Homework 2
CS 642: Information Security

March 3, 2014

This homework assignment covers topics in network security. You may *not* work with a partner. It is due **March 24, 2014** by midnight local time. The deliverable should be a nicely formated hw2.txt or hw2.pdf file, with your ID at the top, as well as the source code for your solution to problem 2 below. Turn in will be via Moodle (Heming will send out further instructions via email to the class list). Each part of each problem is worth 2 points, so you have the opportunity to get partial credit for each. Make answers concise (you will not get credit for rambling and long answer that happens to contain some correct portions).

1 Problem 1

On the web site is a zip file containing packet traces that can be read by the tool WireShark (among others). For this problem there are 4 traces that you will need to investigate in order to find out the information asked for below. To get started, you will want to understand how to apply WireShark's filtering feature.

(Problem and part of dataset originally from Stanford CS 155 course taught by Dan Boneh. Thanks!)

Trace 1: HTTP traffic

1. Give three websites visited from source IP address "192.168.0.100".
2. Give three search queries made from source IP address "192.168.0.100".

Trace 2: FTP traffic

FTP is the file transport protocol. There is a lot of information about it on the Internet.

1. What is the user name and password used to connect to the FTP server?
2. Explain the difference between a passive FTP connection and an active FTP connection.
3. Give the packet number ranges across which there were active connection(s).
4. Give the packet number ranges across which there were passive connection(s).
5. List any files that were downloaded.

Trace 3: Traceroute

Traceroute is a tool used to determine the route between two IP addresses.

1. Identify the source IP address that issued a traceroute command.
2. Identify the destination IP address of the traceroute command.
3. List the IP addresses on the route between the source and destination.
4. Determine the organizations associated with all the IP addresses along the route.

Trace 4: POP

The POP protocol is used for Email.

1. What is the POP username and password?
2. How many emails are in the user's mailbox?
3. Give the contents of from, to, subject, and date for each email.

2 Problem 2

Recall that a SYN DoS attack has an attacker send a large number of SYN packets (TCP/IP packet with SYN flag set) to a victim with a spoofed source IP address. Suppose a well-intentioned, but not-so-clever network engineer decided to setup a system with a custom IDS program that sniffs traffic using the pcap library and logs all the TCP SYN packets to a file for later inspection. A testing version of the C code implementing the scanner is given in the file scanner.c (on the class web page). It reads packets from a pcap file in offline mode, but the idea will be to replace `pcap_open_offline()` with the logic needed to use `pcap_open_live()`. The engineer has asked you for your opinion about his proposal.

1. List as many ways an attacker can abuse such a setup as you can think of.
2. For each issue, explain how you would address it.

3 Problem 3

In this problem, you will write a simple intrusion detection system to detect potential attacks or dangerous behavior in network activity. On the web page are three pcaps with example attacks:

1. arpspoofing.pcap includes an ARP spoof attack. IP address 192.168.0.100 advertises the wrong MAC address for 192.168.0.1.
2. portscan.pcap includes a TCP SYN port scan.
3. tcpflood.pcap includes a TCP SYN flood.

Your job is to write a software IDS executable (in C/C++) or script (in Python) that takes as input a pcap trace and looks for such malicious behavior. The local network you are protecting is configured with two machines (192.168.0.100 with MAC address 7c:d1:c3:94:9e:b8 and 192.168.0.103 with MAC address d8:96:95:01:a5:c9) and a router (192.168.0.1 with MAC address f8:1a:67:cd:57:6e). Your scanner should:

1. (2 points) Detect ARP spoofing attempts. Output a warning including the offending MAC address and the packet number of the offending packet.

2. (2 points) Detect port scans. A port scan is defined to occur whenever TCP SYNs or UDP packets are sent to a 100 or more different ports on a target system[1]. The scanner should output a warning including the offending source IP address, the victim destination IP address, and the offending packet numbers.

3. (2 points) Detect TCP SYN floods. Your tool should detect when the number of TCP SYNs to a particular destination (that are not associated with completed handshakes) exceeds 100 per second. The scanner should output a warning including the offending source IP address, victim destination IP address, and the offending packet numbers.

Your program should take as input the filename of a pcap file that contains captured network packets. The output of your program will be the warning messages as described above. The format of your result is free but it should be clear to the user.

You should use either C/C++ or Python. Both have libraries for reading, parsing, and analyzing pcap files. The TA will give some hints about getting started using these libraries. If you use C/C++ include in your turned in files a Makefile that properly compiles it. Check that your scanner runs properly on the mumble machines before turning it in.

The sample pcap files can be used to test your scanner. We will also test your scanner on fresh pcaps we generate that include other non-malicious behaviors, as well as boundary conditions (e.g., a TCP SYN flood that does not exceed 100 packets per second).

4 Problem 4

Suppose we are in a setting where a small business (on the order of several 10's of hosts) runs its own DNS server. The business hired you as a security consultant.

1. Describe how DNS cache poisoning can be mounted by a corporate competitor. Explain any assumptions about the DNS server's implementation that your attack requires. Explain what a successful DNS cache poisoning attack would enable the attacker to do.

2. In class we discussed how one can use side channels to sneakily perform port scans. The side channels in this case were the TCP/IP stacks of an idle (zombie) host. These used the ability to directly infer the state of the TCP/IP stack (e.g., by seeing the IP ID number or lack of a FIN packet). Another type of side channel is a so-called timing channel, where one uses timing of an operation to infer information about the state of some system.

   You discover that the DNS server accepts address queries ("A" queries) from arbitrary IP addresses and that it does perform caching of resolved queries. Say a competitor might be interested in seeing the how often employees at the business visit a website "www.buysomeparts.com". The authoritative name server for buysomeparts.com sets the TTL to 15 minutes, so that it can do load balance traffic across time amongst a set of servers. Describe how an attacker can use the DNS cache as a side channel in order to infer how often employees visit "www.buysomeparts.com". Discuss how the TTL impacts the effectiveness of the attack.

[1]Note the constant 100 used above in the examples is somewhat arbitrary, and in a real deployment this would need to be fine-tuned.