

Mandiant Red Curtain: Malware identification for incident responders

By Russ McRee

Prerequisites

Windows XP or higher console use
 Windows 2000 or higher for agent use
 Microsoft .NET 2.0 framework for the MRC console¹
 PsTools for remote agent deployment²
 Helix for a trusted toolkit³

Similar Projects

RAPIER 3.2⁴
 CWSandbox⁵
 Norman Malware Sandbox Analyzer⁶

MANDIANT Red Curtain⁷ (MRC) is free software for incident responders that takes analysis of malware to a different level; beyond expected norms you might say. MRC examines executable files to establish how suspicious they are based on a set of criteria, including review of multiple aspects of an executable, for things such as entropy (more on this below), indications of packing, compiler and packing signatures, the presence of digital signatures, and other characteristics used to generate a threat “score.” This score is then used to determine whether files are worthy of further investigation.⁸

MRC includes an analysis engine and a data presentation layer. The engine reads in the file to be analyzed, using the data within the file to calculate the Shannon Entropy across a series of overlapping windows of file segments. The engine also reviews additional aspects of the file, such as the permissions associated with various sections of the file, and whether or not the file has a valid, trusted digital signature applied to it. This data is then fed to the presentation layer, which organizes it for display to the user. It also takes the various

elements of analysis generated by the analysis engine and calculates an aggregate threat score based on that data.

MRC’s road map includes incremental improvements such as refining the threat scoring algorithms, usability improvements, and perhaps the potential for expanding the criteria investigated. Mandiant is happy to receive community feedback to help set the direction of their next release.

MRC techniques will find a home in a pending commercial product, as part of the IR/acquisition features in Mandiant Intelligent Response.

MRC can be used under a free license, but there are restrictions on reverse engineering or extending the product and resale. Mandiant has considered some open source models for some of their work, but have not yet taken that direction.⁹

Entropy

Analysis of entropy, the measure of disorder and randomness, as it relates to malware, is a focus seemingly unique to MRC. Malware often makes use of encrypted, compressed, or obfuscated (depending on the method of obfuscation) data, and as such, its entropy tends to be higher than that of “structured” data, such as user-generated documents and well known computer programs. MRC approaches the identification of these attributes as follows, using Shannon Entropy:¹⁰

1. A file is opened and the bytes read in to calculate a global entropy value for the entire file.
2. MRC then divides the file into overlapping samples and calculates the entropy across them. For arguments sake, assume a file of size X is divided into n samples of size Y .
3. The mean and standard deviation of all entropy values from all samples is calculated. The overall entropy for the input file is derived by taking the mean and adding one standard deviation to it. This value is referred to as the Sample Source Entropy.
4. Sample Source Entropy and Global Entropy are compared to a threshold. This threshold is an empirically

1 www.microsoft.com/downloads/details.aspx?familyid=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en

2 www.microsoft.com/technet/sysinternals/Utilities/PsTools.mspx

3 <http://e-fense.com/helix>

4 <http://code.google.com/p/rapier/> Free

5 www.cwsandbox.org/ Varied licensing

6 www.norman.com/Product/Sandbox-products/en-us Commercial

7 www.mandiant.com/mrc

8 www.mandiant.com/mrc

9 Dave Merkel, MANDIANT

10 C.E. Shannon, “A Mathematical Theory of Communication,” <http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf>.

derived value between 0 and 1. If either entropy value is greater than the threshold, the data block is determined to be entropic, and therefore potentially interesting.¹¹

“That’s really interesting, but what the %#*& are you talking about, Russ?” Let me simplify. Entropic theory is applicable across many scientific practices, not just computer science, and is best explained by Dr. Thomas Schneider in “Information Is Not Entropy, Information Is Not Uncertainty!” as it pertains to biology.

“Shannon called his measure not only the entropy but also the “uncertainty.” I prefer this term because it does not have physical units associated with it. If you correlate information with uncertainty, then you get into deep trouble. Suppose that *information ~ uncertainty*, but since they have almost identical formulae, *uncertainty ~ physical entropy*, so *information ~ physical entropy*. BUT as a system gets more random, its entropy goes up, *randomness ~ physical entropy*, so *information ~ physical randomness*.

How could that be? Information is the very opposite of randomness! The confusion comes from neglecting to do a subtraction: **Information is always a measure of the decrease of uncertainty at a receiver (or molecular machine).**¹²

If you subscribe to the claim that “information is always a measure of the decrease of uncertainty,” you will not only grasp the concept that drives MRC’s methodology, but one of the underlying fundamentals of malware research, or for all intents and purposes, incident handling in general. Eliminate uncertainty and you will be more readily able to build an effective response.

Installation

MRC installation is point and click so long as .NET 2.0 is already installed. If you do not have it on board, the installer will assist in its installation.

Use

I typically use MRC in two situations. The first is as part of my live response toolkit for analysis of suspect hosts. The second is as part of my malware research sandbox, installed on Windows virtual machines destined for intentional infection with a variety of malware. As with any malware analysis under sandbox conditions, ensure you are operating in confirmed isolation where you will do no harm to production or critical systems.

If reviewing live suspect hosts, there are some recommended steps to include as part of your procedure. If we assume prescribed methodology remember your goals include steps to:

- Identify and analyze
- Contain

- Eradicate
- Recover
- Prevent

Incident handlers are not likely to benefit from the same time allotment that may be afforded forensic investigators, given that information must be acquired quickly in order to establish an enterprise response. Tools like MRC provide ample assistance in that endeavor.

MRC can be used directly on the suspect host, but remember the .NET 2.0 framework must be installed.

Assuming you have the appropriate permissions to do so, I suggest running the MRC agent on the suspect host remotely, and analyzing its output on your workstation. Building the agent package is very simple. *File -> New -> Deploy Scanning Agent* will prepare the files you need to copy to the suspect host you are investigating.

A quick tip to consider as part of your incident response repertoire: *only rely on trusted tools*. If a system has been compromised, what guarantees do you have that it has not been rooted or that common system executables have not been replaced? This is most easily overcome via reliance on a trusted toolkit like the Helix distribution.

To deploy and execute the scanning with a trusted `cmd.exe` from your Helix distribution, make use of PsExec from SysInternals and do as follows:

1. Create scanning agent files with MRC.
2. Copy scanning agent files to suspect host.
3. Share your local CD drive as *cdrom*.
4. `psexec -u <admin acct> -p <password> \\<victim host ip> net use x: \\ <localhost ip>\cdrom`
5. `psexec -w x: \IR\xp -u <admin acct> -p <password> \\<victim host ip> x: \IR\xp\cmd.exe`
6. Now on victim host, issue `MRCagent.exe epcompilersigs.dat eppackersigs.dat roamingsigs -r c:\windows output.xml`
7. Copy `output.xml` back to your workstation and open `output.xml` in the MRC console.

I make an assumption in my execution of MRCagent, specifically the likely location of a malicious file on a suspect Windows XP host. Scan C:\WINDOWS if you want to cover the vast majority of probable locations and not risk missing anything. But you will note that the scan time is a lot longer than if you specified just C:\WINDOWS\system32 or C:\WINDOWS\system (common playgrounds for evil).

Analysis

Here is where the benefits of MRC really come to play. The first example shows an immediate and obvious response from MRC, where the findings are clearly delineated by a high entropy score for `wkssvc.exe`. Thanks to instant gratification

¹¹ “The Entropy of Evil,” Mandiant Red Curtain User Guide, section 2.2.

¹² Dr. Thomas D. Schneider, “Information Is Not Entropy, Information Is Not Uncertainty!” <http://www.lecb.ncifcrf.gov/~toms/information.is.not.uncertainty.html>.

Score	File	Size	Entry Point Signature	Entropy	Code Entropy	Anomaly Count	Signed	Details
4.025	c:\windows\wkssvc.exe	132216		0.982	0.982	1	<input checked="" type="checkbox"/>	Details
3.256	c:\windows\system32\drivers\STEALTH4.sys	12960		0.993	0.993	0	<input type="checkbox"/>	Details
3.256	c:\windows\system32\drivers\AES256.sys	18464		1.064	1.064	0	<input type="checkbox"/>	Details
3.256	c:\windows\system32\drivers\RIUN256.sys	21856		1.064	1.064	0	<input type="checkbox"/>	Details
3.256	c:\windows\system32\drivers\AES128.sys	18464		1.064	1.064	0	<input type="checkbox"/>	Details
2.075	c:\windows\system32\drivers\fsndres.sys	2239		0.120	0.000	1	<input type="checkbox"/>	Details
2.000	c:\windows\installer\{90110409-6000-11D3-8CFE-0150048383C9}\wordicon.exe	286720		0.813	0.005	0	<input type="checkbox"/>	Details

Figure 1 – wkssvc.exe stands out

from MRC (see Figure 1), I grabbed wkssvc.exe out of C:\WINDOWS, fed it to Virustotal, and quickly determined that the suspect host had an SDBot variant onboard.

ometimes, the results from MRC output may not be as obvious as those seen in Figure 1, but paying close attention to details will still provide you with invaluable feedback if properly interpreted. Consider Figure 2.

A pretty red alert with a high score did not pop right to the top of my console, let alone a yellow, medium score. In fact, when sorting by score, nothing of interest presented itself. But sorting by Anomaly Count painted a different picture. Winadll.exe, found in C:\WINDOWS\system32 showed three anomalies, including checksum_is_zero, contains_eof_data, and incorrect_imageSize. Aha...clue number one. The Entry Point Signature also refers to Borland Delphi as opposed to MS Visual C++...clue number two. Finally, checking MAC times on winadll.exe (and the fact that I know it has no business in system32) led me to a super-sleuth conclusion...winadll.exe must be malware! As I prove on a daily basis, you do NOT need to be a rocket scientist to go bug hunting and find your quarry. Virustotal confirmed my magical powers of deduction and advised that I was the proud owner of approximately the 16,224th variant of Gao-bot. In all seriousness, MRC directly contributed to identifying winadll.exe as worthy of further investigation and

likely halved my response and analysis time in this particular investigation.

Benefits and drawbacks

Toolsmith is not prone to bringing you discussion of tools with any real drawbacks (this stuff is meant to be useful, after all), and MRC is no exception. The addition of software that readily aids in the identification of malware can only be seen as beneficial.

The prospect of extended functionality in pending releases nullifies any perceived drawbacks as far as I am concerned.

In conclusion

Remember our discussion of trusted tools, and conduct malware analysis in an isolated environment as often as possible. Additionally, enhance, refine, and practice with your incident response toolkit. Your real response will be all the more successful when the time comes. The addition of MRC to your toolkit will further guarantee that success.

Cheers...until next month.

Acknowledgments

Dave Merkel and Anne Mroczynski of MANDIANT for valuable information, feedback, and assistance in preparing this content.

Score	File	Size	Entry Point Signature	Entropy	Code Entropy	Anomaly
0.818	C:\WINDOWS\system32\winadll.exe	184320	Borland Delphi v6.0 - v7.0	0.993	0.818	3
0.987	C:\WINDOWS\system32\dlcache\Up4avel.dll	87257	Microsoft Visual C++ v6.0...	0.853	0.821	2
0.150	C:\WINDOWS\system32\dlcache\shhtml.exe	16437	Microsoft Visual C++ v5.0...	0.606	0.606	2
0.135	C:\WINDOWS\system32\dlcache\lscptext.exe					2
0.129	C:\WINDOWS\system32\dlcache\lclgwiz.exe					2
0.187	C:\WINDOWS\system32\dlcache\lpreadvC.dll					2
0.148	C:\WINDOWS\system32\dlcache\lpreadm.exe					2
0.150	C:\WINDOWS\system32\dlcache\ladmin.exe					2
0.149	C:\WINDOWS\system32\dlcache\lpadmcsi.exe					2
0.150	C:\WINDOWS\system32\dlcache\lauthor.exe					2
0.092	C:\WINDOWS\system32\dlcache\lvrse.dll					1
0.279	C:\WINDOWS\system32\dlcache\lodev32.dll					1
0.279	C:\WINDOWS\system32\dlcache\lodfox32.dll					1
0.279	C:\WINDOWS\system32\dlcache\lodbse32.dll					1
0.091	C:\WINDOWS\system32\dlcache\lchitkid.dll					1
0.279	C:\WINDOWS\system32\dlcache\lodbse32.dll					1
0.092	C:\WINDOWS\system32\dlcache\lmsrep40.dll					1
0.279	C:\WINDOWS\system32\dlcache\lodev32.dll					1
0.089	C:\WINDOWS\system32\dlcache\lup4amsit.dll					1
0.095	C:\WINDOWS\system32\dlcache\lwinadnoe.dll					1
0.089	C:\WINDOWS\system32\dlcache\lntfc40.dll					1
0.095	C:\WINDOWS\system32\dlcache\lzmnetm.dll					1
0.091	C:\WINDOWS\system32\dlcache\lmpicuc.dll					1
0.089	C:\WINDOWS\system32\dlcache\lmsdm.ocx					1
0.112	C:\WINDOWS\system32\dlcache\lscrlntm.exe	36937	Microsoft Visual C++ v5.0...	0.800	0.800	1
0.089	C:\WINDOWS\system32\dlcache\lntfc42.dll	995383	Microsoft Visual C++ v6.0...	0.849	0.825	1

About the Author

Russ McRee, GCIH, GCFE, CISSP, is a security analyst working in the Seattle area. As an advocate of a holistic approach to information security, Russ's website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.

Figure 2 – winadll.exe, less obvious, but no less evil