NIS Lab Report

Lerato Mosegedi
Banele Matsebula
Thuto Aphiri

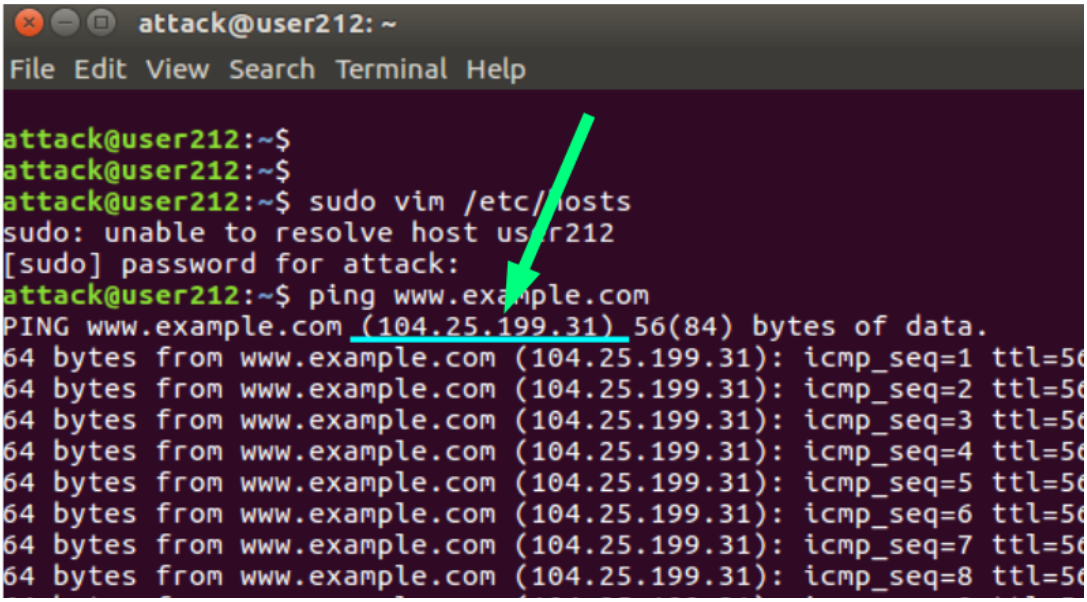
Overview

The lab is about DNS Pharming attacks, which is essentially manipulating the resolution process of the DNS servers in various ways, with an intent to misdirect users to alternative destinations, which are often malicious. The objective of this lab is to understand how such attacks work.

The lab is set up in such a way that the user's computer, DNS server, and attacker's computer are on one physical machine, but using different virtual machine.

Task 1: Attackers have already compromised the victim's machine

This task assumes that we, the attackers, have access or have already comprised the user's computer. With the latter, we manipulated the resolution process of the DNS server by changing the IP address of the host name which the user wanted access to, to where we want to redirect the user to in (/etc/hosts). For our attack we changed the IP address of www.example.com to 104.25.199.31, and as shown below when using the ping command for www.example.com it shows our IP address.



Task 2: Directly Spoof Response to User

In this task it is assumed that we do not have access to the user's computer, therefore to manipulate the user's resolution process of the DNS server we listen to the user's DNS request to the DNS server. After hearing the DNS request we spoof a fake DNS response which met the requirements of an acceptable response by the user's machine. We used Netwox listen to the packets and respond to google.com when requested, by running the command on figure 2 .

We found out that listening to the user's request when they used www.example.com and responding faster then the DNS server was impossible, because the DNS server is configured to find www.example.com and is in the same LAN as the user's machine therefore does not have to go outside the LAN to find the IP address. So, we used google.com to be able to respond before the DNS server did. As shown on the figure e below, the attackers machine which is running netwox can listenen to the user's DNS requests labelled "**DNS question"** and the DNS reponses which are labelled "**DNS answer**". Both DNS answers are our spoofed responses to misdirect the user. Figure 3 shows the google.com spoofed response, as its IP address is the one we set

```
attack@attacker212:~$ sudo netwox 105 -h "google.com" -H "1.2.3.43" -a "ns.example.com" -A "192.168.
212.12" -f "src host 192.168.212.100"
```

Figure 1. Command for Netwox

```
DNS_question
  id=61616  rcode=OK            opcode=QUERY
  aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=0
  search.apps.ubuntu.com. A

DNS_answer
  id=61616  rcode=OK            opcode=QUERY
  aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1
  search.apps.ubuntu.com. A
  search.apps.ubuntu.com. A 10 1.2.3.43
  ns.example.com. NS 10 ns.example.com.
  ns.example.com. A 10 192.168.212.12

DNS_question
  id=2669   rcode=OK            opcode=QUERY
  aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=0
  google.com. A

DNS_answer
  id=2669   rcode=OK            opcode=QUERY
  aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1
  google.com. A
  google.com. A 10 1.2.3.43
  ns.example.com. NS 10 ns.example.com.
  ns.example.com. A 10 192.168.212.12
```

Figure 2. Attacker listening and spoofing user's machine

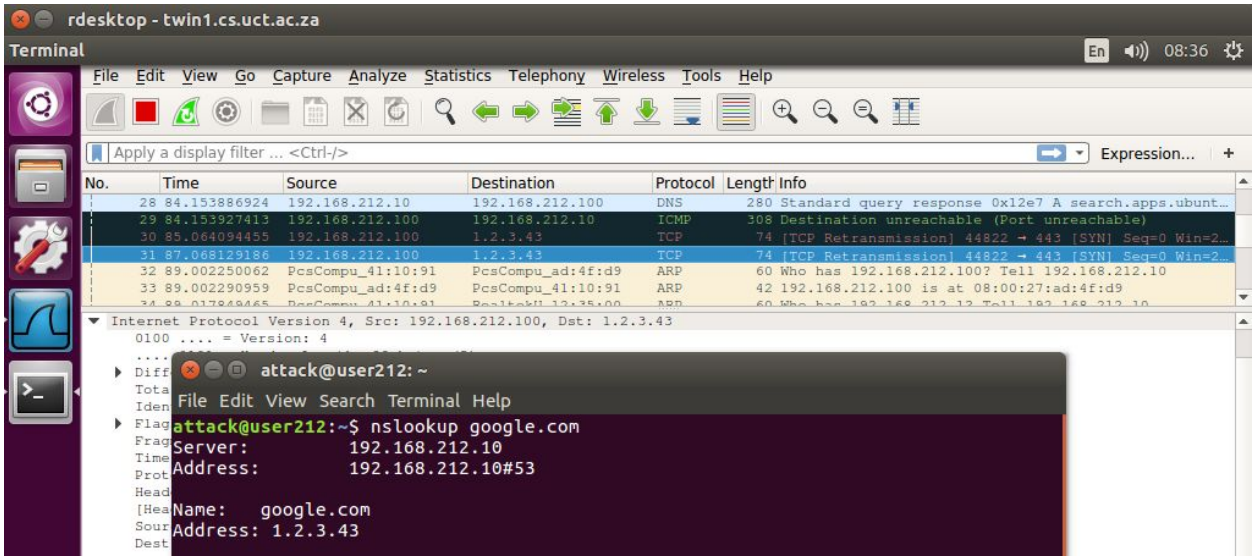


Figure 3. Spoofed response reached user's machine, and was accepted

Task 3: DNS Server Cache Poisoning

The previous attack targets the user's machine, which is does not have a long lasting effect and very inefficient because the attacker's machine must send out a spoofed DNS response

everytime the user the user send a DNS request. A more effecient way to do this is to target the DNS server. When the DNS server can't resolve the domain name, it goes out to ask other DNS sever to resolve the domain. To attack the DNS server we first set up Networx with the command in figure 4. to listen to the DNS server and when the it goes out to resolve www.external.com the attacker should spoof the the DNS server and give it 1.2.3.4 as the IP address to save in its cache and set the time period its going to be there. Initially the cache was empty, after tthe spoofing it was in the cache as indicated by figure 5 and 6.

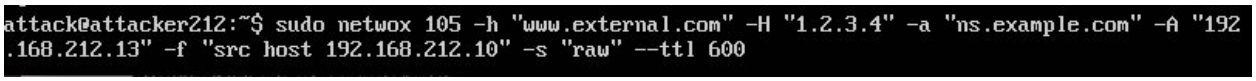


Figure 4. Attacker listening and spoofing DNS server

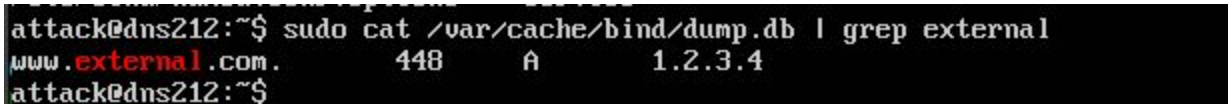


Figure 5. Proof of spoofing. www.external.com is in the cache

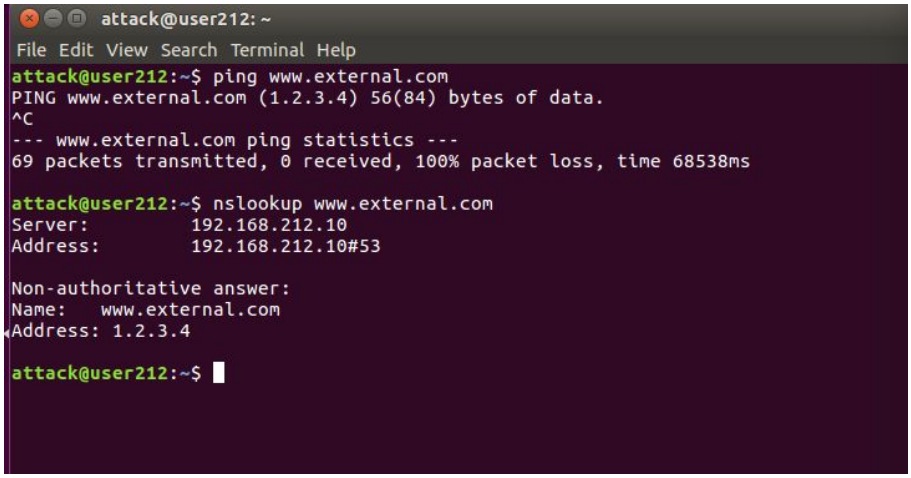


Fig 6. Proof of spoofing