RAPIER v. 3.1
(Rapid Assessment & Potential Incident Examination Report)

By Russ McRee

Prerequisites

32-bit MS Windows (NT/2000/XP)

Numerous third-party utilities (refer to listing at end of column)

.Net Framework version 1.1 or higher

I freely admit the fact that, at times, I find myself really excited when researching potential topics for this column and stumble across a real gem. RAPIER is just such a gem. Though still in its development infancy, it is of massive benefit to malware researchers and forensic investigators. RAPIER is an acronym for Rapid Assessment & Potential Incident Examination Report. The current version may also be referred to as RPIER or Regimented Potential Incident Examination Report and is the work of Joe Schwendt and Steve Mancini of Intel. It's currently offered under the GNU Public License v.2[1] and is a first responder's tool used to obtain volatile information from Windows OS computer systems.

According to their presentation at the 18th Annual FIRST[2] Conference, RAPIER "is a security tool built to assist in malware collection and analysis. It is designed to acquire commonly requested information and samples during an information security event, incident, or investigation. RAPIER automates the entire process of data collection and delivers the results directly to the hands of a skilled security analyst. With the results, a security analyst is provided information which can aid in determining if a system has been compromised, and can potentially determine the method of infection, the changes to the system, and the steps to recover/clean the system. RAPIER can also be used to provide anti-malware vendors with the information necessary to update their definitions files, enabling a highly effective means for rapid response to potential malware infections." [3]

RAPIER is available on SourceForge.net: http://sourceforge.net/projects/rpier

Again, note that, until the next release, SourceForge is hosting the RPIER version. Also note that a user training presentation – a bit immature but a great starting reference – is included in the SourceForge downloads for this project.

I've had the pleasure of discussing this project directly with Joe Schwendt, and he indicates some exciting developments on the horizon. According to Joe, the tool was originally released as Intel RPIER under the GPL v2; however, a fork is in progress which has a less restrictive license to allow for a more complete and better out-of-box experience. It will be renamed back to RAPIER with this fork, as RAPIER is simply a better moniker. A stub project has been setup on Google Code already – http://code.google.com/p/rapier/ – and they're in the process of gathering requirements for version 3.2, which they'll include in the public release.

Part of the issue you'll discover when attempting to use RAPIER at first is finding all of the third party tools (see third-party utilities at the end of the column) that the developers can't distribute due to the GPL. They'd like to provide a much more pleasant out-of-box experience, so they'll be morphing accordingly. Thus, though the modules will be released separately, they'll maintain the Engine under the GPL.

Joe and Steve also have several modules slated for development, so they're planning a whole new release sometime in late Q1. The prospect of a UNIX release is also on the horizon, currently in early beta stages.

Preparing for use

When you download RAPIER and unzip it, it will place all components in a self-contained directory structure that is entirely portable.

You'll find *conf*, *Modules*, and *tools*. After the first run a *Results* directory will populate. The `RPIER.conf` file will refer to a number of elements attributable to a server-based installation. For this exercise, I disabled any server references and kept the entire process local. Parameters are `disable` via the venerable # before the statement. Keep in mind as you use RAPIER in this manner, it will remind you that your connection to the server is offline. By no means does this deter RAPIER from functioning fully.

As mentioned earlier, a number of third party tools reside in subdirectories in the tree. In each you will find a file called `Required_Files.txt`. If licensing issues prevented the developers from including the tool, they listed the needed file here. So, due to these licensing restrictions, you'll have to retrieve the needed files from numerous places. I've hopefully eased a bit of that process in the third-party utilities section.

After you populate each of the *Modules* directories with the missing tools, you'll note they become available for selection in the RAPIER UI, rather than disabled in red.

I didn't add the McAfee module to my install, but I should have updated *ClamAV* before my first run – version 0.88.7 is current, while 0.88.1 is included in the RAPIER install. Grab the win32 native precompiled here: http://oss.netfarm.it/clamav/files/clamav-win32-0.88.7.7z

This version utilizes the Visual Studio 2005 libraries included in the RAPIER install.

1 http://www.gnu.org/copyleft/gpl.html#SEC1

2 http://www.first.org/

3 http://www.first.org/conference/2006/program/rapier_-_a_1st_responders_info_collection_tool.html

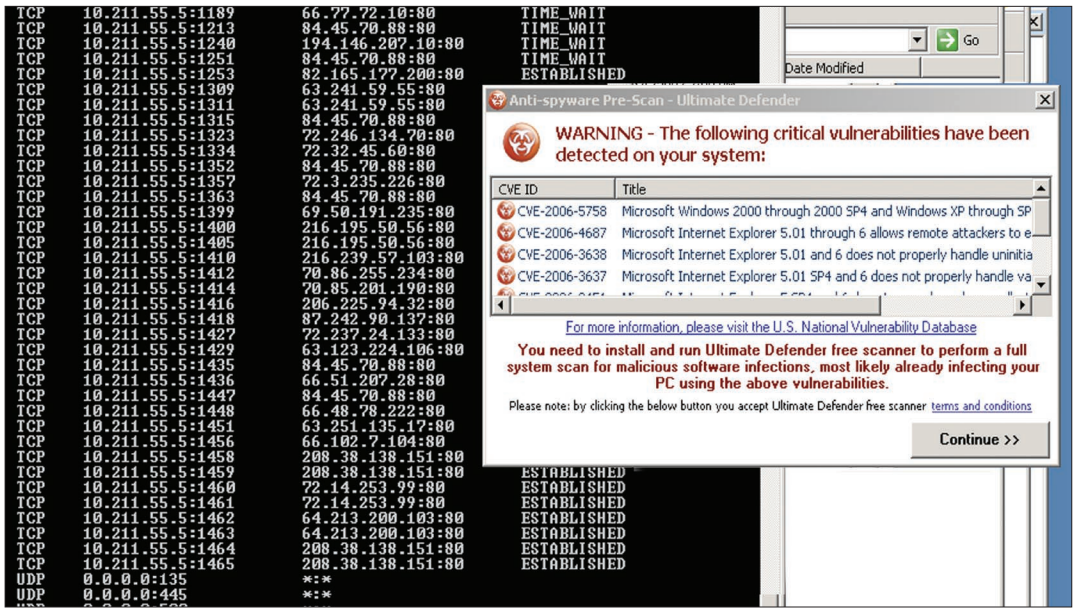

Figure 1 – Netstat scan results

Security notes before use

There are some risks you assume when using this tool. First and foremost, you will potentially discover PII or other private information. Be absolutely clear about what you may uncover, and protect yourself and the system owner by ensuring that the proper policy framework and legal mechanisms are in place before releasing the power of RAPIER on a system.

Also for your consideration: the malware issue. I mostly use RAPIER in an isolated environment (described further below) for malware research. I typically capture a sample, introduce it into a protected bubble, and let it loose on an innocent Windows XP victim. Previously, I had to gather my favorite tools into my virtual environment to perform the analysis I intended. With RAPIER, I have found that it performs a great deal of the same effort from one platform. Please note: If you release a malware sample in an uncontrolled manner, you introduce great risk to your environment. *Do not perform malware analysis on a corporate LAN.* RAPIER is an ideal tool if you already have an infected host and you need more information for remediation. But, if you're using it for research, isolate yourself entirely. To that end…

Use scenarios

The malware

For this exercise I tested RAPIER in a manner I believe best lets it shine; specifically, a virtual instance of Windows XP, infected with the *W32.Licat Worm*, also known as *Trojan. MSNMaker*, running in Parallels on a MacBook Pro. I've done this with VMWare server on a Linux box as well. Regardless, I prefer that my host OS be something other than Windows; call me paranoid.

Let me first describe the nature of a typical *Licat* delivery. Most recently I've seen it delivered via a cute IM message wherein the hapless user sees "You've been sent a Christmas card!" and decides it's a good idea to step on the link. Oh, the opportunities for increased user awareness abound. Engaging the link results in the download and deployment of `card232.exe`, after which all hell breaks loose, including death and devastation to MSN Messenger. See Spywareguide.com[4] for more information on this little funfest. Most antivirus vendors had not yet identified the variant I most recently investigated, so it had a window of opportunity to be annoying before the *AV sig* files caught up. Unfortunately, the only real solution to a worm like this, given that its purpose is largely spyware delivery, is to reimage the PC.

First, consider all the outbound connections made after the PC is infected, as well as the "helpful" popup in Figure 1. Then take note of all the nastiness written to the root of C: in Figure 2.

Finally, look at all the whacked out processes running in Task Manager (before it no longer functions) in Figure 3.

You get the point: this machine is thoroughly hosed up.

RAPIER on point

After firing up RAPIER on this destroyed virtual instance, I selected a custom scan. You can choose fast or slow, but keep in mind, a slow scan can take hours as it is extremely thorough.

My selections included those I believe will most aid in discovering the nature of an infection, including *DumpUsers, Rootkit Revealer, Services, ADS, Hidden Files,* and *ClamAV Scan*.

4 http://www.spywareguide.com/spydet_3138_w32_licat_worm.html

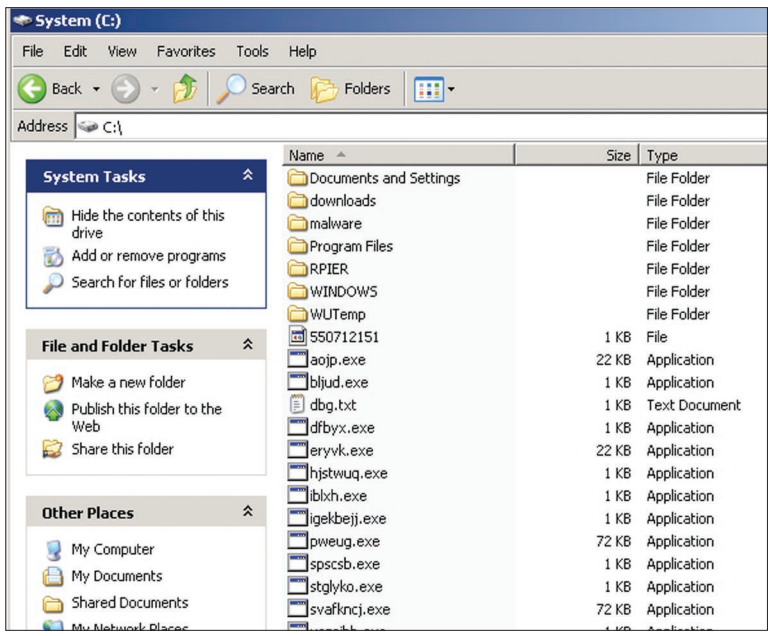

Figure 2 – C: gone bad

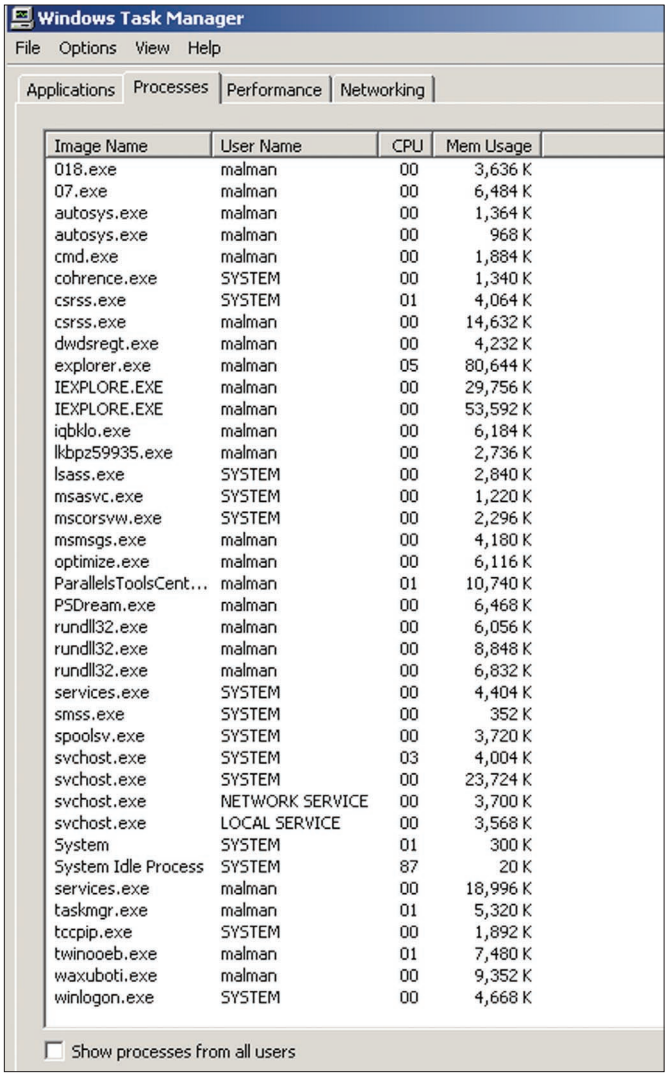

Figure 3 – iqbklo, waxuboti, and lkbpz59935, oh my!

With the selections I'd chosen this scan took just under 26 minutes to execute. The results of a RAPIER scan are written to the *Results* directory and can be immediately reviewed from the UI by choosing *File* then *Open Results Directory*. The `RPIER.log` will summarize the scan, confirming what modules ran and how long each module took to complete.

The results of DumpUser did not reveal any new or unexpected accounts, so no concerns there:

```
junk  User  Built-in account for guest access to
the computer/domain  No  Never  No  No  ?Unknown
Yes  No  Never  Never  EISA030  All  No  None
malman  User  Built-in account for
administering the computer/domain  Yes  1/5/2007
5:06 PM Yes  No  Never  No  No  Never  1/5/2007
5:33 PM EISA030  All  No  None
Execute Duration (in seconds)=6
```

Rootkit Revealer turned up some interesting information:

```
Module Name=RootKitRevealer
Description=SysInternals RootkitRevealer
Execute Time=Fri 2007/01/05 19:35:30HKLM\
SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
yeoimgn:
  Description: Hidden from Windows API.
  Date:  1/5/2007 7:04 PM
  Size:  80 bytes
C:\WINDOWS\system32\yeoimgn.dat:
  Description: Hidden from Windows API.
  Date:  1/5/2007 7:36 PM
  Size:  4.35 KB
C:\WINDOWS\system32\yeoimgn.exe:
  Description: Hidden from Windows API.
  Date:  1/5/2007 6:59 PM
  Size:  271.50 KB
C:\WINDOWS\system32\yeoimgn _ nav.dat:
  Description: Hidden from Windows API.
  Date:  1/5/2007 7:02 PM
  Size:  241.09 KB
C:\WINDOWS\system32\yeoimgn _ navps.dat:
  Description: Hidden from Windows API.
  Date:  1/5/2007 7:36 PM
  Size:  334 bytes
Execute Duration (in seconds)=95
```

Nothing about these results can be good. No search turned up any data stating that "yeoimgn hidden from the Windows API" is an expected norm.

The *Services* and *ADS* modules didn't turn up anything out of the ordinary, but where RAPIER really strikes gold is with the *Hidden Files* module. While there are a plethora of expected, normal files uncovered, there are also a number that are entirely out of place:

```
c:\Program Files\PSDream
c:\Program Files\siteerror search
c:\Program Files\Ultimate Defender
c:\Program Files\Common Files
  Yazzle1162OinAdmin.exe 05/01/2007 18:58:59
  Yazzle1162OinUninstaller.exe 05/01/2007 18:59:01
c:\Program Files\Internet Optimizer
c:\Program Files\NewDotNet
```

This list is but a fraction of those uncovered by the *Hidden Files* module on this heavily infected system.

Take note of the `NewDotNet` entry, we'll see that again below.

The final truth is most evident via the *ClamAV* module.

Even though I hadn't updated the *ClamAV* engine or `sig` file, it discovered quite a collection:

```
Module Name=ClamAVScan
Description=Clam Antivirus Command Line Scan
of the system drive
Execute Time=Fri 2007/01/05 19:37:55
-------------------------------------
Scan started: Fri Jan 05 19:37:57 2007
018.exe: Adware.NewDotNet.B FOUND
07.exe: Adware.NewDotNet.B FOUND
cscript.exe: Adware.NewDotNet.B FOUND
dwdsregt.exe: Adware.NewDotNet.B FOUND
explorer.exe: Adware.NewDotNet.B FOUND
IEXPLORE.EXE: Trojan.Dyfuca-3 FOUND
optimize.exe: Trojan.Dyfuca-20 FOUND
```

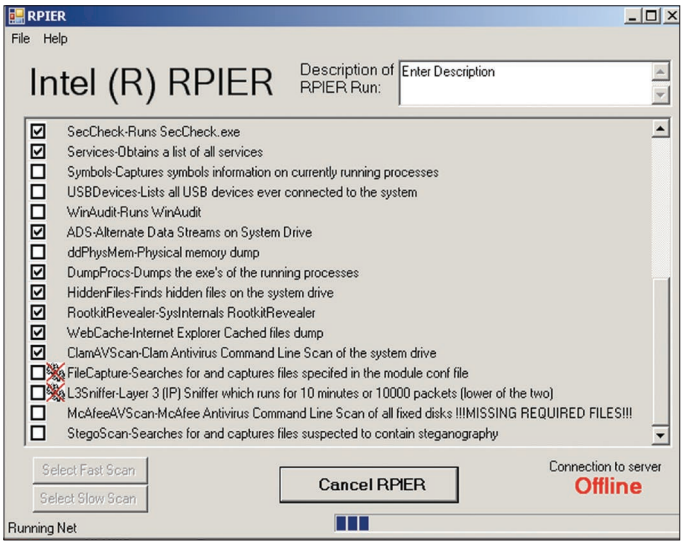

Figure 4 – RAPIER module selection

```
ParallelsToolsCenter.exe: Adware.NewDotNet.B
FOUND
PSDream.exe: Adware.NewDotNet.B FOUND
twinooeb.exe: Adware.NewDotNet.B FOUND
waxuboti.exe: Adware.NewDotNet.B FOUND
winlogon.exe: Trojan.Agent-68 FOUND
nem220[1].dll: Trojan.Dyfuca-3 FOUND
NNSKYA638[1].exe: Adware.NewDotNet.B FOUND
secure32[1].htm: Adware.Atris-1 FOUND
-- summary --
Known viruses: 60727
Engine version: 0.88.1
Scanned directories: 102
Scanned files: 1267
Infected files: 15
Data scanned: 651.55 MB
Time: 443.491 sec (7 m 23 s)
Execute Duration (in seconds)=446
```

RAPIER, with a scan including just eight modules, produced all the results one might expect from a deeply infected host.

Assuming the correct installation of the zip module requirements, the results are neatly zipped for you for offline analysis or export to an AV vendor or malware analyst.

In it's current release, RAPIER is still a bit buggy, most notably when terminating the UI if using .NET 2.0. I've found it's best to open *Task Manager* (assuming it hasn't been mangled on your subject) and terminate the RPIER.exe process. Otherwise you may find the UI struggling to close in unstable state if you close via the UI exit option.

Alternatively, ensure the use of .NET 1.1 wherein I've not had the same problem.

Also, keep in mind as you use RAPIER, future releases will make the module process far simpler and should present a ready-to-go user experience.

Conclusion

RAPIER is a project with extraordinary promise, and one I will watch closely with the expectations of a kid on Christmas Eve. If you enjoy running down badware as much as I do, you will find RAPIER an extremely useful part of your toolbox. Stay tuned to the Google Code site for future releases, likely near the end of Q1 2007.

Consider reading Joe Schwendt's paper, "*RAPIER 3.0 An Advanced Malware Collection Tool*,"[5] for his SANS GSEC certification.

Consider also perusing www.stopbadware.org and supporting their cause in any way. Until next month…cheers.

Third-party utilities

You'll need to download these and populate the appropriate *Module* directory. This list is likely incomplete so after you download and place the necessary tools, open the RAPIER UI and take note of what modules remain unavailable. Then visit the unavailable module's directory and review the `Rquired _ Files.txt` file for what's missing.

You'll also note that RAPIER takes advantage of Mark Russinovich's indispensable Sysinternals tools.

1) dumpsec[6]

2) rootkit revealer[7]

3) handle.exe [8]

4) Listdlls.exe [9]

5) macmatch.exe[10]

6) MBSA: mbsacli.exe wusscan.dll wsusscan.cab[11]

7) reg3.exe – really hard to find, so I posted it on my website. [12]

8) srvinfo.exe from the Windows Server 2003 Resource Kit[13]

9) checksym.exe from MPSRPT_MDAC.EXE[14]

Legalese

RPIER - Copyright Intel Corporation 2006 under the GNU Public License version 2

RAPIER (Rapid Assessment & Potential Incident Examination Report) is designed to acquire commonly requested information during an information security event, incident, or investigation. Individual modules may utilize third-party utilities.

About the Author

Russ McRee, GCIH, is a security analyst working in the Seattle area. He is a member of ISSA, Pacciso, InfraGard, and CCSA (Cyber Conflict Studies Association). Russ maintains holisticinfosec.org. Contact him at russ@holisticinfosec.org.

5 http://www.giac.org/certified_professionals/practicals/gsec/4752.php

6 http://www.systemtools.com/free.htm

7 http://www.microsoft.com/technet/sysinternals/utilities/RootkitRevealer.mspx

8 http://www.microsoft.com/technet/sysinternals/ProcessesAndThreads/Handle.mspx

9 http://www.microsoft.com/technet/sysinternals/ProcessesAndThreads/ListDlls.mspx

10 http://www.ntsecurity.nu/toolbox/macmatch/

11 http://www.microsoft.com/technet/security/tools/mbsa2/default.mspx

12 http://holisticinfosec.org/toolsmith/files/apps/

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

14 http://www.microsoft.com/downloads/details.aspx?FamilyID=CEBF3C7C-7CA5-408F-88B7-F9C79B7306C0&displaylang=en