CS 6262: Network Security January 16, 2009

Problem Set 1: Passwords and Packets

Instructor: Prof. Nick Feamster *College of Computing, Georgia Tech*

This problem set has three questions, each with several parts. Answer them as clearly and concisely as possible. You may discuss ideas with others in the class, but your solutions and presentation must be your own. Do not look at anyone else's solutions or copy them from anywhere. (Please refer to the Georgia Tech honor code, posted on the course Web site).

Turn in your solutions in on **January 31, 2009** by 11:59 p.m. -0500, to the grader.

1 Warm Up

1. Examine the source code or man page for `crypt`. How does this program take a plaintext password and generate the ciphertext that we see in `/etc/password`, or `/etc/shadow`? What cipher is used to generate the cipher from the plaintext?
2. Using `crypt(3)` on a UNIX machine, generate the ciphertext for `security` and `netsecurity`. What do you observe? Why?
3. `passwd` typically uses something called a "salt" to generate the password for each user. Why?

2 Password Cracking

For this problem, you will need to install John the Ripper: `http://www.openwall.com/john/`.

Consider the following password file generated with `crypt(3)`:

```
root:IWpIzqD0jR1.c:100:100:Charlie Root:/home/root:/bin/sh
cs6262:UNzrFi5aYL9DU:101:101:CS6262:/home/cs6262:/bin/sh
mysql:WqCBVG36lcuAc:102:102:MySQL:/home/mysql:/bin/sh
guest:FTQinpjr.VRM.:103:103:Guest:/home/guest:/bin/sh
test:LF2c9qM5l6X7Q:104:104:Testing:/home/test:/bin/sh
```

You can obtain this password file from
`http://www.gtnoise.net/classes/cs6262/spring_2009/psets/ps1/6262.pw`.

1. Run the default mode of John; one of these passwords will be cracked. Which one? Why (which rule of John was applied)?
2. Now, try the "wordlist" mode of John. Which password is cracked now? Which rule of John was applied? (*Hint:* John's default wordlist is quite small by default. You may have to augment this wordlist with one of your own.

3. One of the users has a pw that is a rotated version of a dictionary word. Modify John's rule list to incorporate this feature. Which password does this now reveal? Please include the source for your modifications to the rule list.

4. One user is predisposed to using leetspeak (4 for a/A, 1 for i/I and l, 3 for e/E). His password is also a dictionary word. Modify john.conf to incorporate this feature. Which password is revealed?

5. One user likes to swap two adjacent characters of a dictionary word. Can you modify john.conf to do this using existing syntax? If not, how can you incorporate this feature? What is the password that is revealed?

3 TCP Stream Reassembly

1. Write a program to reassemble an HTTP session. Your program should display for each HTTP session the request and response between the client and the server. It should also output the reassembled HTML response as a file. A test trace is available here: `http://www.gtnoise.net/classes/cs6262/spring_2009/psets/ps1/6262.pcap.bz2`. *We will grade with a different trace.*

   Your program should produce the reassembled HTML: the TCP payload data, without the headers. Your reassembled stream should display properly when opened in a browser like Firefox. Here are the steps we will use for grading:

   - We will run the code you turn in against our test trace.
   - We will attempt to open the reassembled stream in Firefox.

   Hint: you can get the source code of the tcpdump and modify it to record the payload of TCP packets of HTTP sessions.

2. Suppose that a TCP stream reassembler has been installed as part of a firewall, n hops upstream from a server that it is trying to pretect. Suppose also that your TCP stream reassembler can reassemble a telnet stream and is looking for the password "`password`" in an incoming TCP stream.

   (a) Explain how you might construct a TCP stream that transmits the entire stream `password` to the server but prevents the reassembler from raising an alarm. Note that an improved TCP stream reassembler would be able to reassemble strings across packets. (*Hint:* Think about how you might get a packet to reach the reassembler but not the server.)

   (b) **Extra credit.** Write a program that implements your exploit.

Please turn in your code to this assignment to the grader, Priyank Raj, by email. Your code should include a README describing everything we need to do to compile and run your code.