Project 3 Grading Policy

November 24, 2001

1 Introduction

Your grade for project 3 will be computed based on following components. Weights of the components are as follows:

1. Correct Handin (15%)
2. Firewall (30%)
3. Dynamic NAT support (17%)
4. Back channel support for FTP (18%)
5. Testing methodology (20%)

We now describe the evaluation of the components in more detail.

2 Correct Handin

This component consists of the following two parts.

- **Write-up (60%):** You need to submit a write-up file named *proj3_report.{ps|pdf|txt|html}* that includes all the sections (except the testing strategies) mentioned in the project handout. Each missing section will result in a penalty of 20%.
- **Handin (40%):** You need to handin your project in the way similar to project 2. Make sure to follow the instructions below.
  - Do not hand in any executables or object files.
  - Hand in only one version of your code.
  - When executing gmake in your handin directory, the following executables (and the corresponding object files) must be generated: kernel, setfilter, setnat.
  - The compilation/building process must not result in any warnings (and, of course, errors).
  - The compilation/building process may build binaries that you used for debugging, verification etc. Such additional program (both code and executables) must be preceeded by the prefix: *test_* (*e.g. test_firewall*). See the *testing methodology* section for more information.
  - Your makefile must use gcc (not cc)

3 Firewall

This component consists of three parts (functionality, robustness, and code quality), the first one having a weight of 60% and the second and third one each having a weight of 20%.

- *Functionality:* Here is an *incomplete* list of tests to check correct functionality of your firewall.
  - Does it use both the session and policy tables?
  - Does it accept a SYN packet for a connection that is in `ESTABLISHED` state now?
  - Does it accept a packet with wrong sequence number for an existing connection?
  - Does it accept UDP/ICMP packets not allowed by the policy table?

  Note that, this is an incomplete list of tests. We will come up with many other different test cases, so be creative and come up with you own test cases. However, we will not penalize you much for the errors in your project 2 code (stateless firewall). We will mainly focus on revealing the strength (and weakness) of your code as a stateful firewall.

- *Robustness:* As in project 2, your source code is visually inspected for robustness issues.
- *Code quality:* Your code will be graded based on how well-commented and readable it is.

4 Dynamic NAT support

As before, the grading will be based on functionality(60%), robustness(20%) and readability(20%). As functionality, your code has to do NAT for a forwarded outgoing, forwarded incoming, and locally generated outgoing packet. Each correct case contributes 33% (with an additional 1% if all three cases are correct).

5 Back channel support for FTP

Again, as before, the grading will be based on functionality(60%), robustness(20%) and readability(20%). The tools (ftpserver/ftpclient) to test this feature of your firewall has been provided in the project directory.

6 Testing methodology

You need to come up with your own testing methodology to be sure that your code (mainly for the statefull firewall) actually works. For this part, you need to submit the following.

- A report named *proj3_test.{ps|pdf|html|txt}* that describes how you tested your implementation of statefull firewall (60%), dynamic NAT (20%) and backchannel support for FTP (20%). For the firewall part, state how you generated invalid TCP packets with wrong sequence number or flags.
- If you use some additional tools to test the code, you can submit these. But, as mentioned before, the code and executables for these testing purpose should have the prefix *test_* (*e.g.* *test_firewall*). You can also embed some testing code within your actual code, suronded by some `#define` directive. If you do so, mention it in the report.