Practical Assignment: Network Monitoring and Troubleshooting Assignment

Submission: by 8pm, Sunday, 13 April 2003 (Note the change).

Where: Online at `http://ictlab.tyict.vtc.edu.hk/perl2/submit.cgi`. A paper submission is not required (save the trees!)

Put your *email address* in a comment right near the top of your assignment so that I can return the marked assignment to you. Assignments with no email address will not be returned.

Cheating: Your work *must* be original. Copying will be *severely* dealt with. I will use software tools to help identify copying. I will Google for phrases from your submission. There will be *no mercy for cheats*.

1 The Requirements

Select Topic at next class before April 2 2003 You must choose a topic for your assignment and inform me in writing of your choice during your next laboratory class before April 2 2003.

Submit the following:

- A report (can be in plain text, Open Office XML format, Microsoft Word format, or LaTeX) which explains the investigation and its conclusions, having the following sections:
  - Introduction
  - Aims
  - Methodology and tools used, setup, configuration, etc. This should be the "meat" of your report.
  - Data and analysis (where appropriate)
  - Conclusion (important!)
- Text files that include the setup, configuration, any software or scripts written, etc.
- Screen shots (.png format) showing interesting data you collected

Topic: You are required to perform some investigation that relates to topics covered in either or both of the lectures and workshops on *SNMP* or *Network Troubleshooting*. You may, for example, investigate the network in the Institute, at your home or of the Internet. This investigation should show originality, and be different from others. If you study with other students, all members of the study group should submit work that is quite different and that investigates different areas.

Tools: Use any of the tools we have discussed in the topics of *SNMP* and *Network Troubleshooting*, including Net-SNMP, Cricket, MRTG, Ethereal, Ntop, `tcpdump` or `windump`, `pathchar`, `ping`, `dig`, `nslookup`, `telnet`, and all the others I have not listed here. You may use other tools that we did not cover in Lectures or the workshops.

Note: if you are using `tcpdump`, I suggest that you use the current version (3.7.1) and the corresponding `libpcap` package (0.7.1). The reason is that the newer `tcpdump` gives much more detail of many more protocols, including DHCP as well as many others. The previous release of `tcpdump` was quite ancient.

I have made RPMs of these; you can download them from `http://ictlab.tyict.vtc.edu.hk/ftp/redhat/contrib/tcpdump-3.7.1-1nu.i386.rpm` and `http://ictlab.tyict.vtc.edu.hk/ftp/redhat/contrib/libpcap-0.7.1-1nu.i386.rpm`, or within the college, by NFS from your NFS directory at `/home/nfs/redhat/contrib/`.

I have written a handout that describes the `tcpdump` output with DHCP packets. Please refer to that for more information about using `tcpdump` with DHCP.

2 Suggestions

If you have some ideas of your own, great—go ahead and implement them. Here are some other ideas that may stimulate you:

1. Investigate portscans on your firewall, and graph the results using Cricket. See chapter 10 of the *Linux Network Administrator's Guide* at `http://tldp.org/LDP/nag2/x-087-2-accounting.html`. Also, read about the `EXEC` value to `US-source` in the Cricket Reference Manual (`http://cricket.sourceforge.net/support/doc/reference.html`). Search for examples in the sample configurations provided with Cricket:
   `$ find ~/cricket/sample-config -type f | xargs grep -i 'ds-source.*exec'`

2. See the *Cricket Contributed Software Repository* at `http://www.certaintysolutions.com/tech-advice/cricket-contrib/`, and investigate the use of an interesting tool.

3. Make a tool to monitor the network for rogue DHCP servers using `tcpdump` and Perl or the shell. Provide a configuration file to hold the name of the legitimate DHCP servers, and the email addresses that it should send email to.

   Like many other projects, this could start out really simple (listen with a filter than monitors UDP packets coming from port 67 from IP addresses that do not match our DHCP server), and grow into something really sophisticated and useful: runs as a daemon (i.e., you can log out and it can still continue to run—see the next suggestion), can send SMS via email, very configurable (can configure to only send when get bad behaviour, i.e., the rogue server sends `DHCPNAK` to legitimate `DHCPREQUEST`s, or sends `DHCPOFFER`s indiscriminately that do not provide correct parameters for our network.)

4. Write a tool to analyse system logs for particular events, e.g., monitor `/var/log/httpd/access_log` for Nimbda Worm scans, and email the system administrator with the IP addresses of machines in the local network that are infected. The important thing is that it should cope with *log rotation*. Every week, each log file gets renamed, and a new log file is created. Your tool needs to detect this, and reopen the new file. I suggest the

use of the Perl module File::Tail (or the alternatives) from CPAN, since they handle this problem automatically.

Investigate making it a system daemon, so that it can continue to run after you log out. See page 634, recipe 17.45, "*Making a Daemon Server*" of *Perl Cookbook*, Tom Christiansen and Nathan Torkington, O'Reilly, 1998, or just see `perldoc -q daemon`.

5. Use one of the available tools for sending SMS messages via email, and adapt it for use with Cricket to send alerts.

6. Investigate using OpenNMS to monitor our network

7. Investigate network traffic on a switch in our Department and perform meaningful analysis of laboratory traffic, using Ntop or Cricket (or both)

8. Install Cricket on Windows using the current version of the software, and write a short manual explaining accurately how to do it.

9. Write a program to take `tcpdump` text input, and output all DHCP messages that belong to each session (i.e., with common `xid`, or transaction ID). All messages in one session are output together. Provide options to select various types of DHCP exchanges, such as all sessions that contain both `DHCPACK` and `DHCPNAK`, or all sessions that contain a `DHCPNAK`.

10. In the same directory as this document, you will find a large `tcpdump` binary file containing 673596 DHCP packets. You could write some software to analyse this (or draw graphs of frequency of requests, or of types of requests). Note that the file is large (271 MiB), and you possibly need to cut it into smaller pieces using a tool like `tcpslice`, that comes with `tcpdump`.

Marking Scheme There will be four grades for this assignment: A, B, C and F.

Grade A The student has done some original work which clearly demonstrates the student's competence and familiarity with the topic of investigation. The work involves some new application of the material presented in the teaching notes, not just a direct implementation of the laboratory material or the lecture notes. The investigation is interesting, demonstrates something that has not been discussed directly in classes. The investigation is thoughtfully presented and logically organised, and demonstrates a coherent, methodical approach. The student can demonstrate their work and discuss their findings.

Grade B The investigation uses concepts from the workshop notes and lectures and integrates them in a meaningful way. There may not be a particularly interesting conclusion, but the student has demonstrated competence and has definitely gone beyond simply repeating the laboratory exercises in a slightly different context. The student can demonstrate their work and convince me that it is their own.

Grade C The investigation shows little more than a direct implementation of the laboratory notes, and contains nothing much that is new or interesting. However, the student has demonstrated a basic competence and is able to demonstrate in practice that they clearly understand what they are doing.

Grade F The student has copied other students' work, or has provided something that they do not understand, that lacks logical consistency, that they are unable to discuss or demonstrate convincingly. The work has the appearance of being done in a mad rush in the hour just before the deadline.