RDOS TESTER USER'S GUIDE

09 May 2014

Revision 1.4

Team C

Jamie Lane, Bradley Norman, Daniel Ross

Revision History

Date	Revision	Description
04/23/2014	1.0	Initial draft
05/05/2014	1.1	Added cover page and revision history Changed format to match related documents Expanded introduction Added installation instructions Updated GUI screenshot Updated execution instructions
05/07/2014	1.2	Updated operation instructions
05/08/2014	1.3	Added screen shots for finding IP and MAC addresses
05/09/2014	1.4	Added References

1. Introduction

1.1 Purpose

RdosTester will generate the type of traffic but not the volume of traffic necessary to initiate a specific instance of an amplified, reflected denial-of-service attack (RdoS). This attack is a variation of a documented Quake3 DoS attack, that was mitigated in 2010 (McVittie, 2012).

1.2 Background

RDoS attacks require at least 3 hosts. The hosts can be individual computers, networks, servers, or anything with a network connection and the capability to send and receive traffic. For this description (and nowhere else), the hosts will be the Attacker, the Mirror, and the Victim. The Attacker initiates the attack by sending requests to the Mirror. The Attacker has altered the requests so that they include the IP (Internet Protocol) address of the Victim, as their source address. When the Mirror responds to the Attacker's requests, it responds to the Victim, having been fooled by the Attacker's false source address. If the Attacker generates sufficient requests, the Mirror will overwhelm the Victim with responses, making it unavailable to legitimate users. When the Mirror's responses are larger in size than the Attacker's requests, the attack can be described as amplified. By implementing an amplified RDoS, an attacker may generate more response traffic than it might on its own, as well as obscuring its identity from the victim.

1.3 Implementation

RdosTester demonstrates the amplified RDoS technique by crafting an Ethernet frame containing an IP / UDP (User Datagram Protocol) packet containing a "getinfo" request. The packet is based on a packet captured from an OpenArena game client as it requests status from an OpenArena game server. The source and destination addresses of the request are changed to values selected by the user. The request is then transmitted to the destination OpenArena server, which will respond to the user-specified source address. OpenArena servers respond to a variety of requests from game clients, all of which have an amplification factor. The "getInfo" request is utilized by RdosTester, as the server will respond to clients (and our application), which have not established game sessions. The server's response to "getInfo" requests has an amplification factor of around 400%. RdosTester is not intended to be an actual hacking tool, but a proof of concept. Without a proof-of-concept application available, developers may not be motivated to patch security vulnerabilities.

2. Requirements

2.1 Software Requirements

RdosTester is supported on Microsoft Windows XP SP1 and above (32/64-bit). RdosTester requires the installation of WinPcap version 3.1 or above. Execution additionally requires the system to have the Java Runtime Environment v7 (32-bit) installed, as well as the jNetPcap API v1.3. Administrative privileges may be required to execute RdosTester. Instructions on where to find the required downloads are covered in section 3.

2.2 Hardware Requirements

The minimum hardware specifications are determined by the OS hosting the JRE. To execute in Windows XP SP1, only 64 MB of ram and 1.5 GB of hard disk space are required. Supported processors include x86, x86-64, x64 and AMD64. Ethernet / IP networking hardware and network connectivity will be required to utilize RdosTester. Internet connectivity will be required to communicate with OpenArena servers not on the user's LAN.

3. Installation

3.1 Background

RdosTester is written in Java and runs on Microsoft Windows; however, neither Java nor Windows natively support the raw Ethernet access it requires to function. This limitation is overcome through the use of the WinPcap driver and the jNetPcap API. WinPcap is a native Windows network driver that allows software running in Windows to have raw access to the transmit and receive facilities of any Ethernet devices on the system. It allows similar access to Microsoft's virtual Ethernet representation of wireless networking devices. The jNetPcap API provides Java developers with libraries that allow Java programs to interface with WinPcap.

3.2 Java Runtime Environment version 7

Java Runtime Environment version 7 is required to run RdosTester and may be downloaded from the following location: http://www.oracle.com/technetwork/java/javase/downloads/. Installation instructions are available at the same location as the installer. For most users, they will download and execute the installer appropriate for their version of Windows.

3.3 WinPcap

Due to its use of jNetPcap, RdosTester requires WinPcap version 3.1 or above. As of May 2014, the current release of WinPcap is version 4.1.3. It can be obtained at the following location: https://www.winpcap.org/install/. To install WinPcap version 4.1.3, the user downloads and executes its universal installer. WinPcap is then ready for use.

3.4 jNetPcap

RdosTester requires jNetPcap version 1.3. The user can obtain jNetPcap at the following location: http://jnetpcap.com/download/. Download the Windows package appropriate for your OS, either 32-bit or 64-bit. Extract the downloaded package into a directory of its own. Once extracted, a variety of methods are available to make the API available to your system and are described in the jNetPcap release notes. The most straightforward method is to copy the jNetPcap DLL from its directory into the \Windows or \Windows\System32 directory. If you have received RdosTester as an executable JAR, jNetPcap and its license information will likely be included in the distribution package. RdosTester will be able to function as long as the file jnetpcap.dll is located in the same directory as RdosTester.jar. If the user intends to compile RdosTester from source, they need to follow the procedure for their development environment for making jnetpcap.jar available in their build path. Instructions for making jNetPcap available in Eclipse are located at the following location: http://jnetpcap.com/eclipse/.

3.5 RdosTester

If RdosTester was received as an executable JAR, then placing it in a directory with jnetpcap.jar is all that is necessary prior to execution. If the user wishes to compile and execute RdosTester from source, more steps are required. The RdosTester source is ccomprised of four classes, each in its own file. The files are Rdostester.java, RdosPacket.java, PacketTransmitter.java and Analysis.java. The file jnetpcap.jar must also be available, as described in section 3.4. Prior to executing the program, it should be compiled by entering *javac -cp /<jnetpcap install path>/jnetpcap.jar *.java* at a command prompt, while in the same directory as the source files. Once compiled, the program may be executed by entering *java -cp /<jnetpcap install path>/jnetpcap.jar teamC/RdosTester* at a command prompt one level higher than the source/package directory, *teamC*. For execution of the compiled source, jnetpcap.dll must be present in the operating systems path as described in section 3.4.

4. Operating Instructions

4.1 User Interface

Figure 1 is a screenshot of the RdosTester GUI. The GUI is divided into seven rows. The first four rows allow user input. Row five allows for selection of a network interface. Row six allows the user to transmit an “attack” packet. Row seven displays messages advising the user of invalid inputs as well as the status of packet transmission and receipt.

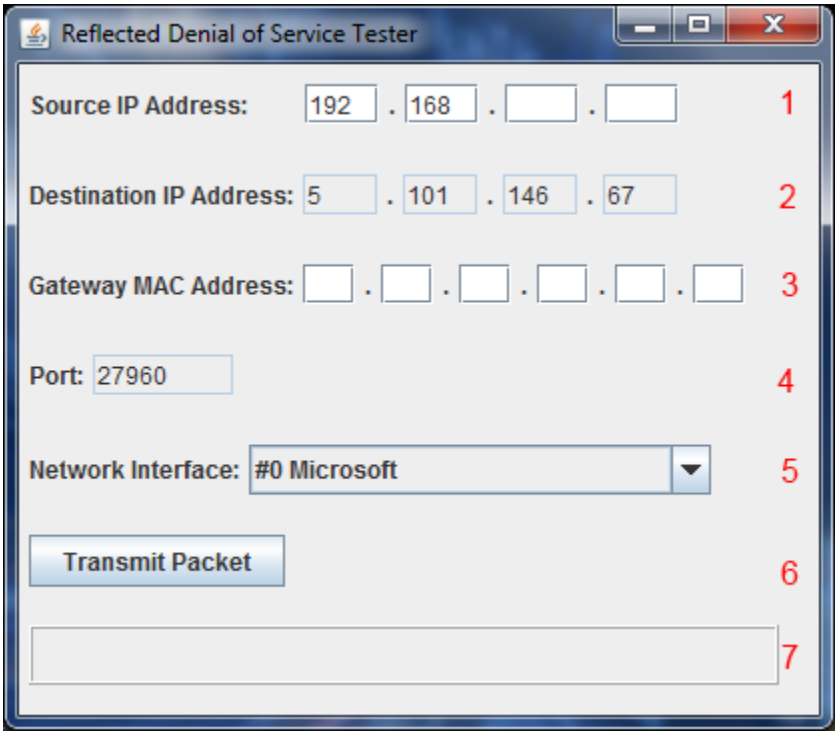

Figure 1. RdosTester GUI.

4.2 Operation

4.2.1 Open the Application

When the RdosTester JAR is executed by double-clicking, or the RdosTester application is compiled and launched as described in section 3.5, the GUI is displayed.

4.2.2 Enter Source IP Address into the GUI

In row 1, the user should enter their own IP address as the source. To find their IP address, the Windows user should type *ipconfig* in their command prompt and hit the enter or return key as shown in figure 2. The destination address and port of an OpenArena server, available for testing, is pre-populated in the destination and port fields, rows 2 and 5, and is not immediately editable by the user. To change destinations and ports, the user must compile RdosTester from source, changing the constant ALLOWANYSERVER in the RdosTester class file to “true”.

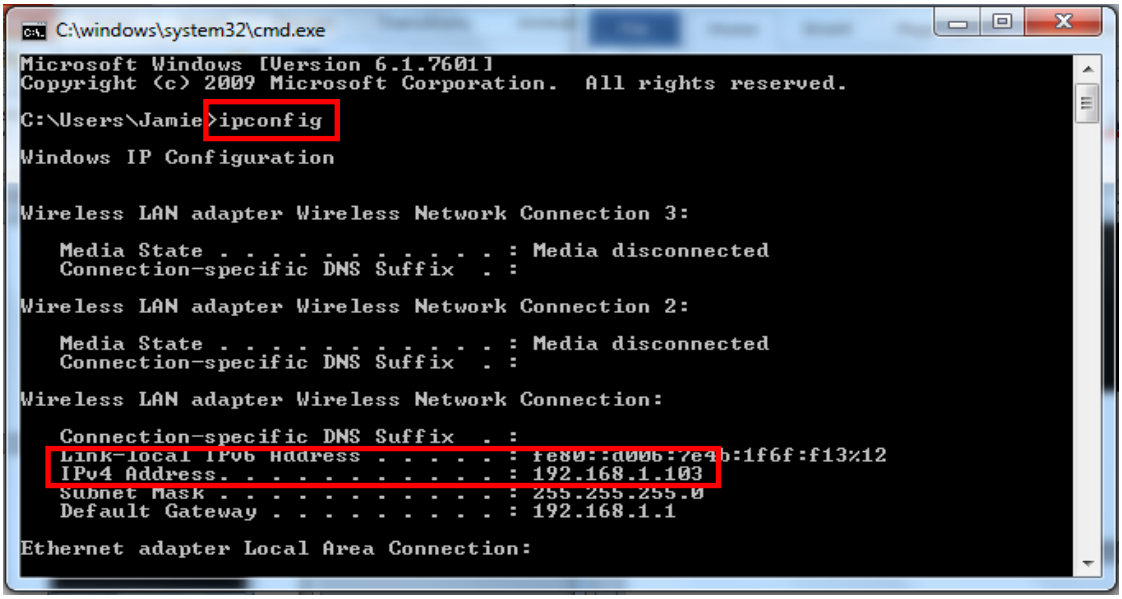

Figure 2. Find User's IP Address.

4.2.3 Enter Gateway MAC Address into the GUI

In row 3, the Gateway MAC address field must be filled by the user with the hex values for the MAC address of the user's gateway router. In Windows, the user can determine this by opening a command prompt, then typing *ipconfig | findstr /i "Gateway"* and then pressing the *Enter* or *Return* key on their keyboard. This will present the user with their gateway router's IP address. Next the user may enter *arp -a* and *Enter* or *Return* at the same command prompt. The user will find the MAC of their gateway, in the column labeled "Physical Address", on the same row as their gateway IP as shown in Figure 3. Input this MAC in row 3 of the GUI.

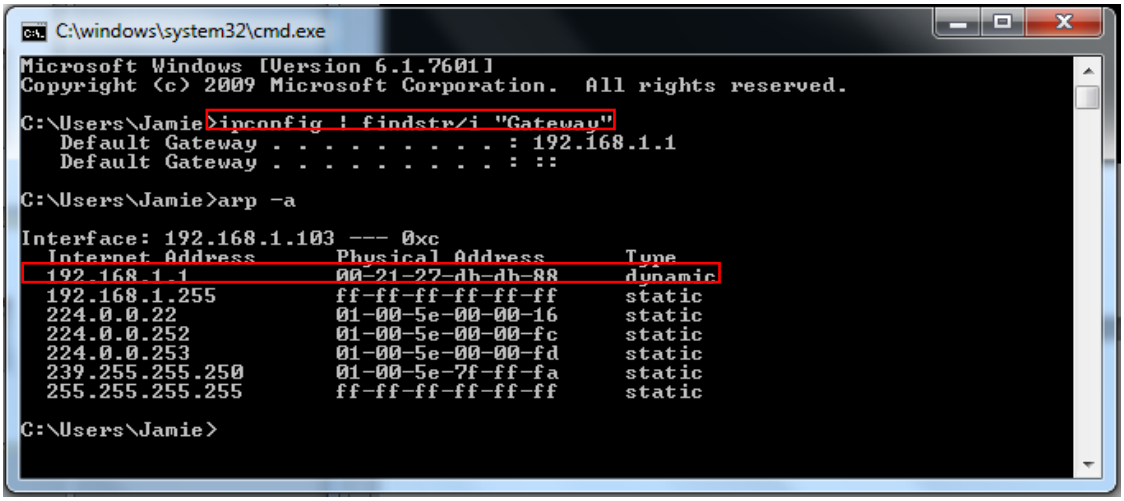

Figure 3. Finding the Gateway MAC Address.

4.2.4 Select Network Interface

The correct network interface must be selected from the drop-down box in row 5.

NOTE: If the user does not know the correct interface, they should select the first one, then click the transmit button in row 6. If the message “Packet not Transmitted” is displayed in row 7, the user should select the next interface from the drop-down box and click transmit packet. The user should continue selecting the interfaces until they reach one that transmits the packet.

4.2.5 Transmit the Packet

The user must click the “Transmit Packet” button. After the Transmit Packet button has been clicked, the application should now display, in row 7, the size of the packet received from the vulnerable OpenArena server. The size of the received packet is formatted as a percentage of the original packet size. If the exploit is successful, the received packet will be much larger than the original packet.

NOTE: If row 7 displays the message “Packet Transmitted. No Response from Server”, this indicates there is either a problem with the user selection of source, destination, port, gateway MAC, or possibly the user’s network, or in unlikely circumstances, the destination server.

4.2.6 Repeat as Desired

The user may repeat steps 4.2.1 through 4.2.5 with different network settings as desired.

4.3 Shutdown

To exit RdosTester, click the “X” in the upper right-hand corner of its display window.

5. References

McVittie, S. (2012, May 27). Quake 3 Denial Of Service. *Packet Storm*. Retrieved March 18, 2014, from http://packetstormsecurity.com/files/111240/Quake-3-Denial-Of-Service.html

McVittie, S. (2012, March 25). openarena-server: [CVE-2010-5077] traffic amplification via getstatus requests. *Debian.org*. Retrieved March 18, 2014, from https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=665656