Confessor & MOLE

Join the Discussion
Connect

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

Prerequisites

Windows OS for Confessor
Virtualized or disposable Windows instance for MOLE

This month's *toolsmith* marks the beginning of our fifth year of bringing readers insight on tools useful to the information security practitioner. The *Journal* has come a long way since November 2006, and I've grown a lot in that time as well.

During September's ISSA International Conference I gave a presentation that was intended to share a few incidents of interest that my team has handled at Microsoft, incidents that had been sanitized and approved for discussion. In addition to demonstrating the use of tools I've discussed before in *toolsmith*, including AfterGlow, NetGrok, and Malzilla, I was most pleased to demonstrate two tools created by team members Bryan Casper and Kris Thomas.

You may recall MIR-ROR, as discussed in the June 2009 *ISSA Journal*,[1] as a tool useful during live incident response to gather and collect system logs and attributes. MIR-ROR is effective on a small number of hosts, but really does not meet an enterprise standard. Bryan created Confessor[2] to answer that challenge, utilizing the same tools as MIR-ROR, but deploying them in a manner for use on hundreds or even thousands of systems at the same time.

The other tool was spawned again from a method I'd been utilizing to cull malware from a list of the top 200 URLs sent across the Windows Live Messenger service each day. Where I'd been using a specific `wget` string at the command-line, Kris built MOLE[3] (Malicious Online Link Engine) as a wrapper for `wget` that includes many additionally useful features.

Confessor configuration

There are a few important steps to configuring Confessor before running it for the first time.

You need to download the required dependencies such as the SysInternals PsTools, as well as a few others. The README PDF will clarify all the setup steps for Confessor.

Equally important is the fact that Confessor, in its current iteration, is an enterprise tool and as such only runs in Active Directory (domain) environments. You need to run Confessor with the same permissions as used for administrative privileges on your target hosts.

Anonymous read permissions are required for the tools directory and anonymous write permissions are needed for the results directory. In the Confessor *Scan* pane you'll configure the UNC path to the share you've established for all tool dependencies; likewise for where you want results written. The results will be written to individual directories named for

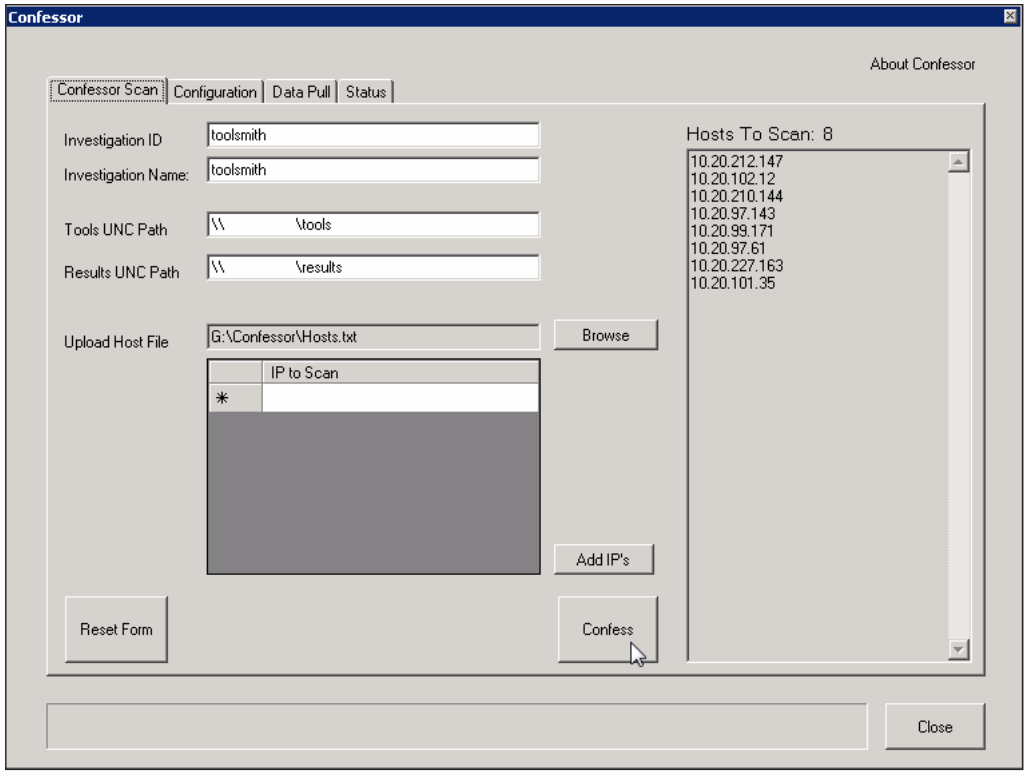


Figure 1 – Configure Confessor scan

1 http://holisticinfosec.org/toolsmith/docs/june2009.pdf.

2 http://confessor.codeplex.com.

3 http://mole.codeplex.com.

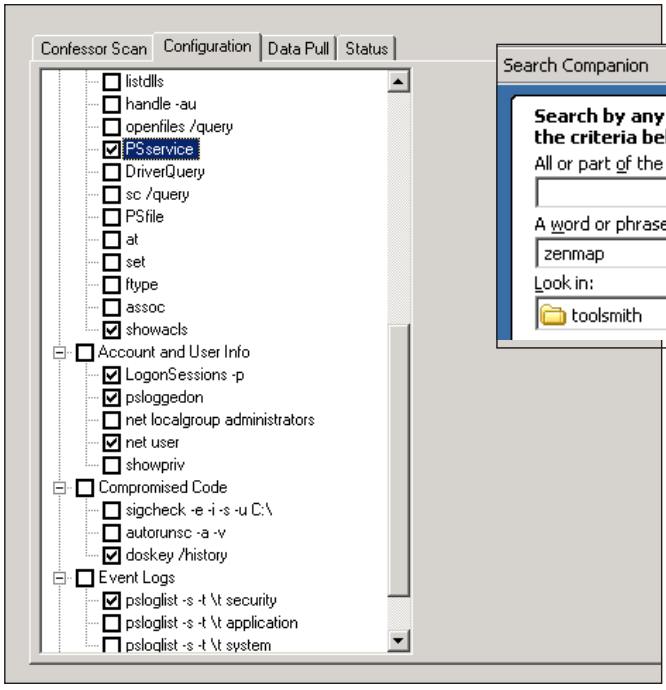


Figure 2 – Select Confessor utilities per category

each target server. You can define IP addresses individually or import them directly from a file (see Figure 1). The next version of Confessor will include use of hostnames as opposed to just IP addresses.

In the *Configuration* pane, you'll choose the categories you wish to run or individual utilities. You can focus the collection on your areas of interest. Examples include the network stack with the likes of netstat or OpenPorts, Volatile Information includes the likes of pslist and handle, Account and User Info via psloggedon and others, as well as event logs and MAC times as seen in Figure 2. Remember, this tool turns the MIR-ROR script into something far more viable in enterprise environments.

You'll find two additional tabs not yet in use. Progress on results details will populate the *Data Pull* tab and errors will present themselves in *Status*. Stay tuned for the next round of enhancements.

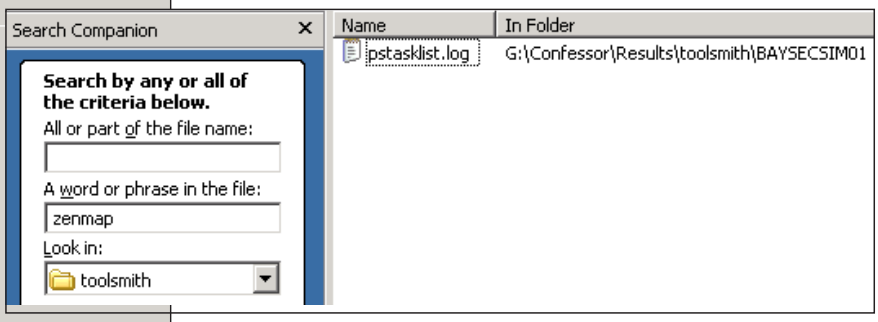
Confessor results and analysis

Once the Confessor run completes, you can begin analysis via a variety of methods.

If you ensure that the results directories are added to the Windows search index, keyword or string searches become pretty straightforward. I admit a major fondness for TextPad, as multiple log files can be loaded and searched with truly robust features including bookmarking and regular expressions.

I put a little scenario together for Confessor with a small run against eight concurrent hosts. Imagine you've received an external complaint regarding a host from one of your server farms perpetually scanning the complainant's network. As it is an environment with distributed responsibilities you have

Figure 3 – Confessor scan search results



no access to network devices, so no SPAN or tap for you; you can only work from the servers themselves. Further ambiguity is created by the fact that all your servers are behind a load balancer with a virtual IP address and all hosts NATed. The one saving grace is that you know that only a small number of hosts in your farm are allowed to initialize egress connections so you have a narrow range to feed to Confessor. Configure Confessor with said host IP addresses and wait for both the progress bar to complete and the `_finished.txt` file to write to each target host results directory. Ideally, you'd probably select pslist from Volatile Information along with other process and network-oriented utilities. Once the scan is complete, your results directory should show directories for all the hosts scanned and a log file for each utility you selected written accordingly.

As logs gathered by the complaining external party indicate the likelihood of nmap in use of one of your servers, you could keyword search the results directory as seen in Figure 3.

Ah-hah, one server coughed up the goods. The `pstasklist.log` is the output from the pslist utility; sure enough you've found your culprit and can now question the server admin.

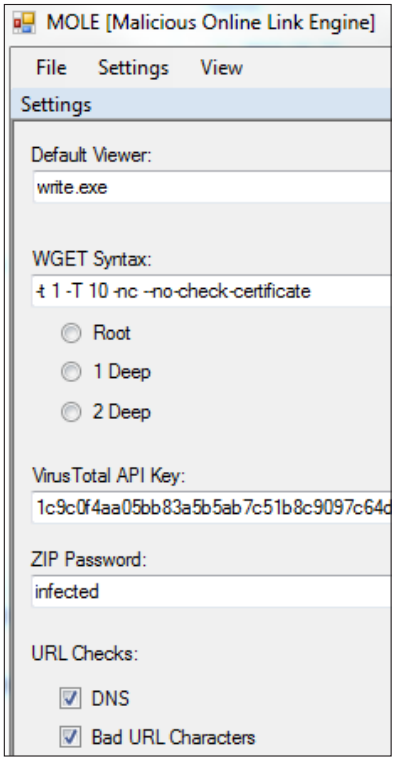
```
Name     Pid     VM       WS      Priv    Priv Pk
zenmap   2164    115136   28104   17652   18272

Faults   NonP    Page     Tid PriCswtch State
21430    15      188      4716    8       12406

User          Time        Kernel Time Elapsed Time
Wait:UserReq  0:00:02.184 0:00:00.967     0:33:56.717
```

Imagine having run Confessor across a thousand hosts rather than eight. Either way the results would have been exactly the same and you would have found your scanner with relative and similar ease.

MOLE configuration

One of the tasks my team regularly undertakes is identifying malicious sites, in short order, from a list of URLs, with immediately actionable results. MOLE fulfills these requirements with a bunch of functionality that you'll definitely not derive from `wget` at the command-line. As required with Confessor, you have to populate the dependencies directory yourself (licensing issues), but it's a short list for MOLE and all file URLs are provided in the manifest (README.rtf). Once the required files are copied to *Dependencies*, simply

Figure 4 – Configure MOLE



run `mole.exe`. You'll be presented what initially appears to be a very simple UI. Copy your URL list in the white-space pane under the project date; then click *Settings* from the menu. You can determine how many directory levels deeps you wish MOLE to crawl, set the zip password (for all the malicious binaries MOLE will find), and set your personal Virus Total API key[4] (See Figure 4).

IMPORTANT: Run MOLE in a sandboxed environment, either a VM or on a system that you're confident you can protect from the malware that will be downloaded to the file system.

Once you're all setup, it's as simple as clicking *Start*.

MOLE results and analysis

Results will be written, by default, to a folder named for the date of the MOLE run. There will subdirectories named for each URL passed to MOLE. If the crawl result is a simple GET for an HTML page met with a 200, the related HTML will be populated to the subdirectory for that URL. But if the URL points to a malicious binary, MOLE will GET it and drop it to the subdirectory as well.

If you're feeling like an URL is not directly malicious but you want to review it, the menu offers you the option to browse it. It will ask you (remember my sandbox warning) *"Are you sure you really want to do this? You might get pwnd!"* Take this seriously, folks. You can also browse the related URL subdirectory and better assess the file MOLE pulled down. Even if not directly malicious via a binary, you might find obfuscated malicious JavaScript tucked in an otherwise seemingly innocent bit of HTML. You can always feed said JavaScript to Malzilla[5] for analysis. Here's where the really cool features kick in. After MOLE runs through the list (or while crawling if you're impatient) you can select *View*, then *Executables*. All the PE32 files that MOLE downloaded are identified for you as culled from the primary URL list. Even if the list is thousands of URLs long and only a small number actually offer binaries, MOLE will weed them out as seen in Figure 5.

4 http://www.virustotal.com/advanced.html.

5 http://holisticinfosec.org/toolsmith/docs/july2009.pdf.

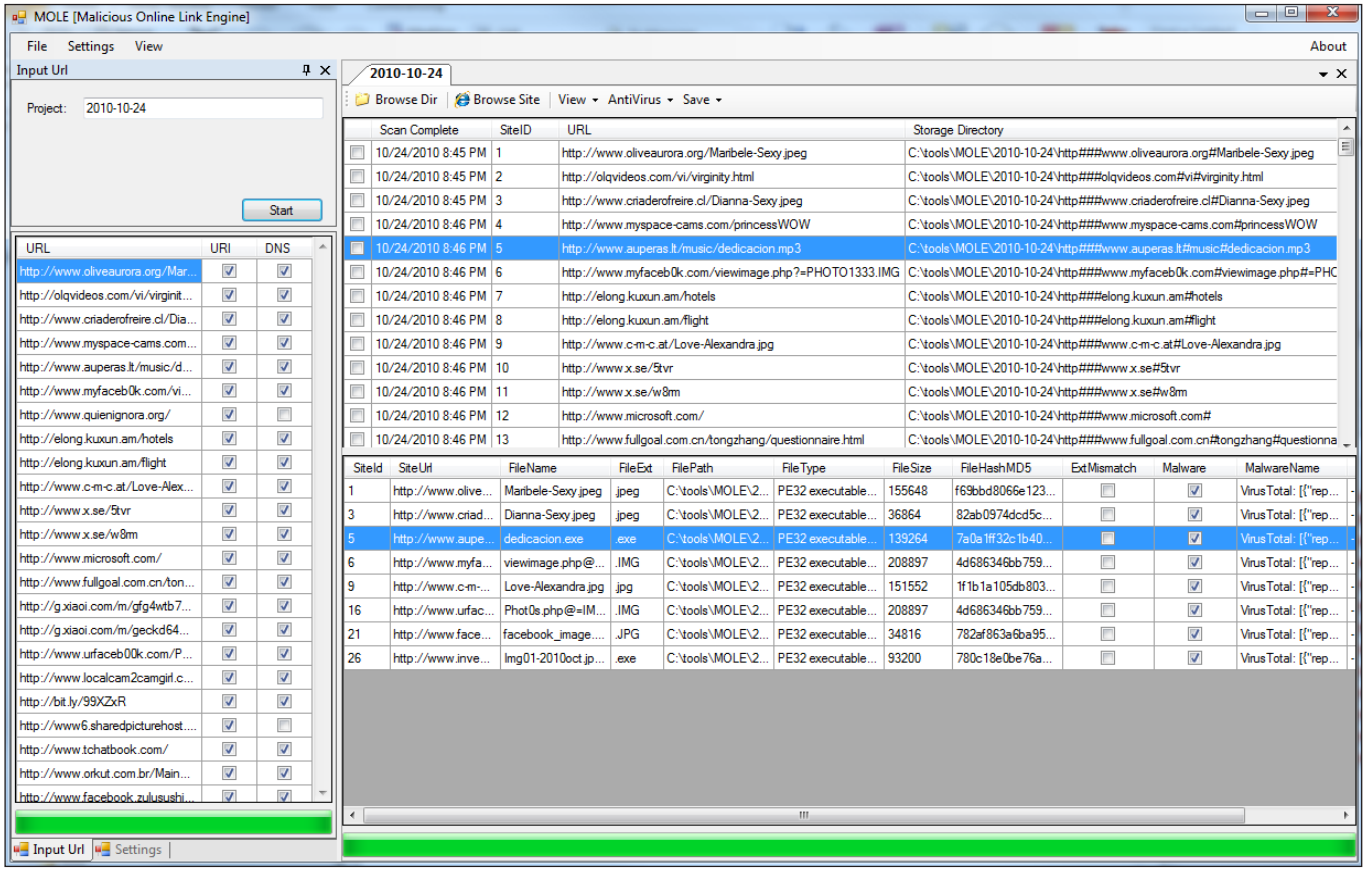


Figure 5 – MOLE finds malicious binaries

Better still, as longs as you've configured your Virus Total API key in Settings, you can choose *AntiVirus* and submit all the binaries to Virus Total in one shot. Results will be written to the MalwareName column in the lower MOLE pane. As seen in Figure 5, `dedicacion.exe` is identified as `Trojan.Win32.Generic.pak!cobra`, amongst other things. If you wish to collect all the binaries in a password protected zip (password set in Settings), just *Save,* then one of the two ZIP options, either all binaries or only those you've checked as Malware after confirmation. You can then analyze them or submit them elsewhere. Finally, you may wish to create a list of just the malicious URLs. Simply select Save then *Bottom URL List (TXT)*. The resulting list from the 200 URLs I submitted to MOLE as seen in Figure 5 follows (they are all malicious):

- hxxp://www.oliveaurora.org/Maribele-Sexy.jpeg
- hxxp://www.criaderofreire.cl/Dianna-Sexy.jpeg
- hxxp://www.auperas.lt/music/dedicacion.mp3
- hxxp://www.myfaceb0k.com/viewimage.php?=PHOTO1333.IMG
- hxxp://www.c-m-c.at/Love-Alexandra.jpg
- hxxp://www.urfaceb00k.com/Phot0s.php?=IMG35011.IMG
- hxxp://www.facebook.zulusushi.com/facebook_image.php?image=IMG00250802010.JPG
- hxxp://www.invent-now.com/inventnow/Img01-2010oct.jpg

You can then choose to block these URLs via a proxy if in use or other preferred method.

Imagine using MOLE when you have thousands of URLs to review (think malicious online advertising) and you need to find the binary and its host quickly. MOLE is extremely capable in this capacity.

In conclusion

Thousands of servers or thousands of URLs, Confessor and MOLE serve us well, and we hope they serve you well too.

I believe that security teams who can create their own tools have an advantage over teams who may be limited to existing commercial or freely available tools. However, where possible, teams who share tools they create can offset that advantage by releasing those tools for public use. In my mind, I even consider this an obligation. It's in that spirit we wanted to make Confessor and MOLE available to you; please use them in good stead with our best wishes.

Cheers…until next month.

Acknowledgements

—Bryan Casper, for Confessor

—Kris Thomas, for MOLE

—Elliot Lazarus for tool graphics and testing

About the Author

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