Introduction to InfoSec – Recitation 12

Nir Krakowski (nirkrako at post.tau.ac.il)
Itamar Gilad (itamargi at post.tau.ac.il)

Today

- ARP in a nutshell
- ARP Poisoning
- Ping reflection (smurf attack)
- MAC Flooding
- Detecting promiscuous hosts
- Ping / traceroute using other protocols
- Firewalking

ARP in a Nutshell

- ARP = Address Resolution Protocol
- A bridge between IP and Ethernet, which helps make a local network "work"
- Most important functionality – translate IP addresses to MAC addresses so we can actually send packets!
- Two major messages –
  - ARP request – "Who is at 192.168.1.1?"
  - ARP reply – "192.168.1.1 is at A1:B2:C3:D4:E5:F6"

ARP Poisoning

- To avoid making an ARP request before sending every IP packet, each host has a local cache.
- Another trick to avoid excessive ARP requests, is that every host will send a broadcast ARP **reply** when it comes online / every interval, to let everyone know its MAC address (known as "Gratuitous ARP")
- Most implementations are state-less by design, and will happily store ARP replies **even if they didn't issue a request** (for reasons stated above)
- Result – everyone on the local network can impersonate any other host, by sending a
- malicious ARP reply in their name.

ARP Poisoning

- Attack scenario –

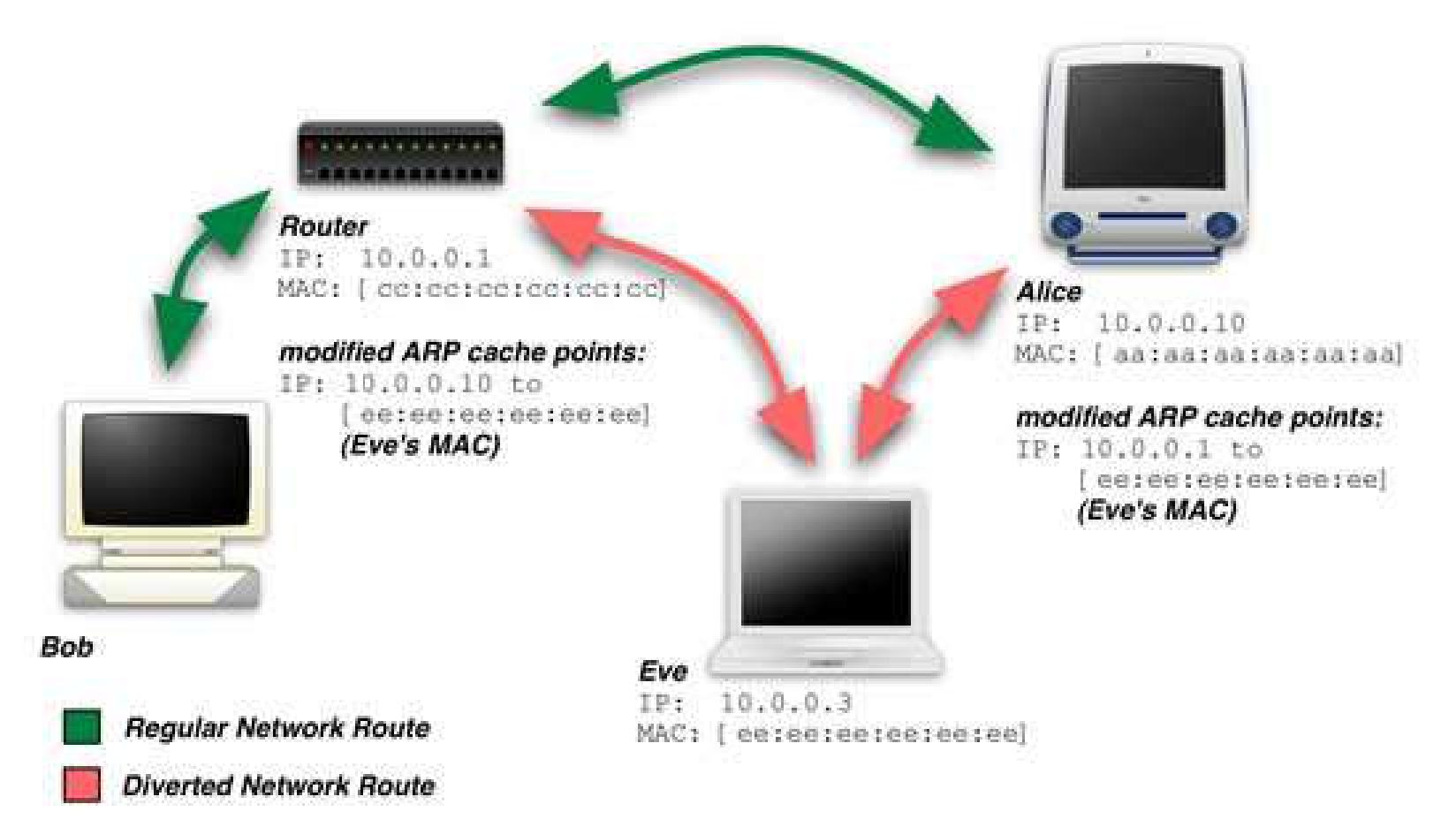

Diagram Copyright: http://gm.kochar.com/post/WEB-SPOOFING.aspx

Ping Reflection ("smurf attack")

- We want to DoS a host, but we're not fast enough…
- So we'll get everyone else to join!

- Basic concept – send a ping request to everyone, but put the target's IP address in the source of the packet.
- Result - everyone

will send a reply to the target, effectively DDoSing it.

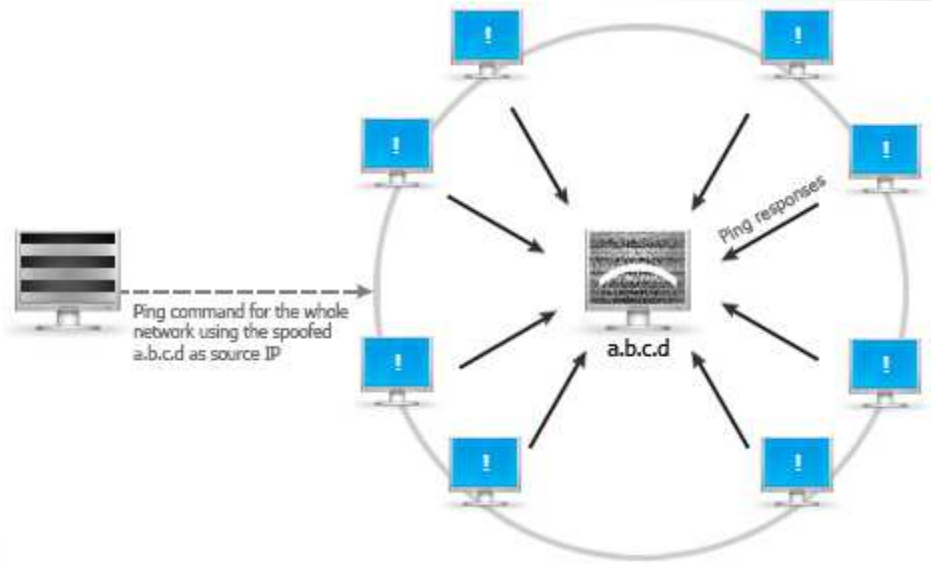

Diagram copyright: http://onlinehelp.avs4you.com/AVS-Firewall/Introduction/NetworkAttacks.aspx

No more sniffing…

- It used to be easy to sniff traffic on the local network
- All traffic went to everyone behind the same router on a HUB based network
- Now – switches galore!
- We still need to sniff traffic…

- Enter MAC Flooding

Switches 101

- Switches know where to route packets by learning which MAC addresses are connected to which port
- This is done by seeing which source MACs appear on which ports, and storing this information in a fast look-up table (CAM)
- This table has to be very fast, so it must be limited in size.
- This is not an issue, since It is highly unlikely to run more than a few 100's / 1000's of hosts on the same layer-2 network due to other reasons.

MAC Flooding

- We're on a network, but that network uses switches, so we can't sniff anything interesting…
- Or can we?

- What happens if we send out packets with different source MAC addresses? Will the switch refuse to learn new addresses?
- No! it will just fail-over to operating like a hub – a 'dumb' repeater

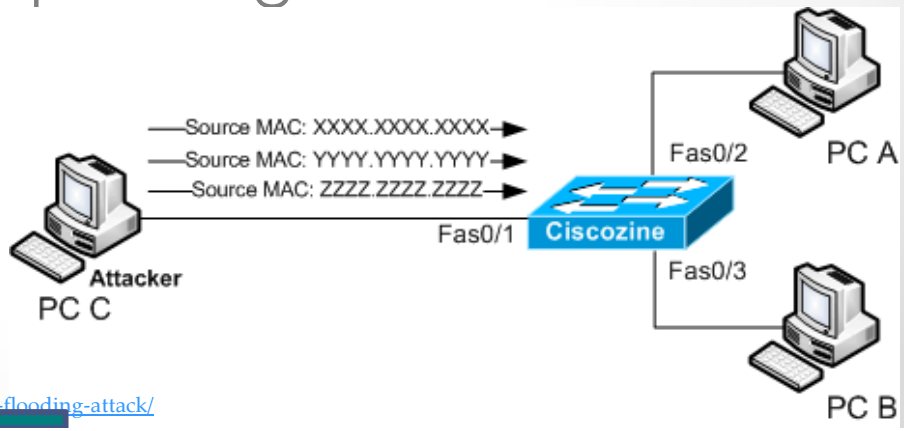

Image Copyright: http://www.ciscozine.com/2009/01/05/protecting-against-mac-flooding-attack/

Promiscuous mode

- Normally, the network card will listen to every incoming packet, and discard any packet whose destination MAC address is not its own.
- When someone is running a sniffer, they'll want to capture as much information as possible about the network.
- Network cards can support this by going into what's called "Promiscuous mode" – where every packet received is sent to the OS for further processing.

Detecting Promiscuous Hosts

We want to detect if someone on our network is using a sniffer in promiscuous mode.

The trick –

- Send out a ping request with the wrong destination MAC address, but the right IP target (or broadcast).
- Regular hosts will discard the packet, **but anyone in promiscuous mode will reply**, since the IP target was valid

Ping / Tracerout Using Different Protocols

- Let's assume TCP SYN / ICMP Echo requests are monitored / blocked but you still want to know if a host is up, and/or what are the network elements between you and the target (traceroute)

- ARP Ping –
  - Send an ARP request for a host on the same subnet(can even use broadcast)
  - If you get a reply – that host is alive

- TCP Port Scan –
  - Instead of using a SYN packet, use a TCP data packet, and listen for an RST packet

Ping / Tracerout Using Different Protocols

- UDP traceroute –
  - You already found out that the host will send you an ICMP Port Unreachable message when you send a UDP datagram to a certain closed port
  - But you want to find all the elements in the way
  - Solution – send and resend the packet, each time with different IP TTL
  - You will get ICMP errors from many intermediate hosts
- TCP traceroute –
  - Same as UDP, and can use SYN on an known open port, arbitrary data packet on a known open port, or data on a known closed port
- Basically – most services could be used for traceroute / ping given the right scenario

Firewalking

- We want to learn which ports/subnets are filtered
- If there is a rule to drop a packet, we'll get no reply
- If the packets can reach past the firewall, we still need to get everything else valid…

- Solution – use TTL!
- Set packets to TTL of the FW + 1
- If we get an ICMP error packet (TTL exceeded) – our packet got through!

This week's exercise

- Implement some of these techniques
- Be careful about affecting your network
- Don't abuse on other networks – you are responsible for any damage you create

Questions?

Extra

- IPv6 –
  - DHCP → RA
  - ARP → NDP
  - Interop / transition –
    - Dual stack
    - Tunneling
- VLAN Hopping
  - Switch / trunk spoofing
  - Double encapsulation
- Spanning Tree takeover
- DNS Poisoning