Network Attacks

Offensive Network Security
Florida State University
Spring 2014

Outline

- Attacking the Network
- Man in the Middle Attacks (MitM)
  - ARP Spoofing/Poisoning
  - Port stealing
  - DHCP Spoofing and DHCP Exhaustion
  - DNS Spoofing
- Denial of Service (Distributed)
- Protocols causing information leakage

Network Attacks

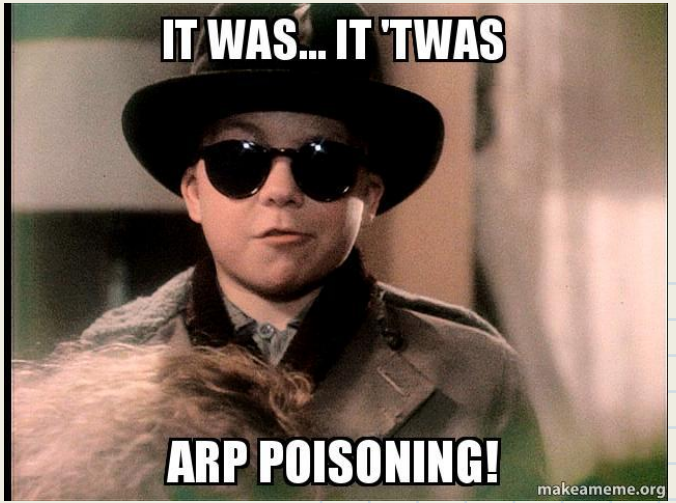
- Inside Attacks
- Outside Attacks
- Passive Attacks
  - Gain information by not altering or disabling system resources
  - Port Scanning
  - Eavesdropping/Wiretapping
- Active Attacks
  - MitM
  - Smurf Attack (DoS attack)
    - Named after the original source: smurf.c
    - Spoofed ICMP attack from multiple hosts to victim
  - Buffer/Stack/Heap Overflow

Eavesdropping

- Sometimes the best way to gain information is to *listen*
- Setup a passive device that captures all broadcasted packets
  - What information can be gained from this type of attack?
  - DHCP requests
  - Locate unused IP addresses in subnet
    - How is this beneficial to the attacker?
    - Use unused IP address with random MAC address
  - MAC address collection (ARP)
    - Gateway
    - Hosts
  - UPnP notifications/searches
- What other information can be captured?

ARP Poisoning

- MitM Attack
- Users lying to Ethernet switches and machines about their identity
- Manipulates ARP protocol
- Capture client network traffic
  - Between client-to-client
  - Between client-to-gateway
- Create new path with unused IP/Random MAC
  - Quick communication path to send info out
  - Control communication
- Benefits
  - Capture usernames/passwords
  - Inject keyloggers
  - Other “fun” tasks



ARP Poisoning: Tools

- ARP Cache Poisoning Tools
  - Ettercap
  - Cain and Abel
  - Subterfuge
  - Intercepter-NG
  - nping/hping3
  - Scapy
- We'll only cover a few of these tool in these slides

ARP Poisoning: Ettercap

- `ettercap -T -M arp:remote /192.168.23.1/ //`
  - ARP Poison between gateway and all hosts in LAN
  - More hosts, more network traffic
- `ettercap -T -M arp:remote /192.168.23.1/ /192.168.23.12/`
  - ARP Poison between gateway and specific host in LAN
  - Reduce network traffic
- `ettercap -T -M arp:remote //25`
  - Sniff only SMTP connections
- Ettercap options
  - -T: options puts ettercap into text-mode
  - -M: MitM attack (i.e. ARP, DHCP, DNS)

ARP Poisoning: nping

- How many command line options must be used to perform a nping ARP poison?
- Are these options enough to perform the correct attack?
  - `nping --arp --arp-type 2 --arp-target-ip 192.168.23.34 --arp-target-mac "a9:9a:bb:cc:96:00"`
  - nping attack will need to be looped or else risk the route being "corrected" by correct destination MAC
  - What about Ethernet header values (src, dst)?

ARP Poisoning: Scapy

- Scapy has a built-in function: `arpcachepoison(target, victim, interval=60)`
- How can we build our own ARP cache poison packet?
  - Left for assignment exercise

ARP Poisoning: Packet

- What are the goals of a successful ARP poison?
  - Ability to read victim data
  - Ability to control packet paths in LAN
- What ARP packet fields can to be spoofed for attack
  - Operation
  - Hardware address (Source/Destination)
  - Protocol Address (Source/Destination)
  - Messing with the length in this case is no good for **Ethernet** ARP Poisoning
- Can an ARP Reply be used to perform an ARP Poison?
- Can an ARP Request be used to perform an ARP Poison?
- Can a Gratuitous ARP be used to perform an ARP Poison?

ARP Poisoning: Packet

0x0001		0x0800	
0x06	0x04	0x0001	
0x77	0x6A	0x9F	0xA2
0x11	0x00	0xC0	0xA8
0x40	0x12	0x00	0x00
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

ARP Poisoning: Packet

0x0001 (hwtype)		0x0800 (ptype)	
0x06	0x04	0x0001	
0x77	0x6A	0x9F	0xA2
0x11	0x00	0xC0	0xA8
0x40	0x12	0x00	0x00
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

ARP Poisoning: Packet

0x0001 (hwtype)		0x0800 (ptype)	
0x06 (hwlen)	0x04 (plen)	0x0001	
0x77	0x6A	0x9F	0xA2
0x11	0x00	0xC0	0xA8
0x40	0x12	0x00	0x00
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

ARP Poisoning: Packet

0x0001 (hwtype)		0x0800 (ptype)	
0x06 (hwlen)	0x04 (plen)	0x0001 (operation)	
0x77	0x6A	0x9F	0xA2
0x11	0x00	0xC0	0xA8
0x40	0x12	0x00	0x00
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

ARP Poisoning: Packet

0x0001 (hwtype)		0x0800 (ptype)	
0x06 (hwlen)	0x04 (plen)	0x0001 (operation)	
0x77 (hwsrc)	0x6A	0x9F	0xA2
0x11	0x00	0xC0	0xA8
0x40	0x12	0x00	0x00
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

ARP Poisoning: Packet

0x0001 (hwtype)		0x0800 (ptype)	
0x06 (hwlen)	0x04 (plen)	0x0001 (operation)	
0x77 (hwsrc)	0x6A	0x9F	0xA2
0x11	0x00	0xC0 (psrc)	0xA8
0x40	0x12	0x00	0x00
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

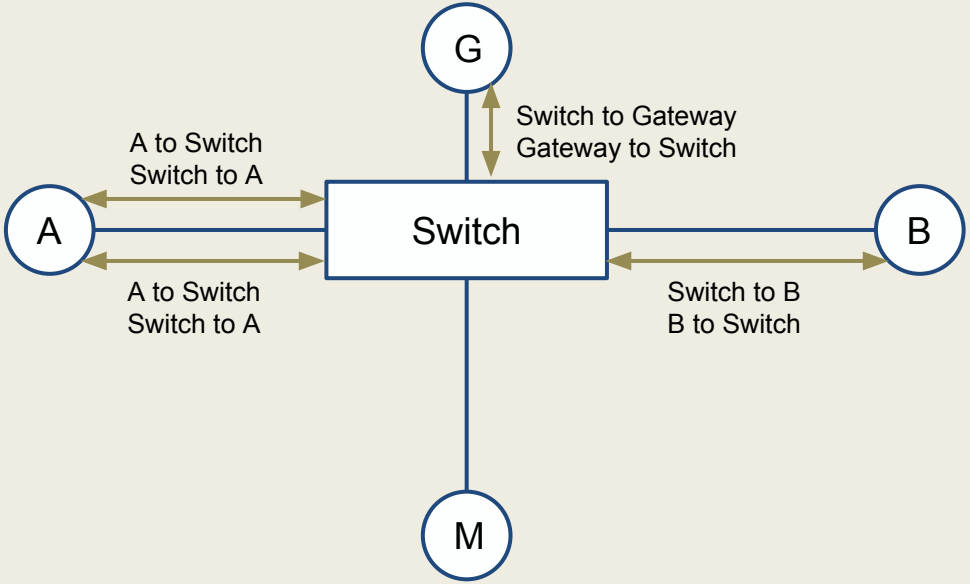
ARP Poisoning: Packet

0x0001 (hwtype)		0x0800 (ptype)	
0x06 (hwlen)	0x04 (plen)	0x0001 (operation)	
0x77 (hwsrc)	0x6A	0x9F	0xA2
0x11	0x00	0xC0 (psrc)	0xA8
0x40	0x12	0x00 (hwdst)	0x00
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

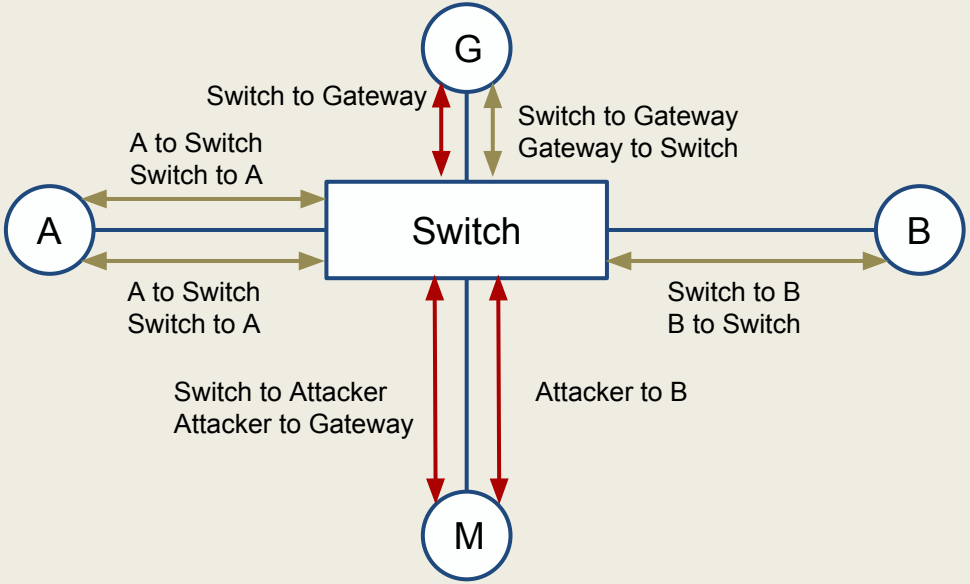
ARP Poisoning: Packet

0x0001 (hwtype)		0x0800 (ptype)	
0x06 (hwlen)	0x04 (plen)	0x0001 (operation)	
0x77 (hwsrc)	0x6A	0x9F	0xA2
0x11	0x00	0xC0 (psrc)	0xA8
0x40	0x12	0x00 (hwdst)	0x00
0x00	0x00	0x00	0x00
0x00 (pdst)	0x00	0x00	0x00

ARP Poisoning: Attack



ARP Poisoning: Attack



ARP Poisoning: Attack

- What would happen if M did not ARP poison the Gateway?
  - ARP Poison will fail when Gateway sends a packet to A
  - Must update both end-nodes to capture both paths of communication
- What would happen if M did not ARP poison B?
- Will M need to keep sending ARP packets to host-victims to guarantee he does not miss any data?
- Think-back: When does an Ethernet Switch update it’s internal table?

Port Stealing

- **NOT** TCP/UDP port *but* Ethernet Switch physical port
- Does an attacker need to ARP poison to intercept traffic?
- Does an Ethernet switch understand ARP?
  - Only cares about Port => MAC address translation
  - Stored in table (CAM)
- How can an attacker redirect traffic without ARP?
  - How can an attacker send an Ethernet frame to itself?
  - How can attacker "**train**" it's connected Ethernet Switch?
- `ettercap -T -M port:remote /192.168.33.1/ /192.168.33.5/`
  - Intercept traffic between both IPs
  - Intercept traffic destined for either IPs
- `ettercap -T -M port:remote /192.168.33.5/`
  - Intercept traffic destined for IP

DHCP Spoofing

- DHCP extends BOOTP
- DHCP *different* between IPv4 and IPv6
- DHCP broadcasted across network
- Purpose is to **lease** an IP address and obtain GW/DNS
  - Also used to **renew** an already leased IP address
  - This request will be unicast to DHCP server
- Attacker, victim, and DHCP server in a race condition
- Attacker can try to reply *before* DHCP server
- Don’t worry about checking use of IP address, save time
- Only the client is spoofed/poisoned (Half-Duplex attack)
- Once assigned, will not change IP until DHCP timeout
- DHCP uses UDP and ports 67 and 68

DHCP Spoofing

- What is gained from this attack?
  - Give out an IP address attacker controls
  - Give out Gateway address attacker controls
  - Give out DNS address attacker controls
- The attacker can now capture all packets from client to gateway
- What about Gateway to client?

DHCP Spoofing: Tools

- Ettercap
- Scapy
- Can we use a tool like `nping` to send out DHCP spoof replies,discoveries?

DHCP Spoofing: Ettercap

- `ettercap -T -M dhcp:ip_pool/netmask/dns`
  - ip_pool must be from **free** address
  - will not check if assigned IP is free
- `ettercap -T -M dhcp:192.168.33.2-30/255.255.255.0/192.168.33.1`
  - Assigned clients address 192.168.33.2-30/24
  - DNS server is 192.168.33.1
  - GW is current IP address of attacker
- `ettercap -T -M dhcp:/255.255.255.0/192.168.33.1`
  - Only reply to DHCP request
  - Will not offer IP addresses
  - Will modify GW/DNS

DHCP Spoofing: Tools

- How can we use Scapy for DHCP spoofing?
- Scapy defines two layers: `BOOTP()` and `DHCP()`
- BOOTP() layer has most of the typical options

DHCP Spoofing: Packet

- op: Request (0x01), Reply (0x02)
  - Discovery: client looking for DHCP servers
  - Offer: client requesting IP address
  - Request:
  - Acknowledgement
- htype: (Ethernet, ARCNET, Frame Relay, etc.)
- hlen: Sizeof(MAC Address)
- hops: How many DHCP servers forwarded request
- xid: Transaction identifier set by client
- secs: Seconds since client started requesting
- flags: Not really used

DHCP Spoofing: Packet

- CIAddr: Client requesting IP address
- YIAddr: Server leased address to client
- SIAddr: DHCP Server IP
- GIAddr: Gateway of network
- CHAddr: client hardware address
- sname: server name
- file: boot file to be downloaded
- options: vendor options, DHCP Magic value: 0x63825363

DHCP Spoofing: Packet

- DHCP Options
  - Message-Type: discover, offer, request, decline, ack, nak, release, inform, force-renew, lease-query, lease-unassigned, lease-unknown, lease-active
  - hostname
  - NTP_server
  - renewal_time
  - tcp_ttl
  - end
- DHCP extended BOOTP to add more messaging types
- Many options can be set, the last option must be end

DHCP Spoofing: Packet

0x01	0x01	0x06	0x00	0xFF334400
0x10		0x00		0x00000000
0x00000000				0x00000000
0x00000000				0x00000000000000000000000000000000
0x00000000000000000000000000000000 00000000000000000000000000000000				0x00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000
0x00000				

DHCP Exhaustion

- What prevents a client from requesting multiple IP addresses from server?
- What happens to the DHCP server when it exhausts all IPs?
- What DHCP operations can be used to exhaust DHCP IPs?

DHCP Spoofing: Thought

How could an attacker utilize DHCP to determine IP subnet, gateway and DNS server but then make the server drop the lease request?

ICMP Redirection

- Utilize ICMP Redirect message to announce better route
- Attacker pretends to be a better route for Internet
- Attack
  - Force hosts to send to a different GW
  - Attacker forwards to real GW
- GW does not accept ICMP redirection
- Half-Duplex attack since GW is prevented from sending packets
- Can be defeated by disabling ICMP Redirect

ICMP Redirection: ettercap

- `ettercap -T -M icmp:00:11:22:33:44:55/192.168.33.1`
  - Arguments passed are Gateway MAC/IP
  - Redirect all connections passing through the GW

References

- http://technet.microsoft.com/en-us/library/cc959354.aspx (Network Attacks)
- http://blog.zorinaq.com/?e=6