IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Issue-1 E-ISSN: 2347-2693

Information Gathering on a Web Application deployed in Ruby on Rails

Harsh Bhardwaj[1*], Manish Aggarwal[2] Neha Gupta[3]

[1*,2,3]*Department of Computer Science and Technology*
Maharaja Agrasen Institute of Technology,
New Delhi, India

www.ijcseonline.org

Received: Dec/13/2015 Revised: Dec/22/2015 Accepted: Jan/09/2016 Published: Jan/30/ 2016

***Abstract*—** In this world of providing effective interface to the user for accomplishing the requirements needed to perform information gathering for the purpose of implementing Penetration testing in a network we need an adaptive scenario of carrying out the same task. Ruby on Rails provides an interactive way of dealing with the user's inputs. This kind of Web application allows a user to perform the basic information gathering, regarding possible threats in its network without having prior knowledge of Penetration testing.

***Keywords*—**Penetration Testing, Ruby on Rails, Information Gathering

I. INTRODUCTION

A penetration test can be defined as the process of systematically and actively testing a deployed network to determine what vulnerabilities may be present and to create a report with recommendations to mitigate or resolve these vulnerabilities [4].

Information gathering is the basic step toward penetration testing. This step is carried out in order to find as much information about the target machine as possible. The more information we have, the better our chances will be of exploiting the target. During the information gathering phase, our main focus is to collect facts about the target machine, such as the IP address, available services and open ports.

There are basically three types of techniques according to [5], used in information gathering:

- Passive information gathering
- Active information gathering
- Social engineering

Passive information gathering is used to gain the information about the target without having any physical connectivity or access to it. The tools required to perform this task can be given as whois query, Nslookup and so on.

Active information gathering provides us with the next level of information, which can directly supplement us in our understanding of the target security. This is done by setting up a logical connection with target in order to gain information.

Social engineering is similar passive information gathering but realize on human error and the information leaked out in the form of printout, telephone conversations, incorrect email Ids etc [11].

Rails is a framework for building websites. As such, Rails establishes conventions for easier collaboration and maintenance. Rails, in a larger sense, is more than a software library and an API. Rails is the central project of a vast community that produces software libraries that simplify the task of building complex websites [9].

II. RUBY MVC ARCHITECTURE

A. *The Model*

- Contains data for the application(often linked to a database)
- Contains state of the application
- Contain all business logic
- Notifies the view of state changes
- No knowledge of user interface, so it can be used

B. *The View*

- Generates the user interface which presents data to the user
- Passive i.e. doesn't do any processing
- View works when the data is displayed to the user
- Many views can access single model for different reasons

C. *The Controller*

- Receive events from outside world through views
- Interact with the model
- Display the appropriate view to the user

III. TERMINAL TOOLS

A. NMAP

Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single host.

-sP : ping scans the networks
-sL : List Scan- simply lists targets to scan
-sC : script scan
-p : full TCP scan using with service version detection
-f : by-pass firewall
-O : Operating system detection
-A : enable OS detection, version detection, scripting scanning and traceroute
-Pn : scan without ping

B. whois

A **whois** search will provide information ragrding a domain name, such as www.google.com. It may include information, such as domain ownership, where and when registered, expiration date and the name servers assigned to the domain.

-h : Connect to whois database host
-p : When connecting, connect to network port
-H : Suppress the display of legal disclaimers

C. Traceroute

Traceroute is a utility that records the route (the specific gateway computers at each hop) through the Internet between your computer and a specified destination computer. It also calculates and displays the amount of time each hop took.

-4 : allows to send IPv4 only

-6 : allows to send IPv6 only

D. IP Calculator

ipcalc provides a simple way to calculate IP information for a host. The various options specify what information **ipcalc** should display on standard out. Multiple options may be specified. An IP address to operate on must always be specified. Most operations also require a netmask or a CIDR prefix as well.

-c, --check : Validate the IP address under the specified family. If no address family is specified, IPv4 is assumed.
-4, --ipv4 : Specify IPv4 address family
-6, --ipv6 : Specify IPv6 address family.
-b, --broadcast : Display the broadcast address for the given IP address and netmask.
-h, --hostname : Display the hostname for the given IP address.
-m, --netmask : Calculate the netmask for the given IP address. It assumes that the IP address is in a complete class A, B, or C network. Many networks do not use the default netmasks, in which case an inappropriate value will be returned.
-p, --prefix : Show the prefix for the given mask/IP address.
-n, --network : Display the network address for the given IP address and netmask.
-s, --silent : Don't ever display error messages.

E. IP Locator

curl is a client to get documents/files from or send documents to a server, using any of the supported protocols (HTTP, HTTPS, FTP, GOPHER, DICT, TELNET, LDAP or FILE). The command is designed to work without user interaction or any kind of interactivity.
curl offers a busload of useful tricks like proxy support, user authentication, ftp upload, HTTP post, SSL (https:) connections, cookies, file transfer resume and more.

$curl ipinfo.io/#{@url} : gives the geographic location of the server

F. nslookup

Nslookup stands for Name Server LOOKUP is a useful tool for finding out information about a named domain. By default, nslookup will translate a domain name to an IP address or vice-versa.

-type=ns : lists all name servers
-type=mx : lists mail exchange servers for a domain
-type=soa : start of authority

IV. USING COMMANDS ON RUBY ON RAILS

A web application is required to have the ability of handling user requests in order to make it advanced enough to perform the tasks that are usually performed on Linux terminal. A GUI based interface that becomes possible with deploying Ruby on Rails can also be used to perform various terminal based tasks that give real-time results.

ERB (Embedded RuBy) is a feature of Ruby that enables

you to conveniently generate any kind of text, in any quantity, from templates. The templates themselves combine plain text with Ruby code for variable substitution and flow control, which makes them easy to write and maintain [9].
Although ERB is most commonly seen generating Web pages, it is also used to produce XML documents, RSS feeds, source code, and other forms of structured text file. It can be extremely valuable when you need to create files which include many repetitions of a standard pattern, such as unit test suites.

In order to run terminal based commands, the following manner can be adopted:

- make a rails app using the command:
$ rails new WebApp

- make a controller that will hold the terminal commands and execute them on the server
$ rails generate controller Commands new show

- in the above command, two webpages with name 'new' and 'show' will be created. 'new' webpage will hold the GUI for taking inputs from the user in order to perform various tasks on the server side. On the other hand, 'show' webpage will reflect the result of the command executed at the backend.

'New' in Views

- open the webpage 'new' located in WebApp/app/views/new.html.erb, in order to create a form that will bring in the desired value of inputs. Create a form_tag, mention the action as 'show' and controller as 'commands'. This can be done as follows:
<%= form_tag(:action => 'show', :controller => 'commands) do %>

- also mention a text_field_tag that will GET the value of the choice based on the numbering made for various available choices. This can be made to be done as:
Choose an Option<%= text_field_tag 'name' %>

- another text_field_url is required to GET the url or the IP address from the user. This can be mentioned as:
IP Address/Domain Name<%= text_field_tag 'url' %>

- a submit_tag can be used to perform the HTTP GET/POST action on the form. This is done as:

<%= submit_tag 'Submit' %>

- Above syntax used in the webpage describes the variables that hold the value entered by the user. Option field uses the variable 'name' that can be used in the controller section of the web application to perform various tasks on the same. Similarly, 'url' variable is required to hold the Domain name or IP address of the victim that we required to be pentested upon.

Controller

- The controller 'commands' is used to perform the basic backend task of executing the terminal commands and sending the result to the webpage named as 'show'.

- make a method 'show' in the CommandsController class locate in WebApp/app/controllers/commands_controller.rb
def show
end

- introduce an instance variable '@choice' that holds the choice value of the user entered in the choice text box on the webpage. The value entered by the user can be retrieved using 'Params' as:
@choice = params[:name]

- In the similar manner, Domain name or IP address can also be retrieved from the form as:
@url = params[:url]

- Create a switch-case statement, that deploys its cases on the basis of choice retrieved from the user:

```
case @choice
when '1'
        #perform something
when '2'
        #perform something
else
        @display = "WRONG INPUT"
end
```

- On the basis of what is being entered by the user, the comment mentioned above can be replaced with the same. For example, if @choice = 1 corresponds to performing PING on a server, then the following statement can be replaced by the comment:
@display = `ping -c 2 #{@url}`

- It should be noted here that @display is the variable that holds the output of the above command. This variable will be used in another webpage to display the results.
'Show' in Views

- open the webpage 'show' that will display the results in HTML format

- The simple text of the @display variable can be used in the 'show' webpage to display the ouput. This can be given as:

<%= (0..(@display.split("\n").length)).each do |i| %>

<%= @display.split("\n")[i] %>

ACKNOWLEDGMENT

We would like to thank our parents for providing us all sorts of means to have a considerable environment for proper studies. We would also thank the Almighty God, for all the protection and care he has been giving to us. A special thanks to our guide Dr. Namita Gupta and Mr. Alok Sharma, for extending their extreme support in writing this paper. We would also extend our extreme gratefulness to the authors who have published their materials with Open-Source Community.

REFERENCES

[1] An Overview of Penetration Testing, International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.6, November 2011
[2] Why Johnny Can't Pentest: An Analysis of Black-box Web Vulnerability Scanners, University of California, Santa Barbara
[3] Improving penetration testing through static and dynamic analysis, Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/stvr.450
[4] State of the Art: Automated Black-Box Web Application Vulnerability Testing, Stanford University
[5] PENETRATION TESTING AND VULNERABILITY ASSESSMENTS: A PROFESSIONAL APPROACH, Published in the Proceedings of the 1st International Cyber Resilience Conference, Edith Cowan University, Perth Western Australia, 23rd August 2010
[6] Penetration Testing: Assessing Your Overall Security Before Attackers Do, SANS Institute InfoSec Reading Room
[7] Arkin, B., Stender, S., McGraw, G. (2005). "Software Penetration Testing", IEEE Security and Privacy, Volume 3, Issue 1
[8] Network Penetration Testing and Research, Brandon F. Murphy North Carolina Agricultural and Technical State University, Greensboro, North Carolina, 27411
[9] Ruby on Rails Tutorials 3rd Edition – Michael Hartl, 2nd Edition, Addison-Wesley Professional Ruby Series
[10] Certified Ethical Hacker – Kimberly Graves, 1st Edition, Wiley Publising Inc.
[11] Core Security Technologies, http://www.coresecurity.com/content/intro-pen-test
[12] Hacking Articles by Raj Chandel, http://www.hackingarticles.in/

AUTHORS PROFILE

Name: Harsh Bhardwaj
Education: Pursuing B. Tech. in Computer Science and Engineering from Guru Gobind Singh Indraprastha University, New Delhi
Certifications : Certificate of Accomplishment in Usable Security, Data Mining, Internet Security
Interests : Python Programming, Ruby Programming, Ruby on Rails, Algorithm Design and Analysis, System Secutiry, Cyber Security

Name: Manish Aggarwal
Education: Pursuing B. Tech. in Computer Science and Engineering from Guru Gobind Singh Indraprastha University, New Delhi
Certifications: Certificate of Participation in Ethical Hacking
Interests: SQL, Ruby on Rails, Java, Computer Networking

Name: Neha Gupta
Education: Pursuing B. Tech. in Computer Science and Engineering from Guru Gobind Singh Indraprastha University, New Delhi
Interests: PHP, CSS, C++, SQL