BOOK REVIEWS

Jigang Liu
Editor
Metropolitan State University
St. Paul, MN 55106

If you have any suggestions on books for review, or you would like to write a book review for us, or you have any comments and concerns on the book reviews published on this column, please feel free to send an email to Diane Barrett, the new editor for this column, at dm_barrett@msn.com.

BOOK REVIEW

Anson, S., Bunting, S., Johnson, R., & Pearson, S. (2012). *Mastering Windows Network Forensics and Investigation*, 2nd ed. Indianapolis: John Wiley & Sons. 674 pages, ISBN: 978-1-118-16382-5, $59.99 USA/$71.99 CAN.

Reviewed by **John C. Ebert**, Metropolitan State University (eberjo@metrostate.edu)

The book is available as a paperback and e-book. The e-book versions allow you to preview several chapters at any of a number of online vendors. The e-book prices vary from the same as the soft cover version ($59.99) to about $38.99. Some of the vendor's e-books retain the color illustrations found in the print version, but others produce them in grey scale, so you might want to look out for that. The book is divided into four parts (17 chapters) plus two appendices.

I am compelled to give the book illustrations a *highly unfavorable* assessment regarding their readability qualities. Their content is otherwise fine and meaningful. Time and again the illustrations are so miniscule that even those of us with the best of vision will be seriously challenged. I hold out the recommendation to review your e-book options over the hardcopy edition. At least in an e-book you have the hope of doing a page-zoom. Review your e-book providers carefully; some I reviewed had "fuzzy graphics." Notwithstanding the problem illustrations, I still recommend this book for its in-the-trenches information and the desktop reference it will become.

This book is strongly Microsoft server centric by choice, with networking limited to discussion on relating processes and process identifiers to established port connections, and Server Message Block (SMB) network authentication interception [SMB is Microsoft's naming for the Common Internet File System (CIFS).] If you are seeking deeper networking forensics, this is not its primary focus. All of the chapters end with *The Bottom Line* summary that poses *Master It* challenge questions. Appendix A replicates the summaries with the addition of *Solutions* to the challenge questions. Appendix B includes test lab configuration information and a table of software references discussed in the chapters. The same detail on Registry paths in the chapters is not included in a companion table, and you will have to make extensive notes to build your own reference table.

Part 1 of the book is titled *Understanding and Exploiting Windows Networks*. Chapter 1 *Network Investigation Overview* is about the detective side of a digital investigation. The military, teaching, and law enforcement backgrounds of the four authors make this an incident and digital crime scene investigation work from the prosecutorial perspective more than an e-legal e-discovery outlook. They discuss ways on how to conduct opening interviews during an investigation, how to lead, how to set expectations, how to inject advice at timely moments, and how to not over-promise what you cannot deliver. In these ways, the book goes beyond only training on how to find, preserve, and collect evidence.

Chapter 2 *The Microsoft Network Architecture* would better be entitled *Microsoft Server Infrastructure and Administration* or simply *Microsoft Architecture* sans the word "Network" in the chapter title. The authors clearly want you to know this book is written for investigators, and not administrators. That said, you soon find yourself wondering if it is not the other way around, but don't panic; they generally keep the depth of the pool shallow enough so that non-administrator investigator's toes can still touch bottom. There are times when you may encounter information requiring more than one read-through. If you are a forensic investigator already, you may still find some new information. If you are new to this, having an experienced forensic friend to bounce questions off of could be very helpful.

For those with some Microsoft Server and Active Directory exposure, this chapter is a smooth and breezy review of domains, trees, forests, transitive trust relationships, organizational units, groups, users and computers, NTFS file permissions, and share permissions of folders on the network. The authors have done a good job of being concise, going straight to the heart of matters. All this important detailing and delving may be alien spaces for some readers, but we learn exactly why, as investigators, we need to know who has access to contested computing resources, who does not, and how we know it. They explain why being clear on who has rights and permissions to Windows networked computing resources is important to the scope of your investigation as well as your credibility in court.

The chapter ends with a walkthrough using Metasploit Framework version 4.0.1. We learn how to attack and own a remote connection into an un-patched victim Windows 2008 Server on the network segment from an attacker machine. When we are done, we are in a command line window at the *C:\Windows\system32* directory and authorized to issue any command in the System context of that folder. Figure 2.20 on page 58 is missing the command line because the incorrect screen image was published. You might consider it an opportunity to solve what the blank command line should have been!

Chapter 3 *Beyond the Windows GUI* digs into Windows vulnerabilities at the operating system level. Discussion examines what Dynamic Link Libraries (DLLs) are, what DLL code injection is, and how an Import Address Table (IAT) is overwritten in order to "hook" the exploited process thread and re-direct it to execute a function(s) in the injected code. We are taught that hijacking the threads' normal chain of activities is how many exploits succeed to avoid detection because they run in the context of an already vetted system process. The discussion examines the Intel CPU hardware architecture design concept of security rings 0 to 3, with 0 being assigned to kernel system privileges and 3 being the ring where all user level activities run. The book explains why hardware privileges and partitioning are the core physical layer underlying the security operations of the software operating system kernel and user layers. We learn how device drivers that are installed using Administrator privileges into the system where they run at the kernel level are dangerous opportunities to redirect process flow and gain a permanent foothold into the system.

Chapter 4 *Windows Password Issues* gets a bit heavy in the first 25 pages, as we enter the dark forest of "Exploring Windows Authentication Mechanisms." The value of this is that once you get through this part, you will have a deeper understanding of how hard (or easy) it may be to capture a users' local machine or domain credentials. The final 24 pages of the chapter of fun hacking and cracking exercises are your reward for suffering the first 25 pages, and you will appreciate when, where, why, and how authentication and weaknesses can be attacked, and how juicy the second part exploits really are. Step 5 on page 131 shows how to use extracted hash credentials (pass-the-hash) to impersonate a user without even knowing their password. Step 5 could also show how using the -*w* option will dump the passwords in clear-text instead of the hash.

Chapter 5 *Windows Ports and Services* is a refresher on ports for those who have studied them. We learn the *netstat* and *tasklist* commands, and a few optional switches that can be used with them. The significant value in this chapter is that the authors bring home the point that to be an effective analyst it is not enough to collect information, but you must learn to correlate different sources to form a story. They show us how to correlate network port process identifiers (PIDs) with the image name (program) responsible for running that process. Since correlation is a foundation to analysis skills, this is a simple but core element to acquire that the chapter provides. The chapter takes it a little further and introduces us to the *svchost* process and relates how it is a common system resource that hosts executing DLLs discussed in chapter 3. When the svchost process hosts a DLL, it becomes a service with a path to a physical program resource. We are introduced to the Windows Registry and shown how to find that physical location on the storage device of the *svchost-ed* DLL running in memory

Chapter 6 begins Part 2 of the book, which is titled *Analyzing the Computer*. Here we find *Live-Analysis Techniques*, a chapter that looks at finding live evidence in memory. The authors explain why and when there are times evidence should be collected from random-access memory (RAM) before pulling the power plug.

The chapter is a winner because it introduces Moonsols Dumpit RAM imaging utility and the RAM dump analysis capability of the Volatility Project. While Moonsols' download link is in the book, it's easier to key-in *http://www.moonsols.com/ressources/*.

The chapter explains three examples of feeding a RAM dump into the Volatility Project tool to extract and analyze the live state of the system at the time of acquisition. The Volatility tool can be run in pre-compiled stand-alone version hosted under a Python runtime shell in Windows, or executed as a call to the *volatility* script from an installed Python interpreter on Windows software available at *http://python.org* (not mentioned in the book). I felt that the authors could have provided more explanation of Volatility's structure as a text or script based "main module" that calls plugins (modules) so that the reader would more clearly comprehend it as a framework sort of tool, along the lines of Metasploit Framework. The book provides a TIP box to basic usage of Volatility, which is essential, but the link is no longer valid; instead, try *http://code.google.com/p/volatility/wiki/VolatilityUsage22*. A reference link to video tutorials on Volatility, case studies, forensic challenges, and exploits at *http://code.google.com/p/volatility/wiki/VolatilityDocumentationProject* could improve the hands-on learning character of the next edition.

Back in chapter 4, Windows Credential Editor (WCE) was briefly described but hides what is going on behind the scenes with the Security Account Manager (SAM) and SYSTEM registry during acquisition. The authors can improve this chapter (or roll it into chapter 4 on Windows Password Issues or the Registry chapter) by adding in how to use Volatility to pull password hashes from memory using the *imageinfo, hivelist*, and *hashdump* modules. An example of this can be found in Daniel Dieterle's exploit blog at *http://code.google.com/p/volatility/wiki/VolatilityDocumentationProject.*

The chapter concludes with an introduction to wire-line digital sniffing basics. Discussion on using an Open Systems Interconnection (OSI) layer 1 hub could include information on using Cisco Switched Port Analyzer (SPAN) ports on a switch, as well as mentioning true network taps that are designed for this purpose. An example screen capture of a username and password sniff session extraction from an insecure HTTP URL encoded form would be an impressive addition. The chapter ends on using Zenmap to actively scan a network for machines, ports, and operating system (OS) fingerprinting although no mention was made of the command line version of *nmap* that is also in the installation folder. The tool

Network-Miner is a similar tool not mentioned.

Chapter 7 *Windows Filesystems* covers all the bases on interpreting the File Allocation Table (FAT) file system spaces of allocated, unallocated, slack space, directory structure, and cluster runs. We learn why FAT is a forensic analyst's friend for portability of tools. We move our way around the FAT file system in screen captures from the Encase disk view editor. In this chapter, the screen captures are just large enough to be legible and useful. A nice nugget of gold is found in how to use *grep* to search for file fragments when the dot-double-dot subdirectory parent entry may have been damaged. The example grep search string on page 197 shows searching for any deleted entries for all *.exe* files. The authors point out that Encase can automate this broken dot-double-dot directory problem recovery process for the user but it was good to have it explained how to do it manually.

Moving on to the New Technology File System (NTFS) file-system, the process of what is what and where it is gets a bit more convoluted, and the authors provide information to follow along in a hands-on practical lab to exercise skills at manually creating, deleting, and recovering a file using *Disk Explorer for NTFS* (*http://www.runtime.org*). The chapter examines the short subject of Alternate Data Streams (ADS), how to make them, and how they can be used to store metadata or hidden information. The streams discussion does not inform us that Windows 7 updated the *dir* command, adding a new */R* option to reveal ADS streams, nor that not all ADS streams are evil, only suspect until proven otherwise. An example of this is that Internet Explorer attaches the *:Zone.Identifier* stream suffix to every file downloaded; thus, if you download the *putty.exe* program using IE, the command *dir /R* will return *putty.exe:Zone.Identifier:$DATA*. The chapter introduces the *streams* tool from the Sysinternals section of *http://technet.microsoft.com* to locate and examine ADS. Another tool not mentioned, but a candidate for doing so, is StreamArmor from *http://securityxploded.com/streamarmor.php*. The chapter finishes with a concise two-page overview of FAT32.

Chapter 8 *The Registry Structure* is an excellent introductory foundation of the Windows registry. There are a few nuggets of information (NTUSER.DAT, UsrClass.dat) on page 232 you may wish to sweep with your yellow highlight marker, and a TIP box on REGSHOT to pay attention to. Chapter 8 bogs down 19 pages later when it delves too deeply into Encase for 12 pages, complete with nine pages of screen captures (pages 236-245) that are too small to read without a robust magnifying glass. We recovered illustration legibility in the follow-on area, with AccessData *Registry Viewer*. The Registry Viewer is a free tool and the authors give a good five-page examination on using it.

The User Assist key and ROT13 discussion shows up on page 249, another yellow marker opportunity. Intelliforms is on the same page. All the Registry juicy bits in the chapters not arranged in an Appendix C table was a missed opportunity to elevate the book's quality.

The chapter finishes with a too short treatment of Harlan Carvey's *Reg Ripper* based on PERL scripting. The authors could have expanded greatly on how to use the tool, not just give it a cursory "[…it] is very good at what it does."

Chapter 9 *Registry Evidence* is one of the chapters that justifies owning the book. If you are compiling a table of notes about Registry paths, this is where your pencil will get a real work-out. It will expand many readers' understanding of how to extract registry hives from restore points, what is the relationship of the Volume Shadow Copy to restore points, how to extract a file from one, how to review the Software key for evidence of installed or uninstalled applications, how to determine who last logged on, locate and inspect the logon banner to inspect for tampering, and where the keys are to examine Windows Security, Action Center, and Firewall notifications for intrusions, and why an attacker will go for these first. The chapter continues on showing us more how-to for the Recycle Bin, the ProfileList Key, info on the Protected Storage Service Provider (PSSP), Intelliforms, and using Nirsoft's tools to recover stored website forms data and passwords. We continue on to the Most Recently Used (MRU) Key, Recent Docs, TypedURLS, UserASSIST (and what it is), LSA Secrets versus Cain & Abel, and discovering historical Internet Protocol (IP) addresses used by an interface. We find the TimeZone Offset in the registry and follow-on to looking at quite an array of possible Start-Up locations for malware to hide. This deep chapter finishes by looking at Scheduled Tasks and using the Autoruns tool to examine Startups. If you are compiling a summary table, you should be tired of writing by now. I did not find any information on using an undocumented switch in regedit.exe to run it silently; the model is *regedit.exe /S yourfilename.reg* and suppresses noisy dialog boxes.

Chapter 10 *Introduction to Malware* first finds us using the Sysinternals command-line *strings* utility to extract text objects from the interior of *exe* files. The authors introduce the GUI BinText and Dependency Walker for similar analysis objectives. The chapter moves on to Process Monitor, Process Explorer, Wireshark, and a brief reference to Netwitness and Splunk. We conclude with discussion on setting up a live analysis multi-machine virtual sandbox environment for capturing live network centric malware activity such as one might encounter emanating from a rootkit zombified computer.

Chapters 11 to 15 comprise Part 3 of the book – titled *Analyzing the Logs* – are about logs, logs, logs, and logs. These five chapters are another reason to own this book; the practical information presented here on File Transfer Protocol (FTP), Dynamic Host Configuration Protocol (DHCP), Firewall, and Windows Application, System, and Security log analysis should be in every analyst's experience kit. In chapter 11, the authors introduce how Splunk 4.2 (at v5.0.1 as of this review) can be used to reduce vast quantities of log data to meaningful reports and timelines. We read about how to use Splunk to filter Internet Information Services (IIS) Web server log events and identifiers of interest over time, and visualize activity that would otherwise overwhelm the process. If we wanted to explore further, once we get savvy using Splunk (which can be quite a bit disorienting at first) we could just as well Splunk system reboots, firewall logs, Remote Desktop Protocol (RDP) events, logon/logoff, etcetera. I did not try the hands-on IIS example in this chapter in my Splunk 5.0.1 version, but believe this particular lab will succeed without complication because it uses well-known source type parsing filters to read well-known log structures. The same cannot be expected for irregular or unknown log types, and they may need conversion to a Comma Separated Value (CSV) file and parsed that way if you encounter problems. Unfortunately, the book does not offer that helpful "what if" suggestion. Two sessions on Splunk are also examined in chapter 14 and 16 where you may need this skill when adding dead event logs exported from some other source server. Note also that Splunk may not install any Start/Programs group in your menu list in Windows; it is accessed using your web browser locally on *http://localhost:8000* or *http://127.0.0.1:8000*, or your local machine name such as *http://4n6labrat:8000*. The book informs us to simply click the icon Splunk installed on our desktop; what if we did not choose that option during install? The book needs a revision to include what to do when the desktop icon is not present. We also learn a great deal about correlating Logon IDs, Handle IDs, and Process IDs with Event IDs to build evidentiary facts pointing directly to who, what, where, when, and why.

Chapter 15 *Forensic Analysis of Event Logs* is gritty and down to the hex editor level. We are shown how to parse the structure and meanings of Windows Event Logfile headers, chunk headers, and event records in the file, and how to locate and reconstruct a log. The authors provide us with the hex string sequences in a table for reference.

Part 4 is titled *Results, the Cloud, and Virtualization*. Chapter 16 *Presenting the Results* is an easy read providing a nice framework for report production that will be and look professional, as well as functional. We are introduced to CaseMap and TimeMap software modules, part of the CaseSoft suite of tools. The chapter ends with an example of report generation using Splunk. The Splunk learning curve can be a bit confusing and geeky (if that is adequately illuminating) but as Darren Kitchen of Hak5 might say, "once you get your Splunk on," you will have a powerful new analysis tool in your kit for building timelines and drilling down through massive datasets. Note that in the book, the authors use Splunk 4.2 and it functions differently enough from Splunk 5.0.1 to cause problems in executing the authors' lab on parsing a *security.evtx* event log. The problem appears to be that 5.0.1 does not have a built-in source type parsing filter on inbound *.evtx* files as it must have had in 4.2. To work around this issue, open any *.evtx* log file in

Windows Event Viewer (eventvwr.msc) first and "Save All Events As" a CSV file. Instead of adding the native *.evtx* file to Splunk as a data source, add the *.csv* file instead because Splunk does have a *.csv* source type that will handle this properly. The book needs to be updated to address this issue.

Since we went so deep into how to sleuth through NTFS file structure as well as recover deleted logfiles, the next edition would be well served to show us how to do more with Splunk since it is such a powerful analysis tool. It would help to know that when we "Add Data" to Splunk, it goes into the "Main" database, and the "Admin" "Role" "Reads" this index by default, *and no other*. It would be instructive of the next revision to show us how to use the "Manager" to create our own "Indexes" "New" database container to "Add Data" to, followed by how to use "Manager" to modify *Users and Authentication*" Access Controls>>Roles>>Admin" *Indexes searched by default* so that the Admin role includes our database in the Search process when we run one. Hopefully, this review provides you some Splunk breadcrumbs to follow until then.

Lastly, a Splunk section showing how to use "Manager" to construct a "Searches and Reports" "Saved" search for some term or "string of interest" present as an element in our private indexed database would be powerful. Acquiring this skill can help you drill down to analyzing the percentage distribution of RDP intruder IP addresses over a timeline as an example.

At Last. Chapter 17 *The Challenges of Cloud Computing and Virtualization*. What is it? What is a hypervisor, and what is a type 1 or type 2, or does it matter? What is a VM and how do we capture live RAM data from it? What if we have no physical port to access? Then what? Will a Snapshot do it for us? Can we use traditional examination tools to view a "dead-guest" virtual disk file? What about a virtual memory file? How do we know which virtual snapshot to examine and how many there are? Is it possible to image a virtual machine and convert it into an actual non-virtual hard-drive image that will boot in a real physical computer? All these questions and more are discussed in Chapter 17 and ends introducing us to *SQLite Database Browser* to open and view the contents of the Dropbox cloud application database, and may be useful in securing that court order to access the server side of the cloud provider or win your e-discovery case.