Systems Security lab 3: Networks

DANGER

This should be obvious, but ...

ONLY do network attacks on a machine that is not connected to the internet, unless instructed otherwise.

DO NOT attack the CS/eduroam networks by accident – think before you shoot!

Lab setup

You will need 3 VMs: 1. attacker 2. client 3. server. I recommend the following setup.

1. Unzip 3 copies of the SEEDUbuntu12.04.zip into different folders called seed1/, seed2/ and seed3/.
2. Go to the folder in seed2/ that contains the vmdk files and run the following command:

```
vboxmanage internalcommands sethduuid SEEDUbuntu12.04.vmdk
```

This changes the HD's GUID, otherwise virtualbox gets confused if there are several disks in different VMs with the same GUID.

3. Repeat step 2 for seed3/.
4. In the virtualbox main window, choose File/Preferences, Network. On the "NAT networks" tab, create a new network with the "add" button on the right.
5. Create 3 VMs in virtualbox for the 3 disk images that you unzipped in step 1. Give each one 512MB RAM – not more unless you have a really powerful computer. Remember that each VM must use a different disk image.
6. For each of your 3 VMs, right-click in the main window and choose Settings, Network. Change "Adapter 1"s "Attached to" type from "NAT" to "NAT network".
7. Start all 3 VMs and run **`ifconfig`** in each of them, noting their IP (v4) and MAC addresses.
8. To stop ubuntu's "phone home" features from creating unnecessary noise when you're watching the network, run the following command in each of your 3 VMs (type it all on one line).

```
sudo apt-get remove unity-lens-video unity-scope-video-remote whoopsie
unity-lens-applications geoclue-ubuntu-geoip
```

We now have a network in which all 3 VMs are connected to a gateway (emulated by virtualbox).

Netwox

The lab vm comes with a tool called netwox preinstalled which allows you to easily run and script network attacks. The general syntax is

```
# netwox [attack-number] [parameters ...]
```

which typically needs to be run as root to have raw network access. You can also do

```
$ netwox [attack-number] --help
```

to display the syntax of a particular attack.

For every attack, netwox will have a plugin to do exactly what you need, see `http://tinyurl.com/netwox-options` for a list of all options.

The challenge in this lab is not usually to get the attacks to work – rather, you should

- Describe what the relevant protocol is supposed to do and how it is supposed to work.
- Describe exactly what is going on in the attack – with evidence (from wireshark or other monitoring tools).
- Why the attack is having the effects that you observe.
- Then, think about how you could protect against the attack in question.

Promiscuous mode

Some attacks need the attacker VM1 to run its network card in promiscuous mode. With VM1 turned off, right-click the machine in virtualbox and choose Settings, Network, Advanced. Set "promiscuous mode" to "allow all". Then start VM1.

On the running Vm1, type "**`ifconfig`**" to get the network card name (e.g. eth12) and then type the following command. Explain in your report what it means.

```
sudo ifconfig eth12 promisc
```

Task 1 – ARP spoofing

Enable telnet on VM3 with the command

```
sudo service openbsd-inetd start
```

The scenario is that VM2 is trying to telnet to VM3 and VM1 is attacking this connection. On each VM in turn, run ifconfig and note their IP addresses.

On VM2, run **arp -n** to display the ARP cache. Then ping VM1 and VM3 (Control-C to cancel):

```
ping [IP address of VM1]
ping [IP address of VM3]
```

Rerun **arp -n** and observe the state of the cache now. You should see the IP:MAC address mappings.

Try **telnet [IP of VM3]** to connect – you should be asked for your username/password. When you get a remote shell, type Control-D to exit.

Similarly, we can do HTTP over telnet: type the following (still on VM2, and you may need to be quick to avoid a timeout):

```
telnet [IP of VM3] 80
GET / HTTP/1.1
Host: vm2
```

Press ENTER twice at the end to end the headers and send the request. You should get a page with "It works!" back.

Next, we'll observe an ARP request.

- On VM2, type **sudo arp -d [IP of VM3]** to clear the cache entry.
- On VM1, type **sudo netwox 7**.
- Telnet from VM2 to VM3. You should see the ARP request in VM1's terminal – but not the response. Why is this?

Now for the actual attack.

Make sure you are NOT connected to a wireless network.

Run the ARP spoofing attack on VM1, targeting the path from VM2 to VM3.

If successful, VM2 should be unable to connect to VM3 at all. (If it just delays the connection for a few seconds, you need to improve the attack.)

You can watch what is going on with **sudo netwox 7** in a terminal on VM1 and/or **arp -n** on a terminal in VM2.

Task 2 – SYN flood

Turn your internet connection off again.

On VM3, type the following to disable some built-in protection schemes:

```
sudo sysctl -w net.ipv4.tcp_max_syn_backlog=64
sudo sysctl -w net.ipv4.conf.all.rp_filter=0
sudo sysctl -w net.ipv4.tcp_syncookies=0
```

Check that VM2 can telnet to VM3, then close the connection again.

Check again that you're not connected to the internet/wireless network.

Launch your attack on VM1 against port 23 (telnet) of VM3.

If successful, VM2 should be unable to telnet to VM3.

- Run the attack, document and explain what is going on. The command **`netstat -tan`** on VM3 and/or wireshark can help you here.
- Try the attack with syncookies turned on again (set the option to 1). What happens now?
- Explain how syncookies work and what the other two protection options do.

Task 3 – TCP reset

Disconnect your machine from the internet.

- Open a telnet session from VM2 to VM3.
- While the session is running, terminate it from VM1 by sending a TCP reset. VM1 needs to be in promiscuous mode (see page 2).
- Use netwox on VM1 to automatically reset any telnet TCP connection from VM2 to VM3, but do not attack other protocols. If successful, **`telnet [IP of VM3]`** should consistently fail on VM2 but **`ssh [IP of VM3]`** should still work.

The attack itself is easy (you just have to look up the netwox commands), the main task is to document what is happening (e.g. "each packet with property X from Y to Z causes VM1 to send out a packet with contents W") and what effect this is having (e.g. on the TCP state machines of the various VMs).

Optional extension:

- Connect to the internet. Stop VM3 from connecting to facebook.com through the browser but interfere as little as possible with other connections (in particular, bristol.ac.uk should still load normally).

Make sure you're targeting VM3 and not facebook!

You have now implemented your own mini version of the "Great Firewall". For your report, you may want to look up some countermeasures that people are using to get around the Great Firewall's RST packets.

Task 4 – TCP session hijacking

If this doesn't convince you never to use telnet again, nothing will.

Warm-up: Put VM1 in promiscuous mode. Open a telnet connection from VM2 to VM3 and type some commands – on VM1, observe the password in the clear and the packets and replies sent. How are commands "buffered"?

Attack: Create a file called "file" on VM3. With VM2 connected to VM3 through telnet but otherwise idle, use netwox on VM1 to inject the command **`rm file`** into the telnet session.

- The trick here is to set the sequence number(s) correctly.
- What happens if you try and continue the session on VM2, and why?

Note: wireshark displays "relative" sequence numbers by default, that is offsets to the initial sequence number of the connection. You can turn this option off in Edit/Preferences by going to Protocols/TCP and unchecking "Relative Sequence Numbers". Or just use "**`netwox 7`**" to observe the network traffic.