Post 3

Thursday, January 24, 2019 11:10 PM

MAC TABLES, CAM OVERFLOW, DHCP SPOOFING AND CDP LEVERAGE ATTACKS

Common Security Attacks:

- Security: Layered process: Never complete: More awareness is always better

MAC tables (recap)	▪ Contains MAC addresses associated w/each physical port & associated VLAN for each port ▪ When Layer 2 switch receives frame: Switch looks in MAC table for destination ▪ As frames arrive on ports: Source MACs are recorded in table **If entry exists for address:** Switch forwards frame to correct port **If entry doesn't exist in table:** Switch floods frame out of every port except ingress (broadcast)

MAC table overflow (aka) **CAM table overflow attack:**

- Be aware of age-out periods of MAC tables when performing audits
- If spoofed MAC addresses start to age-out while performing an audit, valid MAC's can populate table

MAC Flooding	▪ Broadcast behavior for unknown addresses can be used for attack ▪ Tables limited in size ▪ Switch is overwhelmed w/fake source MAC's until table is full ▪ Frames sent are randomly-generated source/destination MAC's to switch ▪ Switch enters fail-open mode **Fail-open mode:** Switch broadcasts all frames to all machines on network (which can be seen) ▪ Tools can generate up to 155K MAC entries on switch per minute ▪ As long as MAC table remains full: Switch broadcasts all received frames out every port **Mitigation**: Configure port security 1. Host A sends traffic to host B 2. Switch receives frames & looks up destination MAC in table 3. Switch copies frame: Floods (broadcasts) every port except ingress 4. Host B receives frame: Sends reply to host A 5. Switch learns MAC for host B is located on X port: Records to table 6. X receives frame from A to B but b/c dest. MAC of frame is B: Host X drops frame

DHCP Spoofing

DHCP: Automatically assigns hosts valid IP's out of DHCP pool

2 types of common attacks performed on switched networks:

1. DHCP starvation
2. DHCP spoofing

Starvation	▪ Attacker floods DHCP server w/DHCP requests ▪ This floods all available IP's DHCP server can issue ▪ Once issued the server can't issue any more IP's, which produces a DoS ▪ New clients can't obtain network access **DoS (Denial of Service):** Any attack that overloads specific devices/network services

	w/illegitimate traffic (preventing legitimate traffic from those resources)
Spoofing	▪ Attacker configs fake DHCP server on network to issue IP's to clients Reason: Force clients to use false DNS/WINS servers (Windows Internet Naming Service). They must use attacker's machine (or one controlled by) as default gateway ▪ Starvations typically come first to deny service to a legitimate DHCP server. That makes it easier to introduce the fake one **Mitigation:** DHCP snooping/port security features on switches

Leveraging CDP:

CDP	○ **Cisco Discovery Protocol:** Proprietary: All Cisco devices can config to use ○ Discovers other Cisco devices directly connected ○ Allows devices to auto-configure their connection ○ By default most Cisco routers/switches have CDP-enabled on all ports ○ Info is sent in periodic unencrypted broadcasts ○ Info is updated locally in CDP db's for each device ○ Layer 2 protocol: Messages aren't propagated by routers ○ Contains info about the device: IP/IOS version/platform/capabilities/native VLAN ○ This info can be used for a DoS attack

CDP DoS:

CDP	▪ Wireshark captures can show contents of a CDP packet ▪ IOS versions can determine security vulnerabilities ▪ It's not authenticated: You could craft bogus CDP packets & send them to a device **Mitigation:** Disable the use of CDP on devices/ports that don't need it no cdp run (global config) (can be disabled on a port/port basis)

Telnet attacks:

Telnet	○ Insecure/unencrypted: Can gain remote access to device ○ Tools: Brute force attacks against VTY lines on switch

Brute Force Password Attack

Brute Force	○ Uses list of common passwds/designed to establish Telnet session using each word on dictionary list ○ If the password isn't discovered: ○ Program creates sequential character combinations in attempt to guess password

Telnet DoS: Exploits flaw in Telnet server software on switch so service is unavailable

- Prevents admin from remotely accessing switch management
- Can be combined w/attacks to prevent admin from core devices during breach
- Usually addressed in patches included in newer IOS revisions

Practices	○ Written security policy/Shut down unused services/ports ○ Strong passwords/changed often/control physical access to device ○ Avoid HTTP/Use HTTPS/Perform backups/test back up files regularly ○ Educate employees/Encrypt sensitive data/Implement security HW/SW ○ Keep security patches up to date ○ Carry out audits in a controlled environment/document procedures ○ Off-line test bed network that mimics is ideal

Disabling unused ports:

- shutdown cmd on every unused port
- no shutdown cmd can re-enable it
- interface range cmd can be used to configure multiple ports

switch(config)# interface range *type module/1st-number – last-number*

DHCP Snooping	▪ Determines which ports can respond to DHCP requests: Trusted/untrusted ▪ Rogue device on untrusted port attempts to send DHCP packet: Port is shut down ▪ DHCP binding table built for untrusted ports ▪ Entries contain: client MAC address/IP/lease/binding type/VLAN #/port ID recorded ▪ Table filters DHCP traffic

	Trusted: Host DHCP server/uplink/source all DHCP messages/offer ACK packets **Untrusted:** Source requests only

DHCP Snooping Config:

1. ip dhcp snooping (global config)
2. Specific VLANS: ip dhcp snooping vlan *number*
3. Define ports as trusted at int lvl by defining them: ip dhcp snooping trust
4. Limit rate of continuously sent bogus DHCP requests: ip dhcp snooping limit rate

Port Security

- Limits # of valid MAC's allowed on port
- MAC's of legitimate devices allowed: Other MAC's denied.
- Can be configured to allow 1/more MAC's
- If # of MAC's is limited to 1: Only device w/specific MAC can connect to port
- If max # reached: Additional attempts by unknown MAC's generate security violation
- Port security won't work until enabled on int using switchport port-security cmd

Secure MAC Address Types: Type of secure based on config/includes:

Static secure	○ Manually configured on a port ○ switchport port-security mac-address *mac-address* cmd (int config) ○ MAC's stored in address table/added to running-config
Dynamic secure	○ Dynamically learned/stored only in table ○ MAC's configured this way removed on restart
Sticky secure	○ MAC's can be dynamically learned/manually configured ○ Stored in table/added to running-config ○ **Must enable sticky learning**: switchport port-security mac-address sticky (int config) ○ Switch converts dynamically learned MAC's (even before sticky), to sticky MAC's ○ **Manually defined?** switchport port-security mac-address sticky*mac-address* (int config) ○ Specified addresses added to table ○ **If saved to startup-config:** Switch restarts/int shuts down: Int doesn't need to relearn ○ **If sticky disabled by using** no switchport port-security mac-address sticky (int config): ○ MAC's remain part of table, but removed from running-config

Violation modes: Occurs when the following happens (security violation):

- Max # of secure MAC's added to table for that int & station whose MAC isn't in table attempts to access int
- Address learned on 1 secure int is seen on another secure int in the same VLAN
- int can be configured for 1 of 3 violation modes, specifying action to be taken

Protect	**When # of secure MAC's reaches limit allowed on port:** Packets w/unknown sources dropped until sufficient # of secure MAC's removed, or # of max addresses increases ○ NO notification a security violation occurred
Restrict	**When # of secure MAC's reaches limit allowed on port:** Packets w/unknown sources are dropped until sufficient # of secure MAC's removed, or # of max addresses is increased ○ Notification a security violation has occurred
Shutdown	**Default violation mode:** ○ Port security violation causes int to immediately become error-disabled /turns off port LED ○ Increments violation counter ○ **When secure port in error-disabled**: It can be brought out by: shutdown & no shutdown (int config)

Change violation mode on port: switchport port-security violation {protect | restrict | shutdown} (int config)

Display port security settings for switch/specified int: show port-security [interface interface-id]

- Default: 1 MAC address allowed on this port
- Sticky MAC's added to table and running-config

Verify Secure MAC's: show port-security address

- MAC's listed along w/types

Error Disabled State	○ Shut down/no traffic is sent/received on port ○ Protocol/link status is changed to down ○ Port LED = **OFF** ○ show interfaces identifies as err-disabled ○ Output shows port status as secure-shutdown ○ B/c it's in shutdown: Port w/sec violation goes to error disabled state

Network Time Protocol (NTP)

NTP: Used to sync clocks of systems over packet-switched, variable-latency data networks

- Allows devices to sync time settings w/NTP server
- NTP clients obtain time/date info from single source: More consistent settings
- Required to accurately track events like sec violations
- Critical for interpretation of events w/in syslog files & digital certs
- Admins can implement private network master clocks/sync'd to UTC/using satellite/radio
- If they don't want to b/c of cost: Resources available on Internet

NTP can get correct time from internal/external source:

Local master clock	Master clock on the Internet	GPS or atomic clock

- A device can be configured as an NTP server/client
- **Synchronized by an NTP server:** ntp server ip-address command (global config)

To config device as NTP master clock/peers can sync themselves: ntp master [stratum] (global config)

Stratum value: A number from 1-15 & indicates NTP stratum number that system will claim

- If system is configured NTP master & no stratum #: It will default to stratum 8.
- If NTP master can't reach any clock with a lower stratum number: System will claim to be sync'd at configured stratum number and other systems will be willing to synchronize to it using NTP

IP of peer devices sync'd to peer/statically configured peers/stratum number: show ntp associations (PRIV EXEC)

Display NTP sync status/peer device sync'd to/which NTP strata device is functioning: show ntp status (USER EXEC)