■ ECE/CS 4984: Wireless Networks and Mobile Systems ■

Project P10 – A Wireless LAN Hot Spot

Part I – Objectives and Project Materials

Objective

The objective of this project is to understand how routing, IP firewalls, and IP masquerading (also known as network address and port translation) can be integrated to offer wireless connectivity or "hot spot" service. Understanding of the following will result from this project.

- DHCP daemon use and configuration
- iptables use for basic firewalling and IP masquerading
- Configuring a laptop running Linux to work as a router
- Basic web authentication using a web interface

Hardware

You will need to use the following hardware for this project.

- Notebook computer
- iPAQ handheld computer
- Intel Wireless Gateway
- Two 802.11b network interface cards (NICs)

Software

You will need to use the following software for this project.

- Linux on the notebook computer
- iptables on the notebook computer
- DHCPd on the notebook computer
- Apache Web Server on the notebook computer
- CGI development language (C++, Perl) or PHP on the notebook computer
- Standard Internet Explorer under PocketPC on the iPAQ

Part II – Technical Specification

For this project, you are to design and implement a "hot spot" gateway and network. A hot spot is a wireless local area network (WLAN) commonly used in small businesses and public areas to offer convenient wireless connectivity to customers and visitors. WLAN access offered at a coffee shop is an example of a hot spot service. The following are major concerns in a hot spot environment.

1. Simplicity – a user should be able to gain access using a standard notebook or handheld computer with an 802.11b NIC and dynamic host configuration (i.e., using DHCP).
2. Security – only registered (paying or otherwise authorized) users should have access to the network.
3. Maintainability – since most restaurant employees aren't your average Linux guru; ease of administration is a must.

Congratulations, you are a consultant hired by Stu's Pizza Shack to build and install a hot spot for their pizzeria. The following is the network specification requested by the manager, Stu:

1. 802.11b Wireless access
2. Everything must work off of a single external IP address since they only have a cable modem
3. As simple for the customer to use as possible
4. Provide user authentication; only paying customers should have access.

Configuration Overview

To accomplish this task, you must use your Dell notebook computer running Linux and your Intel Wireless Gateway. For testing, you will use your Compaq iPAQ with an 802.11b NIC as the client. Figure 1 shows the network configuration for testing. Note that a "public Internet" will be provide by the GTA for testing your implementation. Also, your hot spot should be able to support additional wireless devices, as represented by the second iPAQ in Figure 1.

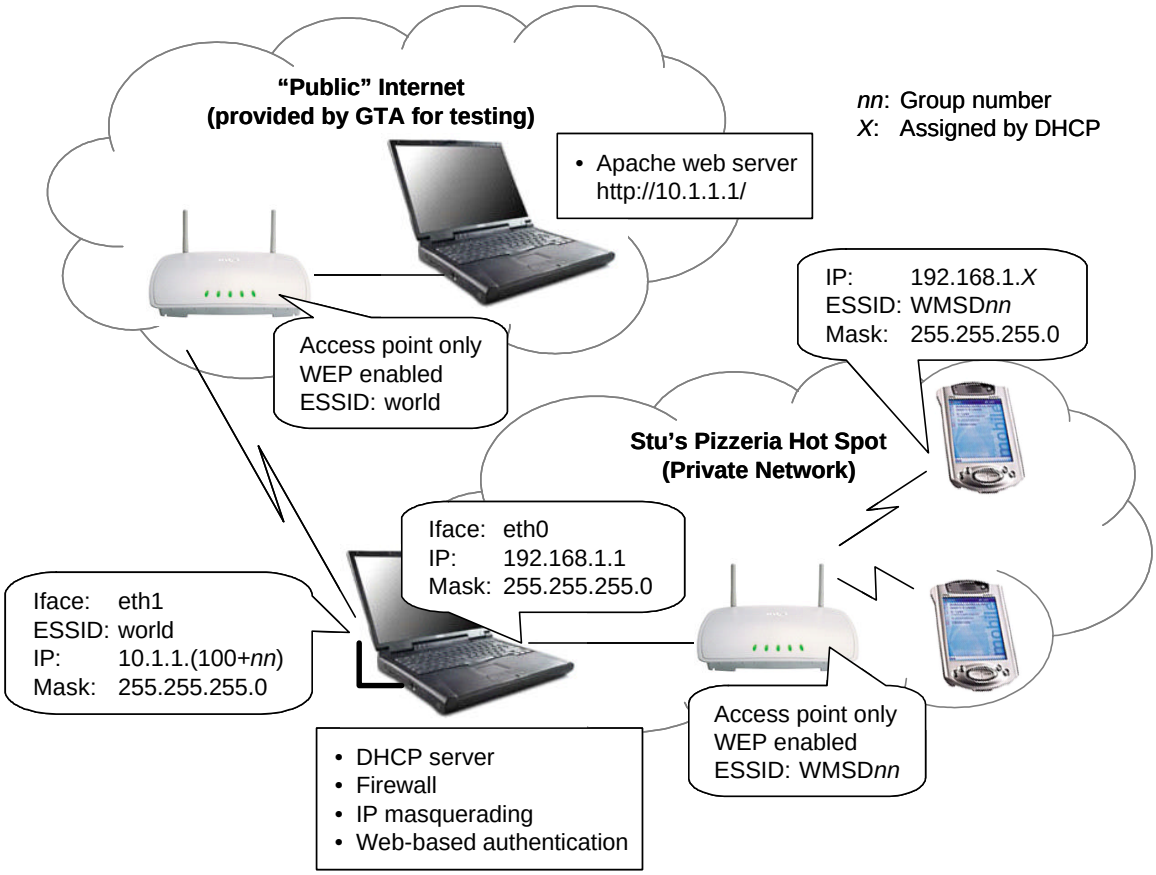

Figure 1. WLAN hot spot hardware configuration for testing.

The components shown in the configuration of Figure 1 are used as follows.

1. Intel Wireless Gateway (in the private network)

a. The gateway acts as an access point only. Disable all of the gateway functionality of the access point. DHCP client and server capabilities in the wireless gateway are not needed and must be turned off.

b. Configure the access point to use ESSID of WMSD*nn*, where *nn* is your group number.

c. The wireless client (the iPAQ) uses this as the access point to connect to the network.

2. Notebook Computer

   a. The notebook computer, running Linux, acts as the DHCP server, IP masquerading gateway, and authentication host. (Typically, a desktop PC or server would be used for this function, but we will use a notebook computer. Also, a single computer could provide services for multiple access points. For this project, we will use only one access point.)

   b. The DHCP server software of choice is the standard DHCP daemon, dhcpd, found with the Linux distribution on the notebook computer. (See /etc/dhcpd.conf for configuration.)

   c. The wired (internal or private network) interface has the address 192.168.1.1 with a 24-bit netmask. The wireless (external) interface has the address 10.1.1.<100+group number>, also with a 24-bit netmask.

   d. Iptables will be used for firewalling and IP masquerading.

   e. The notebook provides authentication, as described below. It runs the Apache web server to aid in this function.

   f. The notebook connects to the access point via the crossover cable. The access point will connect to the internal network. For ease of demonstration and testing, the 802.11b interface will act as the external interface to the public network.

3. iPAQ

   a. The iPAQ serves as a test client. Your service should work for multiple simultaneous users, although the available equipment limits your testing to one client. If possible, use another wireless client, e.g., another iPAQ or a notebook, to test your system.

Authentication

The notebook must perform authentication so that only recognized users may use the hot spot service.

1. The client should be able to access the hot spot without any modifications, so authentication must be done at the gateway (the notebook computer). Iptables on the gateway is used to allow or deny access to the outside world.
2. By default all clients must be blocked from accessing the outside world. This is done using iptables. Until authenticated, all client HTTP requests should be REDIRECTed (hint) to the authentication page on the gateway. All other protocol requests to the external network (except, optionally, DNS) should be blocked. No protocol request should be sent to the outside network before the client is authenticated. Once a client has successfully authenticated, the client will have access to the outside world.
3. The authentication page should be a document in the root directory of the web server running on the notebook computer. The user enters a username and password that is processed by the laptop. This can be done using a Common Gateway Interface (CGI) program implemented in Perl or C++ or a PHP program. An allow file on the gateway contains valid username and password pairs.

4. The “allow list” is a list of usernames and associated passwords that authenticate the users. It should be contained in a file, the allow file, “allow.txt”. Usernames and passwords are checked against this file for authentication purposes. This file may be stored as clear text and edited by hand for ease of creation (although this would not be a good approach for a production system).
5. The access point should have WEP enabled to protect against those not on the network. The SSID of the access point should be WMSD<group number>. Use 64-bit WEP with ABCDEF4984 as the key.

Firewall Configuration

1. Policies. The policy for the filter:INPUT, filter:FORWARD, and filter:OUTPUT chains is DROP. All other chains have an ACCEPT policy.
2. Ports. The laptop should not have any open ports to the outside world. The laptop should have DHCP and HTTP ports available to the inside and nothing else. ICMP connectivity is optional.
3. Rules. Rules will be added to all of the filter chains and the nat:PRE/POSTROUTING chains. These are the only chains that should have rules added.

Shell Scripts

1. Startup Script. The startup script should have all the components to initialize the interfaces, firewall, and services. The script should follow this order: shutdown network interfaces, clear the firewall, setup the firewall, bring up the network interfaces and start any needed services.
2. Shutdown Script. The shutdown script should set the firewall back to no firewall mode; all rules flushed, extra chains removed, and all polices set to ACCEPT. It should also shutdown the services needed for the project. It does not need to set the network interfaces back to their pre-project state.

Hints

1. Work to get DHCP and IP masquerading working before moving on to the authentication portion.
2. The web page root is: /var/www/html. The HTTP daemon configuration file is: /etc/httpd/conf/httpd.conf.
3. For your authentication scripts, iptables must be executed by root. Research `cron`, `sudo`, or `chmod` for ideas. (Running Apache as root is not an option since this is a huge security hole.)
4. You do not need to recompile the kernel for this project. The provided kernel has iptables support.
5. All of the servers (Apache web server and dhcpd) and development software (PHP, Perl, and C++) are already installed on your notebook. The DHCP configurations in dhcpd.conf will need to be modified. The web server configuration file, httpd.conf, will need to be modified slightly to support the authentication function.
6. The laptop’s address is 192.168.1.1 and the internal (private) network is 192.168.1.0/24.
7. Packet capture utilities such as ethereal and tcpdump can provide some help in debugging.
8. Functionality is the key. Make sure all your parts work together before you worry about aesthetics.
9. If you do not implement the removal of clients, make sure you flush (-F) iptables often. The firewalling chains have a size limit.

10. Test to make sure your firewall is working correctly. The 192.168.1.0/24 network should not be accessible from the outside world. No packets marked as being from 192.168.1.0/24 should be sent out onto the external network.
11. Backup any configuration files before modifying – just in case!

Optional Features

Here are some additional features that you may want to add. You can obtain full credit by just implementing the basic features. But, inclusion of any of these features will be considered in grading to compensate, at least partially, for any weaknesses in the project or report.

- Add an administration web page with the ability to add username/password pairs to the allow list.
- Add an administration web page with the ability to remove username/password pairs from the allow list.
- Add time limits to hosts on the allow list. A host has a time window in which they may authenticate and use the network. After a certain number of minutes the host's firewalling rule is removed and it is also removed from the allow list. (This is difficult.)
- Implement the ability to block certain IP addresses from successfully authenticating. Check the IP address as well as the username and password. A list of deniable IP addresses should be put into a deny file, "deny.txt".
- Implement a log file feature to record at least usernames, time of authentication, and host address. Any other significant information may also be included.

Part III - Demonstration

Each group must demonstrate their working hot spot service for the GTA during the period May 3-5. Demonstration times will be scheduled in Blacksburg and at the NVC. This demonstration will consist of the following steps.

- The team will bring their notebook computer, fully configured for the hot spot service, to the scheduled testing session.
- The GTA will provide:
  - "The Internet," i.e., an network connection and an IP address for the outside world;
  - an Intel Wireless Gateway, with all gateway features turned off and WEP enabled; and
  - A client for testing.
- The TA will test the setup by becoming a customer of the network.

In the demonstration, a generic client should be able to connect to your internal network, get an IP address using DHCP, authenticate itself to the hot spot service running on the notebook, and then access services in the outside network (e.g., an external web server). Functionality is the priority. Each team should be prepared to demonstrate all working functionality, including any optional features.

Part IV – Deliverables, Report, and Grading Guidelines

Deliverables

You need to provide the following deliverables by the project due date.

1. The startup and shutdown shell scripts.
2. All script files, source files, and any other files that you have created to implement the authentication mechanism. All code (scripts and such) should be appropriately commented.

3. Ethereal or tcpdump traces that show the authentication process and the DHCP process. Perform the capture on the internal interface. The authentication capture should include the HTTP request, the authentication process, and the retransmission of the HTTP request and its fulfillment.
4. A project report, as described below. The report should be submitted as a Word or PDF file.

All deliverables should be submitted as a single .zip file named:

P11_*Partner1LastName_Partner2LastName*.zip

This file must be uploaded to Blackboard's Digital Dropbox by the due date and time.

Project Report

The project report should contain the following items in the order specified.

- A cover page specifying the course number and name; the project name (P11-Hot Spot Service); the name, student identification number, and email address of each team member; location (Blacksburg or NVC); and the date of submission.
- A listing and brief description of all files included in the deliverables.
- A brief discussion (of at most one page) of the firewalling approach and the firewalling rules. (A printout of the firewalling rules is not part of the one-page analysis.) Also, address any problems you faced in creating the firewalling portion of the project and how these problems were resolved.
- A printout of the firewalling rules should be included as an appendix. (iptables –t filter –L, iptables –t nat –L, iptables –t mangle –L)
- A brief discussion (of at most one page) of the routing scheme and related issues and printouts of the routing tables. (The printout of the routing table should be embedded in the project report. It is not part of the one-page discussion of the routing scheme.)
- A brief discussion (of at most one page) of the authentication mechanism. You should provide an overview of the code and/or scripts that you used to implement authentication.
- A brief discussion (of at most one page total) of the two packet traces described above.
- A list and brief discussion (with length not to exceed one page) of any optional features that you implemented.
- Answers to the following questions. Your answers should be concise, but complete.
    1. What are at least two possible security holes with this implementation?
    2. What are security measures against these, if any?
    3. What are at least two applications (or application layer protocols) that have issues with IP masquerading (these may helped by kernel modules)?
    4. What were the major problems encountered during this project?
    5. Were there any strange occurrences or other matters about which you wish to comment? (Provide details.)

Use fonts no smaller than 10 points and no larger than 12 points in your report. Use single-spaced paragraphs and a single-column format. Include all figures in the body of the report and label them clearly. Use headings to clearly identify the separate sections of your report.

Grading

Grading will be based on the following factors. The project will be graded based on a maximum score of 100 points.

- Correct operation as demonstrated to the GTA (70 points). Partial credit will be awarded for partial functionality. Be prepared to demonstrate all functionality that works, including optional features.
- Report content, including discussion of firewalling, routing, and authentication and answers to the questions (25 points).
- Overall quality of the submission, including writing of the report, comments in code, etc. (5 points).
- Implementation of optional features can compensate for up to 10 points, depending on the number and difficulty of optional features. The overall grade cannot exceed 100 points.

Appendix A – Useful Sources

This appendix provides links to information that may be helpful to you in completing this project.

Linux Networking

- Basic Networking knowledge and basic Linux implementations
  http://www.linuxdocs.org/HOWTOs/Net-HOWTO/index.html
- Advanced Routing Howto – if you want to know more about what is going on underneath
  http://www.linuxdocs.org/HOWTOs/Adv-Routing-HOWTO.html

DHCPd

- dhcpd.conf man page
- DHCPd mini-howto
  http://www.tldp.org/HOWTO/mini/DHCP/x369.html
- Linux Magazine dhcpd configuration help
  http://www.linux-mag.com/2000-04/networknirvana_01.html

iptables

- iptables man page
- iptables home page
  www.netfilter.org
- Iptables tutorial (very thorough)
  http://iptables-tutorial.frozentux.net/

Perl (version 5.6.1 comes with Red Hat 7.3)

- Perl books from O'Reilly
- Perl Homepage
  http://www.perl.com

CGI

- Use those Google searching skills! There are plenty of online references.
- Apache documentation on CGI
  http://httpd.apache.org/docs/howto/cgi.html
- Decoding forms with CGI, Links to CGI libraries for C++, Perl and Bash
  http://hoohoo.ncsa.uiuc.edu/cgi/forms.html

PHP (version 4.1.2 comes with Red Hat 7.3)

- PHP.net, the documentation section is excellent, including a basic tutorial for PHP programming and lots of example code
  http://www.php.net
- PHP Manual (see sections on "Getting started – Introduction," "Getting started – A Simple Tutorial," "Language Reference," "Variables – Predefined Variables - $_SERVER, $_POST, $_GET," and "Filesystem Functions")
  http://www.php.net/docs.php

Apache (version 1.3.23 comes with Red Hat 7.3)

- Apache configuration help (a minor modification is required)
  http://builder.cnet.com/webbuilding/pages/Servers/Apache/ss02.html

HTML

- Basics
  http://www.htmlprimer.com/lesson1.shtml
- Forms
  http://www.htmlprimer.com/forms.shtml

Appendix B – Food for Thought (if your mind is still hungry)

Think about these issues.

- Stateless and stateful features of the Linux IP masquerading implementation
- Other uses for network address translation
- Security applications of IP masquerading
- Security shortcomings of IP masquerading
- Concerns with IP masquerading for TCP, UDP, ICMP, and other layers above IP