MIR-ROR: Motile Incident Response – Respond Objectively, Remediate

By Russ McRee – ISSA member, Puget Sound (Seattle), USA Chapter

Prerequisites

Windows XP/2003/Vista/2008
Windows Systinternals tools
Windows Server 2003 Resource Kit
Seccheck.exe

Similar Projects

RAPIER[1]

I learned long ago from Steve Mancini and Joe Schwendt of the RAPIER (Rapid Assessment & Potential Incident Examination Report) project, you can't publish a cool tool without a cool name.

To that end, I am proud to present **MIR-ROR: Motile Incident Response – Respond Objectively, Remediate**. If that doesn't qualify me as an über-dork (like that needed qualification), nothing will. ;-)

I was rooting about all my USB fobs and discovered one I received while at LE Tech last year.[2] Hiding therein was a handy script that Microsoft forensics mastermind Troy Larson[3] had written to gather investigative data from target machines using a USB stick. I reached out to Troy, and he graciously agreed to allow me to brand the script, as well as maintain and optimize it for your use during incident response engagements.

I consider MIR-ROR a specialized, command-line, RAPIER-like script that makes use of the all-important Windows Sysinternals tools, as well as some other useful tools. Further, as you will see, you can easily enhance the script to your liking with whatever command line tool tickles your fancy.

In fact, Michael Panico,[4] my team lead, upon cessation of heckling me for the cheesy script title, made an excellent suggestion: tune or isolate script features for mission specifics, such as malware investigations versus miscreant user investigations. While I will take this on as part of the ongoing development process, I won't in the context of this column.

Download MIR-ROR,[5] as discussed here, and keep an eye on the site for mission specific instances as mentioned above.

This is a fundamentally simple script capable of great things. As an incident response resource, we've found it indispensable. I look forwarding to hearing back from you with regard to how well it's served you.

Preparing for MIR-ROR

Windows Systinternals licensing prevents me from bundling the tools in a distribution package; thus you'll have to retrieve them.

I suggest simply downloading the complete Sysinternals Suite[6] and unpacking to a preferred directory on your system, then move the necessary tools listed in *fetch.txt* to a directory you create: `C:\tools\MIR-ROR`. You'll need to grab *seccheck.exe* from my site.[7]

You'll also need *now.exe* from the Windows 2003 Resource Kit.[8] As you did with the Sysinternals Suite, download the Resource Kit to a preferred directory, and then copy *now.exe* to `C:\tools\MIR-ROR`.

Keep in mind that there are obvious system nuances depending on the system you run MIR-ROR on. I'm a huge fan of *seccheck.exe* from MyNetWatchman; it works well on Windows XP and Windows Server 2003 but fails on Vista and Windows 7, even when run as administrator. I've added a couple of tools specifically from the Win2k3 ResKit that will obviously fail elsewhere; you can opt to comment them out in the script if you wish. Remember, this is a tool you can experiment with and modify to your liking for your environments.

Once you've gathered all the appropriate elements, your MIR-ROR directory should appear as Figure 1.

I've left *sigcheck.exe* disabled in MIR-ROR as it is a very time-consuming tool that confirms digital file signatures. Enable it on a per need basis or, if you believe you've identified questionable code in a specific directory from the results of your initial MIR-ROR run, perhaps in `C:\Windows\System32`, you

1 http://code.google.com/p/rapier.

2 http://windowsitpro.com/article/articleid/98992/microsoft-hosts-le-tech-2008-training.html.

3 http://www.linkedin.com/pub/6/180/231.

4 http://www.linkedin.com/pub/4/48a/1a8.

5 http://mirror.codeplex.com.

6 http://technet.microsoft.com/en-us/sysinternals/bb842062.aspx.

7 http://holisticinfosec.org/toolsmith/files/seccheck/seccheck.exe.

8 http://www.microsoft.com/downloads/details.aspx?FamilyID=9d467a69-57ff-4ae7-96ee-b18c4790cffd&displaylang=en.

ISSA Information Systems Security Association The Global Voice of Information Security

©2009 Information Systems Security Association • www.issa.org • editor@issa.org • Permission for author use only.

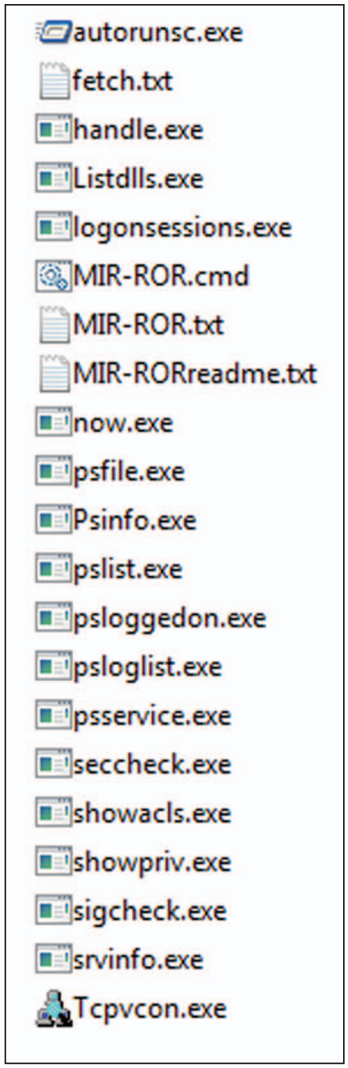

Figure 1 – MIR-ROR directory

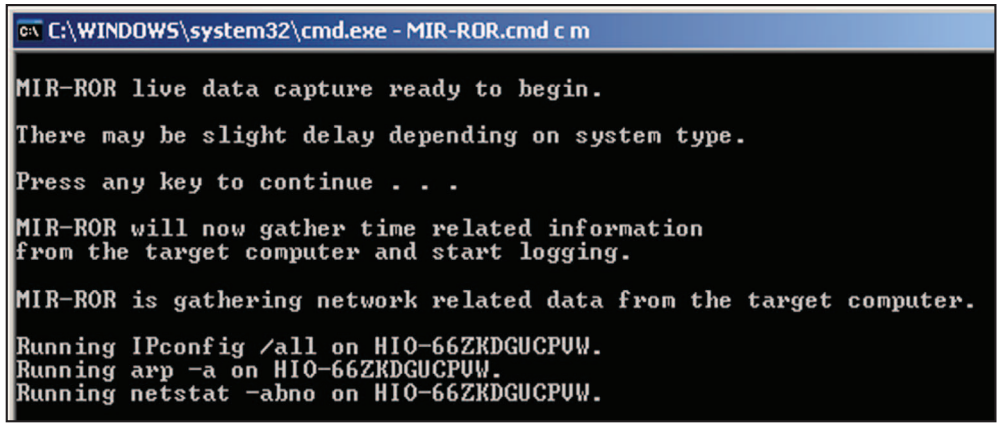

Figure 2 – MIR-ROR running on a victim system

can always choose to run sigcheck manually:

```
sigcheck –u –e c:\
windows\system32
```

If a file isn't signed in that directory, it's likely up to no good.

Much as the gentlemen from the RAPIER project have indicated, MIR-ROR does not meet the bar for sound forensic evidence collection; this is an incident response tool and it touches the system. If you must meet a legal standard take an image of the victim system.

That said, if I had a motto for MIR-ROR as an incident response tool, I would say simply this: *It just works.*

Making use of MIR-ROR

MIR-ROR calls *net *, ipconfig, arp, netstat, nbtstat, systeminfo, tasklist, openfiles, driverquery, sc, at, set, ftype, assoc,* and *doskey* from a given system's `%SYSTEMROOT%`.

MIR-ROR calls the remaining tools, *autorunsc, handle, listdlls, logonsessions, now, psfile, psinfo, pslist, psloggedon, psloglist, psservice, seccheck* (from MyNetWatchman), *showacls, showpriv, sigcheck, srvinfo,* and *tcpvcon* from the MIR-ROR directory you've populated.

If you aren't familiar with what each of these tools performs, review the Sysinternals Utilities Index[9] and the Windows Server 2003 Resource Kit Overview[10] for *showacls* and *showpriv.*

Running MIR-ROR is incredibly simple.

`mir-ror.cmd <tool drive letter> <target drive letter>`

Your first option, while logged in to a victim machine (using an investigation-only (disposable, but privileged) account), is `mir-ror.cmd c d` to call the system (netstat, nbtstat, etc.) tools from the C: drive and write the results to D:, your USB stick.

As a second option, you can establish a share called *IR to C:\tools\MIR-ROR* on what we'll call the *<MIR-ROR server>*; you'll utilize the name of whatever system you choose to run MIR-ROR from.

From what we'll refer as the *<VICTIM system>*, execute:

`net use M: \\<MIR-ROR server>\IR`

Logged on to <VICTIM system>, change directories to the M: drive.

`Execute mir-ror.cmd c m`

This will run MIR-ROR against <VICTIM system> but write the live capture results to <MIR-ROR server> at `C:\tools\MIR-ROR\Livecap_<VICTIM system>`.

You can also likely make use of PSEXEC, but I haven't yet experimented with this option; when I do I'll write up a how-to and put it on the CodePlex site.

To legitimize my claim that "It just works," I put MIR-ROR to use against a malware infected Windows XP virtual machine, having infected with a malicious binary I'd done no prior analysis against.

After issuing `mir-ror.cmd c m` on my victim, I returned to the MIR-ROR server and the `Livecap_victim` directory to review all the resulting log files.

The first log file that produced something that looked interestingly out of place was *autorunsc.log.* It's a sizeable log that can take some time to review as a whole, but if you zero in on the `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` key (a malware favorite), you'll often mine gold. Sure enough there was:

`MSN`

9 http://technet.microsoft.com/en-us/sysinternals/bb545027.aspx.

10 http://www.microsoft.com/downloads/details.aspx?FamilyID=9D467A69-57FF-4AE7-96EE-B18C4790CFFD&displaylang=en#Overview.

©2009 Information Systems Security Association • www.issa.org • editor@issa.org • Permission for author use only.

```
C:\Windows\z80.exe
(Not verified) ------
2.2.0.0
c:\windows\z80.exe
01d698b318b1a479370cf50e73912cec (MD5)
30f588294d0cb6a6f108d3f5c882df2c90a34388 (SHA-1)
01b6adf268728458c69b4aee5f2d636ef06ec89558b84e598
    5080e0faf6a1f15 (SHA-256)
```

The hash can obviously be helpful with well-known binaries, but not in this case. That said, I know full well that *z80.exe* has no business in the C:\Windows directory associated with MSN.

Using z80.exe as a keyword reference, my next hit came in *handles.log* where there were a number of process/thread references to z80.exe.

Next up, *netstat.log*. Now we're really gaining speed.

```
TCP     192.168.248.109:1292    66.252.13.221:5555
  ESTABLISHED      1660
  [z80.exe]
TCP     192.168.248.109:1293    213.131.252.251:80
  CLOSE_WAIT       1660
  [z80.exe]
```

Ah, an established connection to an external host over port 5555. Hmm, methinks the Ircbrute trojan perhaps?

Pstasklist.log showed the z80.exe process associated with PID 1660, as seen in the *netstat.log*, thus our correlation is coming together; ditto for *tasklist.log*.

The *SecCheckLog.txt* findings always speak for themselves; no less than 15 references to z80 were noted, including PID, network connections, threads, modules, and memory addresses.

Finally, *tcpvcon.log* further confirmed outbound network connection to 66.252.13.221:5555.

You may sense some redundancy here, but I firmly believe the more conclusive findings you gather to confirm your suspicions, the more successful your incident response.

Of 36 data capture logs that MIR-ROR accumulated, seven identified our culprit; more than enough to correlate findings and reach an objective conclusion. z80.exe was indeed Trojan:Win32/Ircbrute.

In conclusion

Troy's intentions in creating this script should be obvious: efficient, sound, successful incident response engagements.

My goal for formalizing the script, as well as optimizing, documenting, and maintaining it, is to ensure you have yet another valuable resource in your tool arsenal; this time inclusive of my involvement and investment in its continued development and support, as well as *toolsmith* coverage.

We hope MIR-ROR serves you well, and would really like to hear from you regarding your use of MIR-ROR. *Cheers…until next month.*

Acknowledgments

Troy Larson, for the original script and agreeing to share it with the readership.

Michael Panico, for script enhancement ideas to come.

Bryan Casper, for script testing.

Thanks, gents.

About the author

Russ McRee, GCIH, GPEN, GCFA, CISSP, is a security analyst on the Security Incident Management team for Microsoft's Online Services. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.

ISSA Information Systems Security Association The Global Voice of Information Security ©2009 Information Systems Security Association • www.issa.org • editor@issa.org • Permission for author use only.