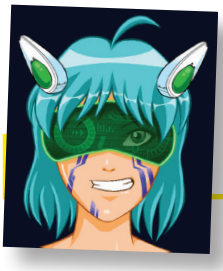

Armitage

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

Prerequisites

Virtualization platform or dedicated physical host for BT4R2

First, Happy New Year.
I hope 2011 will be a good one for you all.

Likely you've all heard of Metasploit if not used it as part of ethical hacking training exercises or penetration testing engagements. Depending on your background or the availability of commercial tools in your environment (Core, Canvas, etc.), your comfort with Metasploit likely varies with the depth of your experience. Armitage[1] is designed to help close some of the experience or comfort gaps, described by the developer as useful for "non-hackers".

Raphael Mudge, Armitage's developer, has *"met too many people involved with the defense of large networks who do not understand hacking and what's possible today. Some smart people think if they don't know how to do something, then it must be difficult, so they're willing to assess the risk of it as lower. This is very dangerous in network defense. Armitage exists to make it easier for non-hackers to understand what today's tools are capable of."*

Raphael proposed of few use cases for Armitage, first as a learning tool for people who are new to Metasploit and find themselves struggling with three questions:

1. What can I do?
2. Which exploit do I use?
3. Ok, I compromised that host, now what can I do?

Per question 1: Armitage is logically organized around the vulnerability discovery and exploitation process. The documentation[2] will help orient the Metasploit workflow process and orient the user accordingly.

Per question 2: Armitage uses Metasploit's capabilities to help out where possible. Armitage recommends exploits to help narrow the number of exploits a user must search through. For services that have many exploits associated with them, Armitage can run each exploit's check command to help the user find the right exploit to use.

Per question 3: Raphael put a lot of effort into making it easy to manage post-exploitation through Armitage. The user can escalate your privileges, capture hashes, or take a screenshot with one click. Armitage also allows users to browse files and interact with a Windows command shell simultaneously.

A second Armitage use case is as a demonstration tool. Have you ever watched a demonstration of Metasploit? It can be painful for a non-techie; imagine a lot of gray text scrolling on a black background. Once the demo is over, it usually requires several slides to explain what happened. Armitage captures the action in a way anyone comfortable with computers can follow.

According to Raphael, professional penetration testers seeking to replace commercial products may need to wait a while. Armitage lacks the reporting and auditing features that these tools provide. Raphael would like to hear your needs before addressing these features. For now, he's focused on empowering non-professional penetration testers and making it possible for system administrators to test their own networks.

During red team exercises, Raphael noticed a few problems that today's tools aren't solving well as it can be difficult for a red team to coordinate efforts, share sessions and information. His original goal for Armitage was to make a UI for Metasploit that made team cooperation possible for these CTF/exercise environments. He's hit milestone 0 by providing an effective local client for Metasploit. The next milestone is to make it possible to manage Metasploit remotely as well as Armitage does locally. Imagine the possibilities for coordinating multiple Metasploit instances collocated in the cloud. Beyond that, he hopes to implement the team cooperation features.

As for additional Armitage functionality improvements, Raphael would like to see it handle attacks against web applications as well as it handles attacks against the OS and client-side applications. Metasploit includes WMAP[3] but it requires additional development before Armitage can leverage it.[4]

Setting up Armitage

I used the opportunity to test Armitage to also test BackTrack 4 R2, downloaded the VMWare image,[5] and installed Armitage with ease. Metasploit 3.5.1-dev is native to BackTrack 4 R2 (run `msfupdate` to update it to the current version, 3.6.0-dev as I write this), as is MySQL, which makes Armitage setup very simple.

1 http://www.fastandeasyhacking.com.

2 http://www.fastandeasyhacking.com/manual.

3 http://www.metasploit.com/redmine/projects/framework/wiki/WMAP.

4 Interview feedback provided by Raphael Mudge.

5 http://www.backtrack-linux.org/downloads.

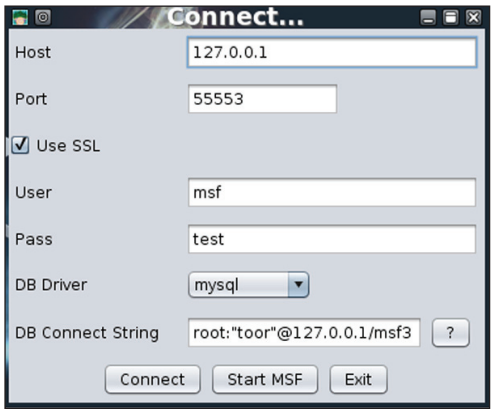

Figure 1 – Connect Armitage to Metasploit RPC daemon

Unpack the Armitage archive, then `cd /pentest/exploits/framework3`, followed by `./msfrpcd -f -U msf -P test -t Basic`. This will fire up the Metasploit RPC daemon with the user msf, password test, and an SSL listener on the default port 55553. You can modify this as you see fit. Be sure to start MySQL: `/etc/init.d/mysql start`. Change directory back to your Armitage installation and run `./armitage.sh`; be sure you check the Use SSL box when connecting as seen in Figure 1.

Using Armitage

Once Armitage is connected and running, define a workspace via *Workspaces*, then *Create*. You scan targets via the Hosts menu; enter the IP addresses of the host(s) you seek to "explore." You can opt to run an Nmap scan from Armitage, but it is recommended that you import Nmap results from a direct client scan rather than a scan called from the Armitage UI. Armitage does not report results back to the console in real-time, thus leaving you in the dark on scan progress. That said, the console from which you launched msfrpcd will report Nmap activity to you. You can also choose to run MSF scans which will launch 19 discovery modules.

Armitage includes extensive import functionality, consuming scan results and host lists from THC-Amap, Nessus, NeXpose, and Qualys amongst others.

My test network (192.168.122.0/24) included a couple of vulnerable Windows Server 2003 virtual machines to exemplify host pivoting where one compromised host can then be used as a jump-off platform for exploration and further exploitation.

Once your scans are finished, you will be advised to use *Attacks* then *Find Attacks by port* or vulnerability. You'll note icons for host IPs populate in the Target UI. I used Find Arracks by port; once analysis is complete you will be advised of a right-clickable *Attack* menu attached to the host(s). One note: if you use *Find Attacks by vulnerability*, if no vulnerabilities have been identified, no attack menu will be populated.

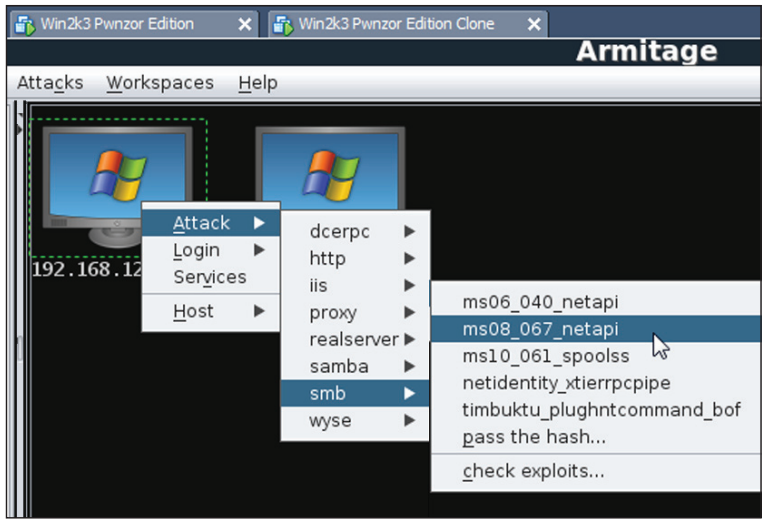

Figure 2 – Armitage attack options

As seen in Figure 2, I opted to exploit the server service vulnerability typically exploited by the Conficker worm, specifically MS08-067.[6] The related Metasploit module "exploits a parsing flaw in the path canonicalization code of NetAPI32.dll through the Server Service." After launching the exploit, Armitage will report back a compromised host as a lightning stricken red icon inclusive of a Meterpreter session with access, interact, explore, pivoting, and MSF scan options as seen in Figure 3.

Note that *Access* options include privilege escalation and hash dumping for later use as part of pass-the-hash attacks. First set up a pivot by right-clicking your initial compromised host. Select *Meterpreter n*, then *Pivoting*, then *Setup* to define the subnet to pivot through. In the hierarchical target view you'll note a dim green connector arrow after setting up the pivot; the same arrow will become bright green once achieving a successful compromise of a secondary host.

One can take advantage of the hash dump as seen in Figure 3 to attack the secondary host. Right click the second-

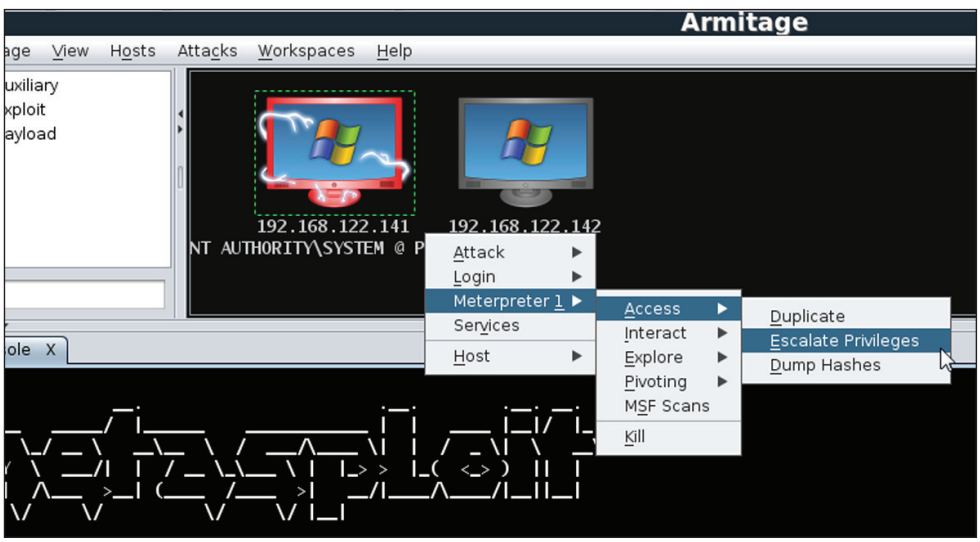

Figure 3 – Armitage compromised host Meterpreter session

6 http://technet.microsoft.com/en-us/security/dd452420.

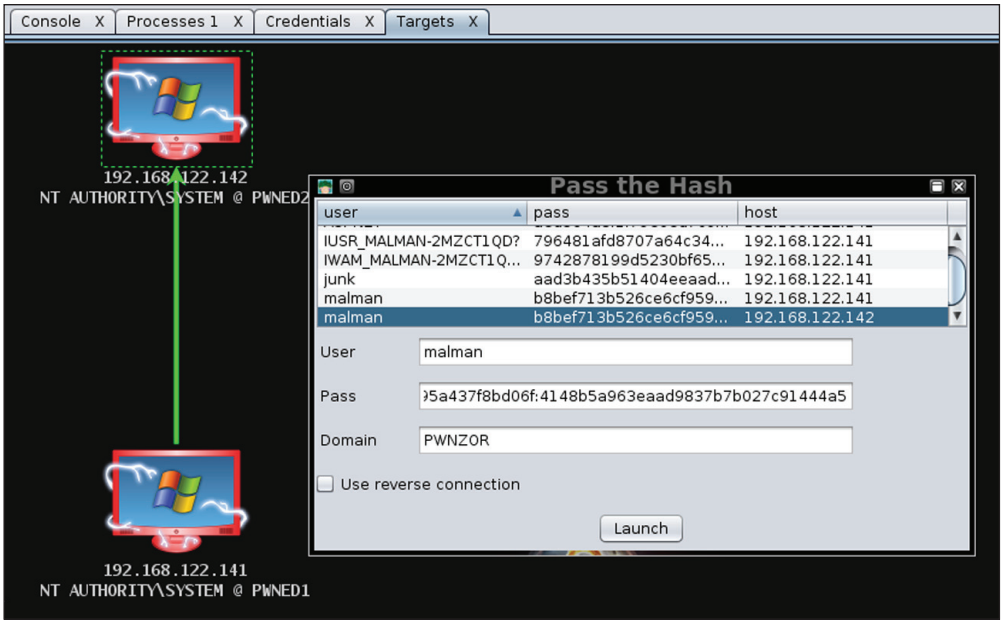

Figure 4 – Armitage pass the hash pivot session

ary host, select *Attack*, then *smb*, then pass the hash. Select one of the hashes grabbed from the initial host and click the *Launch* button; you can review them prior to passing them via *View* then *Credentials*. Figure 4 shows the *Pass the Hash* UI and a fully completed pivot session for posterity.

Once you've established a Meterpreter session, there are endless possibilities, many of which are well suited to the above mentioned demonstration functionality. There's nothing like compromising a system as part of an authorized engagement, and using Armitage to grab a screenshot of the desktop for the active system user and returning it to the Armitage tabs menu as seen in Figure 5.

Other host exploration options include running processes, browsing the filesystem, or spawning a command shell via the *Interact* menu. You can even run VNC if wish, not that you couldn't enable RDP with escalated privilege.

The Armitage manual in comprehensive and includes information far in excess of what we've discussed here. Read through it before getting started to avoid time sinks that are explained clearly (why would I read the manual first?).

In conclusion

Armitage delivers exactly as promised. I'm looking forward to continued feature enhancements and heartily suggest you give it a close look as a learning or demonstration tool. I'm already seeing the upside of being able to show clients or senior managers as one of their systems fall due to a woeful patch state or vulnerable application.

Use Armitage and similar tools carefully; have your "get out of jail free" card hand at all times.

Cheers…until next month.

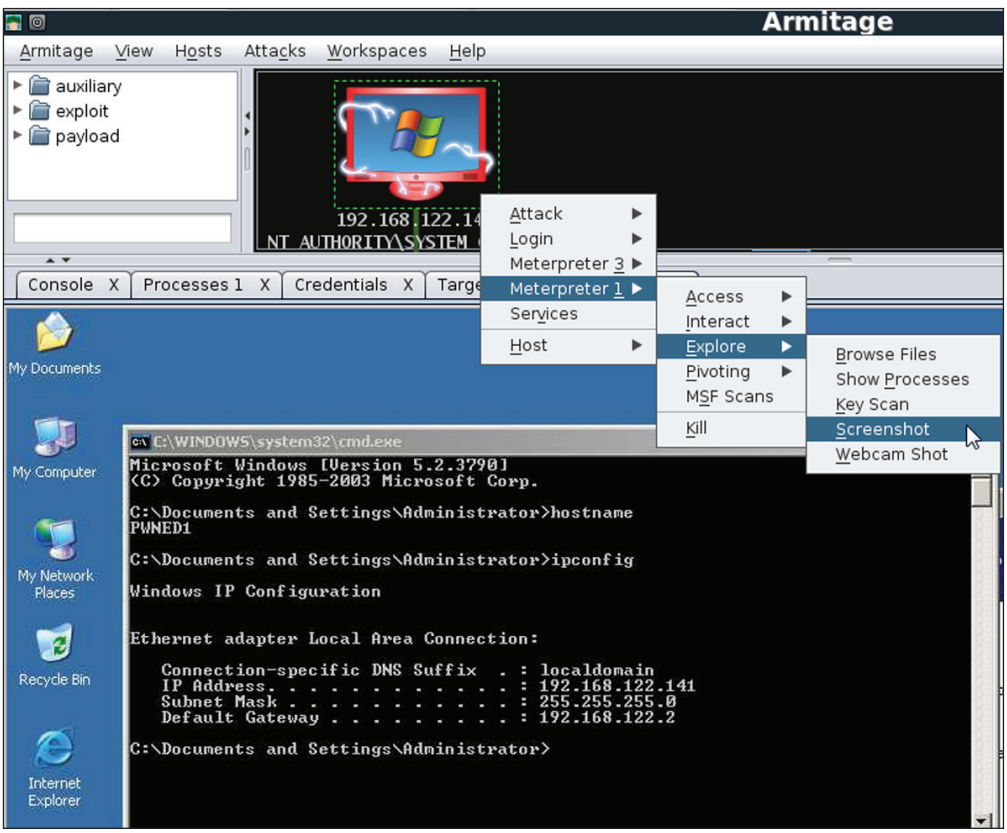

Figure 5 – Armitage screenshot of compromised system desktop

Acknowledgements

—Raphael Mudge, Armitage developer and project lead.

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.