next to impossible. Adding the dynamic-link libraries (DLLs) to the CD itself along with the tools didn't do much good, either, because those DLLs that were already loaded in memory couldn't be removed without possibly damaging parts of the system itself, such as the Explorer shell.

Eliot decided that some of the tools he needed were commands that came with the NT distribution itself. The same was true for 2K, as well. For example, one of the first programs he felt he needed to include was the command interpreter, cmd.exe. After all, many of the programs he would need to use were run from the command prompt, and there was always the chance someone would find a way to trojan just about any program on the compromised

system. In a dry run of the first iteration of his toolkit CD, Eliot had found that the cmd.exe from one system didn't work on the other. That meant he'd need either two separate CDs, or two separate folders on the same CD, once he'd tested out all tools he collected. Other programs he'd decided to use from a clean system included netstat.exe and net.exe for network connections, doskey.exe for command history, at.exe for scheduled jobs, and hostname.exe and ipconfig.exe for system configuration information.

Eliot was surprised at the number of useful and freely available tools he'd found on the web. At the FoundStone web site, he'd found fport.exe and the Forensics Toolkit. He was amazed to find how useful fport.exe was, as it provided the process-to-port mapping he'd been searching for, showing which program was using which port to listen for connection attempts. Eliot was also amazed the more sysadmins didn't seem to know about the existence of this program. Many times, typing "fport" at the command prompt would give the administrator certainty where port scanning and comparing the results to a list of default trojan ports tended to only add to the confusion. He'd also found that the SysInternals web site had quite a number of extremely useful tools, many for providing a thorough listing of process information, such as pslist.exe, handle.exe, and listdlls.exe. There were other tools at the site, such as psuptime.exe for determining how long the system had been running.

Suddenly, the ring of his phone shattered Eliot's concentration. Shaken, he looked at his watch and realized it was after 4pm. Geez, he thought. He'd been at this Internet searching and reading for most of the day. The phone was within easy reach, but it took him until the third ring to answer it.

"Hello," he said.

"Eliot? It's Dave."

Dave was an administrator in another office. After the last incident Eliot had dealt with, he'd put out the feelers to other administrators in the company. He'd contacted them about tools, how to respond and deal with incidents, and how to go about tracking distributed attacks. He was surprised to learn how poorly prepared they were to deal with incidents across the company. When Eliot had asked which tools the various administrators preferred to extract and consolidate the EventLogs from critical systems, for example, he'd received some surprising answers. Most of the admins weren't bothering to review the EventLogs, and some had even admitted that they hadn't even enabled auditing on their critical systems. The reason most

often stated was that there was too much information and it was too difficult to understand.

Eliot remembered that Dave administered several NT and 2K systems in one of the offices on the West Coast.

"Hey, Dave, what can I do for you?"

"I think I've got an incident here."

Eliot sat up, suddenly interested. "What's that?"

Dave sighed. "I think I've got an incident on my hands. You seem to have some experience with this sort of thing, so I thought I'd give you a call and see what you thought."

"Okay. So what's up?"

"Well, after those emails about the EventLogs the other week, I figured I'd turn on some auditing and see what happened. I figured I'd start with something easy, like auditing successful and failed logon events. I set this on our PDC, BDC, and a couple of the servers. For the most part, all I've seen so far is when folks here login in the morning, and every now and then someone types their password wrong."

Eliot understood. He'd done the same thing, and was seeing the same things.

"So, Dave. When you said you had an incident, what did you mean?"

"Well, yesterday I saw a ton of failed logins into our PDC, BDC, and one of the other servers. They were all coming from one user in another office that's in our domain."

Eliot knew that the default settings for NT and 2K EventLogs didn't provide for a lot of room for recorded events, particularly when auditing was enabled.

"Dave, did you save these events?"

"No, but I can send you the events. They're still in the logs."

Eliot thought for a moment. "Yeah, why don't you do that. Save all of the entries, and send me

the ones you're worried about."

"Sure thing. I'll get them out to you right away."

"Thanks, Dave."

With that, Eliot hung up. Things were certainly starting to happen. He began to think that a lot of incidents were even recognized simply because the administrators weren't watching. After all, that's what the logs were...the administrator's eyes into the activity of the system. If auditing isn't enabled and events aren't being logged, then how does an administrator decide whether a particular anomaly is a security incident, or simply a misbehaving application?

With that, Eliot decided to call it a day early. He didn't want to wait around for Dave's email, and besides, the logins had failed. Whatever was going on, the user hadn't gained access to the systems. It could wait.

The next day, Eliot got in early and was looking through his email when he saw Cynthia walk by his office. A thought occurred to him, so he jumped out of his seat and bolted to the door.

"Hey, Cynthia!" he shouted, trying to get her attention before she disappeared into some meeting.

"Yes? Eliot! Good morning. How are you?"

"Fine, thanks. Hey, I've got a question for you. I'm waiting on some more information, but an administrator from another office found something that looks like a user from another office tried to access a couple of his servers. Since this involves an employee of the company, who should I tell about this?"

Cynthia thought for a moment. "You said you were waiting for more information. What have you got so far?"

"Well, not much. I was just checking my email to see if what I was waiting on had arrived yet."

"What geographic area are we talking about," Cynthia asked?

"West Coast."

“When you know more, let me know. If it looks like it’s something that needs to be addressed through HR, we can contact the rep for that area, and have them contact the employee’s manager. But be sure to let me know...if it ends up being something serious, we’ll definitely need HR involved, and we might have to contact legal counsel for the company.”

“Okay, thanks. Will do.”

With that, Eliot went back to reviewing his email inbox. He found the email from Dave he’d been looking for and opened it up.

After reading the email twice, Eliot was confused. Dave had said that he’d seen “a ton of failed logins”, but there were only three listed in the email. One of the failed logins was from the EventLogs from the PDC, the other two from the BDC. Eliot figured that either Dave had misunderstood what he’d asked, or had been in a hurry and provided only a representative sampling of the failed logins. However, Eliot felt that the only way to really be able to speak from a position of authority in this issue, he had to have all of the available information, so he put this in an email to Dave, asking for clarification.

Later in the day, Eliot received an email response from Dave. The email said, quite simply, “that’s all there is.” This didn’t seem to be right. First, Dave had sounded pretty emphatic when he’d said there’d been “a ton of failed logins”. Second, one or two failed logins didn’t seem to be indicative of an attack. In fact, it didn’t even seem to indicate a bored user who was just playing around on the network. Just to be sure, Eliot decided to send out a quick email asking other administrators if they’d seen similar activity. Also, he figured he’d have to call Dave directly to clear up the issue of how many failed logins were actually available, and how many systems were affected.

Next Time

In the third installment of “No Stone Unturned”, Eliot gets to the bottom of this unusual incident, finds some more tools, handles another incident, and learns a thing or two along the way. Stay tuned.

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

Relevant Links

[No Stone Unturned: Part One](#)

H. Carvey, SecurityFocus

[No Stone Unturned: Part Three](#)

H. Carvey, SecurityFocus

[Privacy Statement](#)

Copyright 2006, SecurityFocus