A Hands-On Approach to Computing Security Education: Metasploit Module Development

1. Abstract

As the demand for skilled computer security professionals continues to increase, curricula at colleges and universities must continually evolve to meet current industry needs. Incorporating Metasploit module development into these curricula increases the value of the education received by the graduate and makes them a more attractive candidate to potential employers. The flexibility of the Metasploit framework, especially for targeting mobile device platforms, proves to be a valuable tool in evaluating the security of many different devices. This paper discusses the importance of incorporating Metasploit module development into computing security curricula to better prepare graduates for the constantly changing environment they will encounter. Hands-on lab outline, module development experiences, and related work are also discussed in the paper.

2. Introduction

The computing security industry is evolving rapidly as cyber security concerns continue to grow across all major industries. This is undoubtedly influenced by the large number of devices that are capable of being connected to the Internet. Some estimates suggest there are at least 16 billion devices that are capable of being connected as of late 2015 [10]. This trend shows no signs of decreasing as it is said that up to 127 new devices are connected to the Internet each second [9]. This rapid growth vastly increases the demand for qualified computing security professionals and it is essential that their education is relevant in this quickly changing industry.

The introduction of Metasploit module development in computing security curricula adds significant value to the education that a college or university provides to a graduate of their programs. Metasploit is a popular framework in the computing security community that is used to create and develop exploits [11]. This flexible framework allows users to test and easily repeat public exploits against a variety of software and devices. While there are already a large number of Metasploit modules available in the framework that can be used to test for vulnerabilities, the ability to take an existing module and modify it, even slightly for a unique situation, could prove significantly valuable. Additionally, creating your own module to fit the needs of a targeted vulnerability are advantageous not only to the author but for others in the industry as well.

Mobile and embedded devices are an especially explosive area of the industry; end users may not only have a smartphone, but a tablet, watch, pair of glasses, appliance, or a vehicle that are all connected to the Internet. For a computer security professional concerned with the vulnerabilities associated with these devices and the risk that is posed to those who use these devices, the scale of these issues is significantly larger than it was only several years ago.

To add further complication to the drastically increasing quantity of devices that are being used, many of them run on the Android operating system. The security of a device running Android depends heavily on whether the device receives timely software updates that fix critical vulnerabilities [12]. This presents additional challenges as a large percentage of these devices are using an outdated or vulnerable version of Android [14]. The Android platform has gone through many updates and improvements over the last several years and due to a variety of factors, many of these devices still are not regularly updated to the most recent version. This continues to present a significant risk, because even though new versions of the operating system are introduced regularly, most devices are not updated for a long period of time due to the manufacturer and carrier-specific software deployment schedules. This leaves some Android devices vulnerable to exploits that have already been patched in newer versions of the operating system [12].

This paper will discuss a hands-on laboratory exercise designed to instruct students on the basic skills necessary to modify or develop their own Metasploit module. In addition to this exercise, this paper will also review related work in the field of computing security education as well as experiences and methods that were experienced during the development of a Metasploit module to make an existing vulnerability easily repeatable.

There are many great resources that can assist students in the process of writing a Metasploit module, including [11]. Using these references, in conjunction with a laboratory exercise, students will acquire a basic understanding of the process they can build on if needed in the future. This process further develops and encourages the skills needed for students to discover new vulnerabilities in software and devices, which is a benefit to the entire industry. The addition of Metasploit module development to a computing security curriculum is one way academic institutions can ensure their graduates have an industry-relevant education and that they are well prepared for the challenges they will encounter in their careers. The purpose of this paper is not only to provide an example hands-on lab format that can be incorporated into existing computing security curricula but to also serve as an overview for individuals looking to begin developing their own Metasploit modules. An overview of the steps involved in creating a Metasploit module is shown in Figure 1.

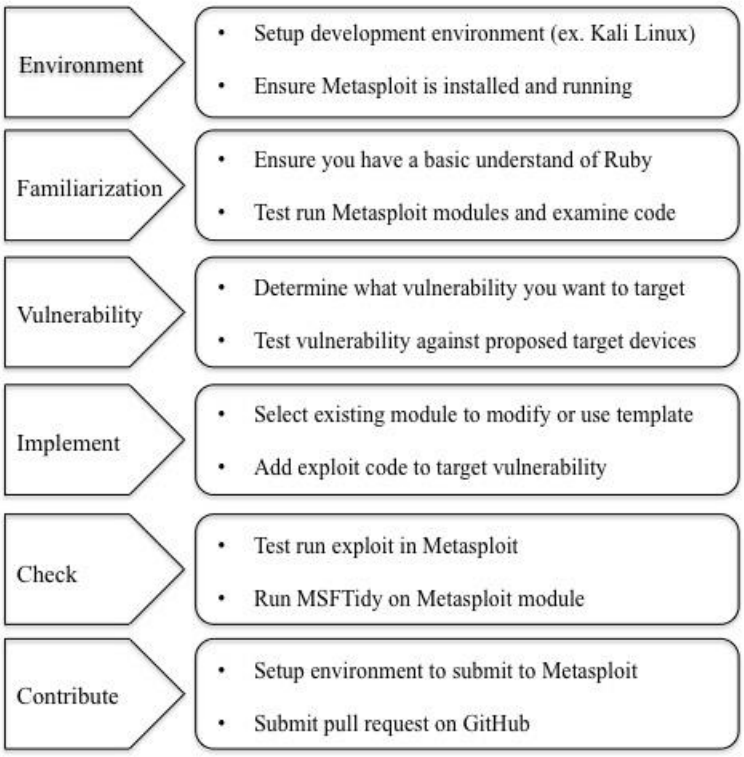

Figure 1: Steps to Develop Metasploit Module

3. Android Metasploit Modules

As of late 2015, the Metasploit framework contains only a small number of modules specifically designed to exploit Android devices. The quantity of Android-focused modules has been growing over the past few years, especially as new critical vulnerabilities have been discovered. The ability to develop or modify an existing Metasploit module for your own specific use is especially useful when there is an existing vulnerability that has been discovered and there is a need to replicate it numerous times for testing purposes. This would be useful to test a known vulnerability against a variety of Android versions, for example.

4. Hands-On Lab

In order to overcome the challenges described previously and many others seen by computer security professionals, colleges and universities need to continually update and improve their computing security curricula. One way that institutions have increased the relevance of the computing security programs is by incorporating realistic hands-on lab modules. These labs expose students to real-world problems and build skill sets that

will be valuable after graduation [3]. These exercises are also very useful for ensuring the student has taken the opportunity to try the skill on their own after learning the theory behind a given topic. This will increase the likelihood that the graduate is more proficient at the skill as opposed to simply seeing it demonstrated. If a computing security course only provides instruction on the development of a Metasploit module without a hands-on lab exercise, the student will have a much more difficult time developing proficiency in the skill due to the lack of opportunity for students to set up the environment, write and test modules, and troubleshoot issues on their own.

The primary goal of integrating basic Metasploit module development and testing into a computing security curriculum is to provide sufficient guidance for students to come away from the exercise with an understanding of the components and have the ability to create or modify their own modules. The student can pursue advanced development at a later time if needed, as this outline is intended to introduce the topic. One of the most time-consuming parts of this process is to set up the environment for module development. If this part is done prior to the exercise by having a virtual machine template with an environment already setup, it drastically reduces the time spent by instructors and students ensuring the environment is fully functional. The outline below in Table 1 is an example of a hands-on lab exercise that can be integrated into an existing computing security course. Full documentation of the exercise and implementation are intended as future work beyond the scope of this paper.

Lab #	Topics	Format
1	• Metasploit introduction • Development environment setup • Verification of environment setup	Instructor-led demonstration and student hands-on exercise
2	• Introduction to Ruby • Metasploit module review	Hands-on lab exercise
3	• Simple modification of existing module • Testing of existing module • Troubleshooting module	Instructor-led demonstration and student hands-on exercise
4	• MSFTidy use and troubleshooting	Hands-on lab exercise
5	• Metasploit module submission options	Hands-on lab exercise

Table 1: Outline of Metasploit Module Lab Exercises

Hands-On Lab Descriptions

- **Introduction and Environment Setup:** This module covers an introduction to the Metasploit framework for those that have not used it before. To minimize the environment setup and troubleshooting time, it is recommended that virtual machines running Kali Linux [16], which has Metasploit already installed, be available to students. The goal of this lab is to introduce students to Metasploit and get an understanding of all that needs to go into setting up the environment.

- **Ruby and Module Review:** This portion of the lab uses excerpts from the many resources that are available to give students an introduction to the Ruby programming language, which is the language used to write Metasploit modules. A good example tutorial can be found in [19]. The second part of this lab will identify the major parts of a simple Metasploit module. The goal of this lab is to begin to get students familiar with how a module looks and the language used to write them.

- **Modification and Testing of Existing Module:** This lab will walk students through the process of placing an existing module into a local directory in the environment where they can modify and test it in Metasploit. Students will be asked to test the module against other virtual machines as specified. The primary goal of this lab is to familiarize students with running modules locally and how to make minor modifications and test them in their environment.

- **MSFTidy:** MSFTidy [13] is a tool that should be run against a Metasploit module to ensure it meets syntax standards and other best practices set forth in the Metasploit framework. It is a reasonably easy tool to use and it is beneficial to students in debugging and finalizing their modules, whether they choose to submit them to Metasploit project or not. If a student chooses to contribute their module to the Metasploit framework, this tool must be run and any corrections need to be made prior to submission [18].

- **Metasploit Module Submission:** In the event that students want to submit modules for inclusion into the Metasploit framework, they will need to know the process for submitting a pull request and working with the Metasploit Development Team to have their exploit verified and merged into the master repository. This lab walks students through the ways that a module can be submitted for review, whether it be through the GitHub website or directly from their development environment.

5. Module Development Experiences

For someone who is new to the Metasploit framework, obtaining a high-level understanding is relatively straightforward as there is a significant amount of documentation on the Internet about how to use the framework and its modules. A background in the following topics is recommended, as it makes the process of developing a Metasploit module more manageable, but not required: computing security, Linux, scripting, and an understanding of how exploits work.

After Metasploit is set up in your environment, the modules are not difficult to run. Sometimes it is challenging to find an exploit that fits your needs, but there is an easy *search* command to which will show a description of the module and its location so you can use it. The options available while running most modules are self-explanatory with the *show options* command that is also available [15]. Metasploit modules can be selected separately based on the needs of the exploit that you're attempting to target.

Kennedy et al. provide a valuable resource in [11] for learning how to use Metasploit and beginning to write your own modules. As part of the research for this project, the authors went through the process of taking a publicly available vulnerability and creating a module that is able to easily replicate it. Using an existing vulnerability to create a Metasploit module is a favorable situation for the person who discovered the vulnerability, the module author, and the Metasploit Community as long as proper permission and citations are utilized. The authors chose an Android vulnerability because of the rapid growth that the industry has seen in this area over the past several years. There have been a number of serious vulnerabilities in the Android operating system that have since been resolved through updates. As an example, in their paper, Hamandi et al. introduce a number of attacks on the messaging system within Android and suggest a way they could be remediated [2].

While going through the process prior to writing this paper, the authors found assembling the pieces was not an intuitive task. Working with a mentor who has experience developing Metasploit modules and submitting items to GitHub [17] will provide valuable insight into the process and how to put all the pieces together. The authors also found that the Metasploit Development community will provide helpful feedback after submitting a module for inclusion into the framework. This proves useful in a situation where a student, who is not intimately familiar with Metasploit module development, has extraneous inclusions in their module that they are not aware of. Figure 2 shows an example Metasploit module in sections.

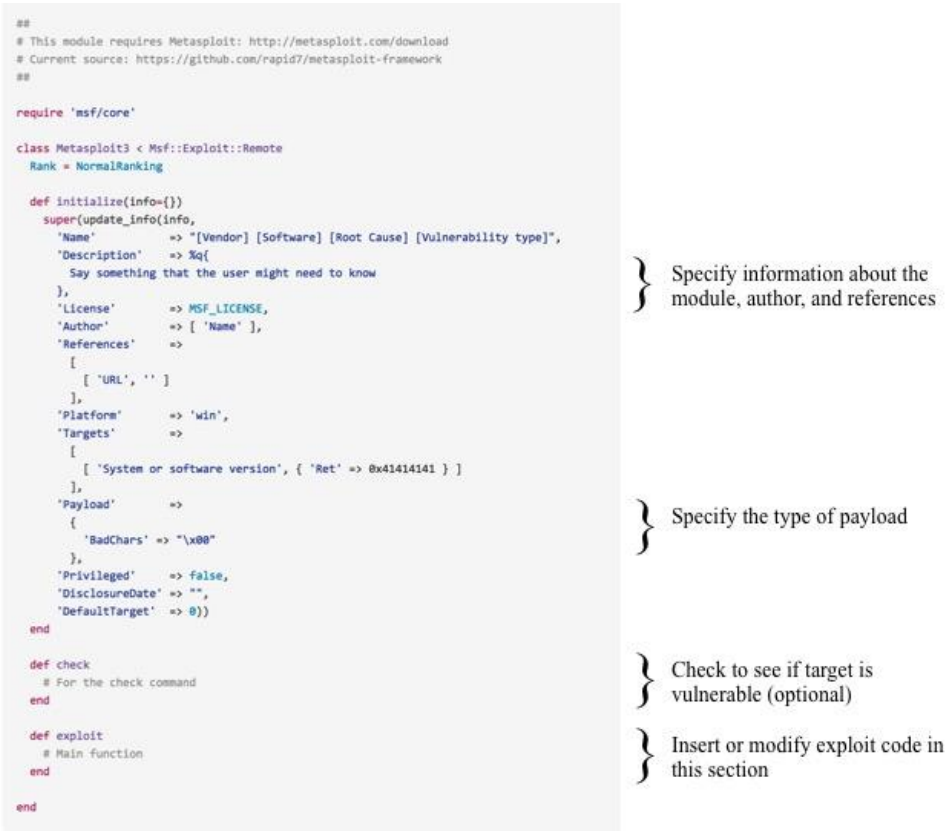

Figure 2: Major Sections of a Metasploit Module [20]

6. Related Work

There are a number of related papers that describe different ways to enhance computing security curricula in educational institutions as well as in industry. Some of these works are directed at increasing the overall computing security knowledge of students who may not be in a highly technical program [1], while others focus on integrating it into a very technical program such as Computer Science or Computer Engineering [6].

Weiss et al. describe the growing demand for well-trained computer security professionals in the industry. Their position is that the way to meet this demand is to integrate computing security topics into existing courses. They state that students learn best when they are exposed to hands-on experience [4]. The same challenges that face the industry also impact computing security curricula. Given the pace at which everything from hardware technology to malware is changing, it makes keeping course content and lab material current a challenge. In [4], the authors give the example that tools or specific malware a student may have learned about as a freshman at a college or university may be obsolete by the time they graduate. This is one strong attribute of the Metasploit framework, as malware changes, new modules are developed or existing modules can be modified as needed. Students that learn the basic framework of developing or modifying a Metasploit module will be able to use that skill well into the future. The authors tested their methodology by teaching two independent classes at different institutions and one of the labs included using Metasploit to exploit a Windows XP machine. This Metasploit lab was one of the top three highest rated for student interest [4].

Echoing the importance of hands-on exercises in computing security education are Wang et al. who believe that they provide a three main benefits: they expose students to real world challenges, they help students consolidate knowledge and gain in-depth understanding of the material presented in class lectures, and they help students to be well prepared for their careers in the industry [3]. All three of these benefits are solidly provided for with the incorporation of Metasploit module development hands-on exercises into a curriculum. Instead of students only learning how to use Metasploit, which the authors believe many do not find very challenging, they are asked to take a more in depth look at the framework and how it works. The benefits of this are well aligned with the authors' beliefs that this will build a more comprehensive understanding and better prepare them for a career in the computing security industry.

Another aspect discussed in [3] is the typically large amount of time that is required to setup and troubleshoot individual environments for students to use. A standardized lab environment is crucial to the success of integrating new topics into existing computing security curricula.

In [6], the authors discuss the CEeST project, which aims to empower computer science faculty to integrate security concepts into their curriculum to improve the security content of the provided education. Metasploit module development is also well suited to integrate into computer science curricula for several reasons. Firstly, computer science students and faculty may already be familiar with the Ruby programming language from other coursework. If they are not already familiar, they are likely keenly aware of how to pick up new programming languages quickly as most computer science coursework

requires students to use many different programming languages. Additionally, as computer science students and graduates are developing software with secure coding techniques in mind, having the ability to write or modify custom Metasploit modules to test their software could be significantly valuable.

Idziorek et al. state that while fundamental aspects of security are addressed as part of a general computer engineering or computer science curricula, these topics are often not covered in sufficient depth. To address the increasing cybercrime epidemic, they outline a three-tiered framework to improve computing security education. This framework includes formal literacy-based training for students of all backgrounds, inquiry-based learning through security and technically focused student groups and activities, and finally classical technical based initiatives [1]. The authors do not believe traditional course and lab-based learning on their own are the only pieces of the framework necessary to develop successful computing security curricula. Incorporating Metasploit module development is certainly aligned with one of the tiers in that it increases technical skill level and, therefore, better prepares the student for their future career.

The LOST project discussed by Abella et al. is an eLearning environment that helps educators to teach hands-on computer security training. One relevant aspect to this paper is the test bed that is discussed by the authors. In order for Metasploit module development to be integrated without the time involved becoming prohibitive, a consistent lab environment is required similar to the testbed described in [7]. The authors use virtualization to provide individual environments to students without requiring large infrastructure resources. This also enables the environments to be duplicated and modified easily to further reduce the setup time required.

Another area of the computing security curriculum that would benefit from integrating Metasploit module development is critical infrastructure and control systems. [5] and [8] both discuss their need to better prepare graduates and the existing workforce for continuously changing threats that not only face the entire industry, but especially critical control systems. They state that many industrial controls are not prepared to defend against attacks that could be carried out, especially if connected to a network that has access to the Internet. Foo et al. describe the different courses in the SCADA Security Curriculum and their intended audience. The one most related to this paper is the penetration-testing course, which is intended for those who will be performing testing on control systems to determine vulnerabilities against a cyber attack. They continue on to say that another target audience of this course are managers who will oversee those doing penetration testing on the systems as it raises awareness about the process and what they should expect [5]. While this course focuses on known vulnerabilities, rather than discovering new ones, the use of Metasploit and the ability to modify a module to fix a specific situation would certainly be a beneficial tool for the penetration tester.

In Red and Blue Team exercises conducted as part of the Critical Infrastructure and Control System (CICS) curriculum in [8], Metasploit modules and other methods were used to take advantage of vulnerabilities in the control systems and escalate privileges, obtain user passwords, collect asset information, and gain access to the hardware.

The related work described above covers a wide range of similar examples where incorporating the basics of Metasploit module development into a curriculum may add

significant value. Using a hands-on lab exercise, like the one outlined in this paper, would improve many curricula in the different disciplines of computing security education and will better prepare students for the challenges faced in the industry.

7. Conclusion

This paper outlined a hands-on lab exercise for integrating Metasploit module development into existing security curricula along with the reasoning and process associated with it. Educators will likely be more willing to integrate this into an existing course, rather than creating a separate course, as there are many other important topics to cover in computing security curricula. The related work outlined in this paper supports the assertion that there is a need for enhanced and continuing education in the computing security industry. Colleges and universities must continue to evolve and improve their curricula to ensure that their graduates are well prepared for the constantly changing threats that they will face in the industry.

There are no competing financial interests associated with this paper.

8. References

1. Idziorek, Joseph, Julie Rursch, and David Jacobson. "Security Across the Curriculum and Beyond." *Frontiers in Education Conference (FIE), 2012*. IEEE, 2012.

2. Hamandi, Khodor, et al. "Messaging Attacks on Android: Vulnerabilities and Intrusion Detection." *Mobile Information Systems* 2015 (2015).

3. Wang, Xinli, Yan Bai, and Guy C. Hembroff. "Hands-on Exercises for IT Security Education." *Proceedings of the 16th Annual Conference on Information Technology Education*. ACM, 2015.

4. Weiss, Richard, Jens Mache, and Erik Nilsen. "Top 10 Hands-on Cybersecurity Exercises." *Journal of Computing Sciences in Colleges* 29.1 (2013): 140-147.

5. Foo, Ernest, Mark Branagan, and Thomas Morris. "A Proposed Australian Industrial Control System Security Curriculum." *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, 2013.

6. Siraj, Ambareen, et al. "Integrating Security in the Computer Science Curriculum." *ACM Inroads* 6.2 (2015): 77-81.

7. Abella, Jaume, Guiomar Corral, and Agustin Zaballos. "LOST Project, a Learning Platform for Security Training." *Computers in Education (SIIE), 2012 International Symposium on*. IEEE, 2012.

8. Luallen, Matthew E., and Jean-Philippe Labruyere. "Developing a Critical Infrastructure and Control Systems Cybersecurity Curriculum." *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, 2013.

9. Smith, Ms. "127 Devices Added to the Internet Each Second, but Congress Is Clueless about IoT." *Network World*. 1 July 2015. Web. 28 Dec. 2015. <http://www.networkworld.com/article/2942596/microsoft-subnet/127-devices-added-to-the-internet-each-second-but-congress-is-clueless-about-iot.html>.

10. Foster, Richard. "The next Generation." Virginia Business. 30 Nov. 2015. Web. 28 Dec. 2015. <http://www.virginiabusiness.com/news/article/the-next-generation1>.

11. Kennedy, David, et al. *Metasploit: The Penetration Tester's Guide*. No Starch Press, 2011.

12. Thomas, Daniel R., Alastair R. Beresford, and Andrew Rice. "Security metrics for the android ecosystem." Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices. ACM, 2015.

13. "Msftidy." GitHub. 15 Apr. 2014. Web. 28 Dec. 2015. <https://github.com/rapid7/metasploit-framework/wiki/Msftidy>.

14. "Dashboards." Dashboards - Android Developers. 7 Dec. 2015. Web. 28 Dec. 2015. <http://developer.android.com/about/dashboards/index.html>.

15. "Penetration Testing Software, Pen Testing Security | Metasploit." *Metasploit*. Web. 28 Dec. 2015. <http://metasploit.com/>.

16. "Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution." *Kali Linux*. Web. 28 Dec. 2015. <https://www.kali.org/>.

17. "Build Software Better, Together." *GitHub*. Web. 28 Dec. 2015. <https://github.com/>.

18. Weidman, Georgia. *Penetration Testing: A Hands-On Introduction To Hacking*. No Starch Press, 2014.

19. "Ruby Tutorial." *www.tutorialspoint.com*. Web. 28 Dec. 2015. <http://www.tutorialspoint.com/ruby/>.

20. "Rapid7/metasploit-framework." *How to Get Started with Writing an Exploit*. GitHub. Web. 28 Dec. 2015. <https://github.com/rapid7/metasploit-framework/wiki/How-to-get-started-with-writing-an-exploit>.