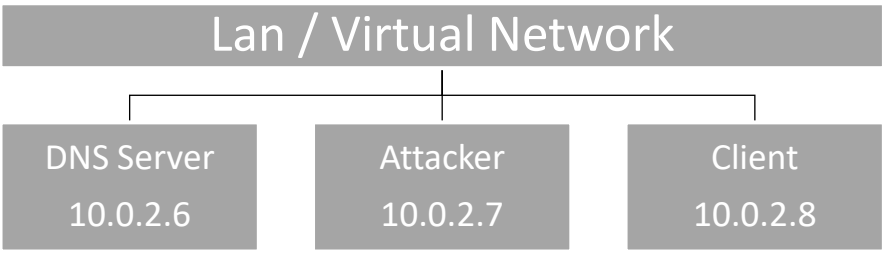
Lab - DNS Pharming

Network Security

DNS or Domain Name System is the Internet's phone book; it translates hostnames to IP addresses (or IP addresses to hostnames). This translation is through DNS resolution, which happens behind the scene. DNS Pharming attacks manipulate this resolution process 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.

Environment Setup

We will be using 3 Virtual Machines, a DNS Server, an Attacker and a Client. These machines are pre-configured as shown below. The Virtual Network is a VMware Internal network that connects the virtual machines together. The website to be used is: www.netsec-week3.com



The tools being used in this lab are Wireshark, Netwox/Netwag and Bind9.

Wireshark

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course). Read More at www.wireshark.org.

Netwag/Netwox

Netwag is a graphical front end for netwox. Netwox is a toolbox for testing an Ethernet/IP network which includes a network library for administrators and hackers. Its objective is to let programmers easily create network programs. This library provides features for Ethernet, IP, UDP, TCP, ICMP, ARP, and RARP protocols. It supports spoofing, sniffing, client, and server creation.

Bind9

It is open source software that implements the Domain Name System (DNS) protocols for the Internet. It is a reference implementation of those protocols, but it is also production-grade software, suitable for use in high-volume and high-reliability applications. The name BIND stands for "Berkeley Internet Name Domain", because the software originated in the early 1980s at the University of California at Berkeley.

Lab Objective

The main objective of Pharming attacks on a user is to redirect the user to another machine B when the user tries to get to machine A using A's host name. For example, when the user tries to access the online banking, such as www. commbank.com.au, if the adversaries can redirect the user to a malicious web site that looks very much like the main web site of www. commbank.com.au, the user might be fooled and give away password of his/her online banking account.

When a user types in www. commbank.com.au in his browsers, the user's machine will issue a DNS query to find out the IP address of this web site. Attackers' goal is to fool the user's machine with a faked DNS reply, which resolves www. commbank.com.au to a malicious IP address. There are several ways to achieve such an attack. In the rest of the lab description, we will use www.netsec-week3.com as the web site that the user wants to access.

To check if the lab environment is working properly, we will initiate a DNS lookup from the client and attacker VM. To do this simply open the terminal and use the following command:

- `dig www.netsec-week3.com`

If the response received is similar to the one below, the network configuration is working normally.

** Important** Take a screenshot of the results as it will be used for comparison later.

```
<<>> DiG 9.9.5-3ubuntu0.8-Ubuntu <<>> www.netsec-week3.com
global options: +cmd
Got answer:
->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29823
flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2
OPT PSEUDOSECTION:
EDNS: version: 0, flags:; udp: 4096
QUESTION SECTION:
www.netsec-week3.com.       IN    A
ANSWER SECTION:
www.netsec-week3.com. 259200     IN    A     10.0.2.101
AUTHORITY SECTION:
netsec-week3.com.     259200     IN    NS    ns.netsec-week3.com.
ADDITIONAL SECTION:
ns.netsec-week3.com.  259200     IN    A     10.0.2.10
Query time: 1 msec
SERVER: 10.0.2.6#53(10.0.2.6)
WHEN: Mon Aug 22 23:20:05 PDT 2016
MSG SIZE  rcvd: 98
```

Task 1: Attack by modifying HOSTS file

The host name and IP address pairs in the HOSTS file (located at /etc/hosts) are used for local lookup and they take the preference over remote DNS lookups. If an attacker is able to access this file and modify the local file, the system will ignore the lookup data received from the DNS server. As we are using a virtual machine and the network configuration is manually defined, the same results can be obtained by **modifying /etc/network/interfaces**. Use The terminal on Netsec-Client and change the DNS as below.

- `dns-nameservers 10.0.2.7`

The IP address provided here is of the attacker machine. Once the changes have been made, reboot the system for them to take effect. Once rebooted, run the `dig` command and compare the results with the original results received earlier. Take a screenshot and note the difference in your observations document.

Hint: Use the following command to edit the file. You can use the arrow keys to navigate in the editor.

- `sudo nano <file name>`

Once done editing, press `'Ctrl X'` then `'Y'` and finally press `'Enter'` to save the changes.

Task 2: Attack by spoofing DNS response

In this attack, the victim's machine has not been compromised, so attackers cannot directly change the DNS query process on the victim's machine. However, if attackers are on the same local area network as the victim, they can still achieve a great damage. When a user types the name of a web site in a web browser, the user's computer will issue a DNS request to the DNS server to resolve the IP address of the host name. After hearing this DNS request, the attackers can spoof a fake DNS response. The fake DNS response will be accepted by the user's computer if it meets the following criteria:

1. The source IP address must match the IP address of the DNS server.
2. The destination IP address must match the IP address of the user's machine.
3. The source port number (UDP port) must match the port number that the DNS request was sent to (usually port 53).
4. The destination port number must match the port number that the DNS request was sent from.
5. The UDP checksum must be correctly calculated.
6. The transaction ID must match the transaction ID in the DNS request.
7. The domain name in the question section of the reply must match the domain name in the question section of the request.
8. The domain name in the answer section must match the domain name in the question section of the DNS request.
9. The User's computer must receive the attacker's DNS reply before it receives the legitimate DNS response.

To satisfy the criteria 1 to 8, the attackers can sniff the DNS request message sent by the victim. They can then create a fake DNS response, and send back to the victim, before the real DNS server does. You must use the Netwox/Netwag tool on the attacker VM to conduct such sniffing and responding. Remember to take screenshots for your observations document. Use Wireshark to capture the DNS response.

Hint: Use Netwox/Netwag tool 105 along with the appropriate parameters:

- `sudo netwag` **`or`** `sudo netwox`

Task 3: DNS Server Cache Poisoning

The above attack targets the user's machine. In order to achieve long-lasting effect, every time the user's machine sends out a DNS query for www.netsec-week3.com, the attacker's machine must send out a spoofed DNS response. This might not be so efficient; there is a much better way to conduct attacks by targeting the DNS server, instead of the user's machine.

When a DNS server receives a query, if the host name is not within its domain, it will ask other DNS servers to get the host name resolved. Note that in our lab setup, the domain of our DNS server is netsec-week3.com. Therefore, for the DNS queries of other domains (e.g. www.google.com), the DNS server will ask other DNS servers. However, before it asks other DNS servers, it first looks for the answer from its own cache. If the answer is there, the DNS server Apollo will simply reply with the information from its cache. If the answer is not in the cache, the DNS server will try to get the answer from other DNS servers. When it gets the answer, it will store the answer in the cache, so next time, there is no need to ask other DNS servers.

Therefore, if attackers can spoof the response from other DNS servers, the DNS server will keep the spoofed response in its cache. Next time, when a user's machine wants to resolve the same host name, it will use the spoofed response to reply. This way, attackers only need to spoof once, and the impact will last until the cached information expires. This attack is called DNS cache poisoning.

We will be using the same procedure as in Task 2, however make sure before attacking that the DNS Server's cache is empty. You can flush the cache using the following command:

- `sudo rndc flush`

Use the details below to run this task for `www.uts.edu.au`

- `Hast name:` `www.uts.edu.au`
- `Host name IP:` `10.0.2.7`
- `Name server:` `ns.uts.edu.au`
- `Authns IP:` `10.0.2.10`

You can tell whether the DNS server is poisoned or not by using the network traffic captured by Wireshark.