Vulnerability: mDNS Detection

Wednesday, June 15, 2022 9:14 PM

(Remote Network).

CVE-CODE: CVE-2017-6520

Severity: Medium

What is mDNS?

In mDNS, there is no central DNS server. If you wish to query an IP whose hostname you are aware of, then you send a multicast message to all the devices in the network asking if any of them identify with the hostname. One of the devices will match the hostname that you are querying. It will then respond with its IP address (again via multicast, to all devices on the network). All the devices on the network can then update their local phonebook (mDNS cache), mapping the hostname with the local IP.

The multicast query is always executed on either IPv4 address 224.0.0.51 or IPv6 address ff02::fb. It is a UDP message, on port 5353. BTW, unicast DNS is also a UDP message, on port 53.

The resolution of hostnames to IP addresses is only allowed for hosts ending with .local suffix (as opposed to .com, .in, etc. extensions for normal websites). Hostnames ending with extensions other than .local are not processed by mDNS.

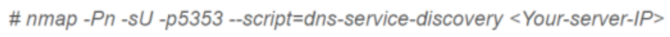
Apple Bonjour and Avahi software (on Linux distributions) implement both mDNS and DNS-SD. Windows 10 has also started providing native implementation for mDNS and DNS-SD.

Impact of mDNS:

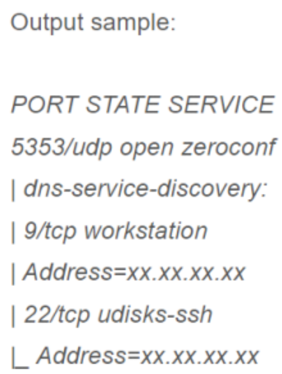
- It is possible to obtain information about remote host. The remote service understands the Bonjour (also known as ZeroConf or mDNS) protocol, which allows anyone to uncover information from the remote host such as its operating system type & exact version, its hostname, and the list of services running.
- This plugin attempts to discover mDNS used by hosts that are not on the network segment on which Nessus resides Filters the incoming traffic to UDP port 5353.

How to verify if the server is vulnerable?

Use the following command from a remote machine, as root, to query the mDNS service:



```
# nmap -Pn -sU -p5353 --script=dns-service-discovery <Your-server-IP>
```



Output sample:

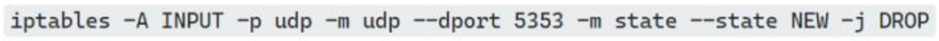
```
PORT STATE SERVICE
5353/udp open zeroconf
| dns-service-discovery:
| 9/tcp workstation
| Address=xx.xx.xx.xx
| 22/tcp udisks-ssh
|_ Address=xx.xx.xx.xx
```

If the command returns a time-out, the service might already be filtered.

Mitigation:

The best way to mitigate this risk is to disable mDNS on the server.

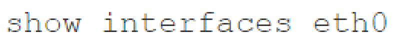
If mDNS must be used which is exposed to the internet, then the packet from the internet in the port 5353 must be dropped, which can be configured over firewall. A simple example of dropping 5353/UPD packet in Cent OS is show below.



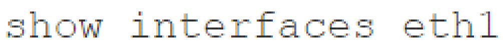
```
iptables -A INPUT -p udp -m udp --dport 5353 -m state --state NEW -j DROP
```

How to mitigate vulnerability mDNS Detection (Remote Network) on a cluster?

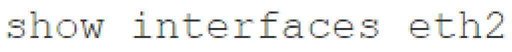
For each device in the cluster, confirm and note the following IP address



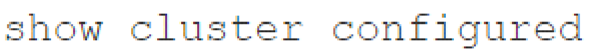
```
show interfaces eth0
```



```
show interfaces eth1
```

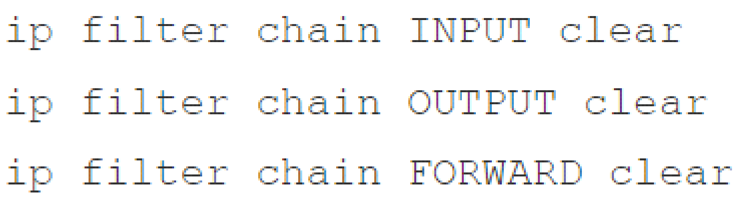


```
show interfaces eth2
```



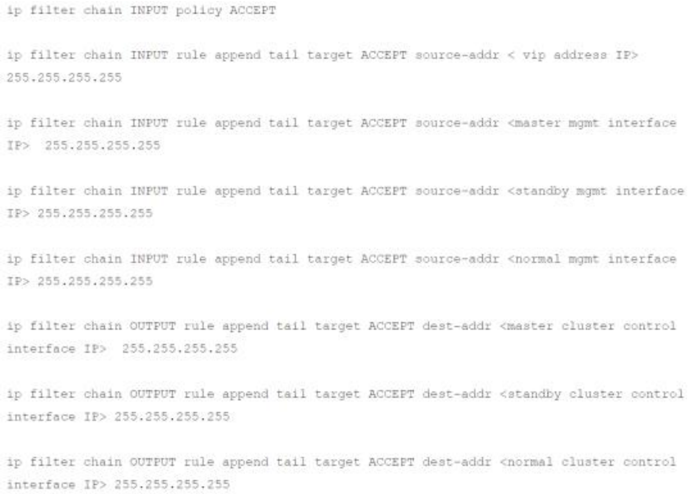
```
show cluster configured
```

For each device in the cluster, confirm and note the following IP address



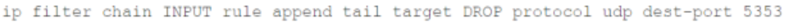
```
ip filter chain INPUT clear
ip filter chain OUTPUT clear
ip filter chain FORWARD clear
```

Using the following components, modifying the script below and apply to all nodes (via local Mgmt Port).
(a) captured interface IP addresses from each device (master node, standby node, slave nodes).
Set INPUT Policies on master device in the cluster and on each node in the cluster



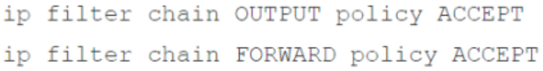
```
ip filter chain INPUT policy ACCEPT

ip filter chain INPUT rule append tail target ACCEPT source-addr < vip address IP>
255.255.255.255

ip filter chain INPUT rule append tail target ACCEPT source-addr <master mgmt interface
IP>  255.255.255.255

ip filter chain INPUT rule append tail target ACCEPT source-addr <standby mgmt interface
IP> 255.255.255.255

ip filter chain INPUT rule append tail target ACCEPT source-addr <normal mgmt interface
IP> 255.255.255.255

ip filter chain OUTPUT rule append tail target ACCEPT dest-addr <master cluster control
interface IP>  255.255.255.255

ip filter chain OUTPUT rule append tail target ACCEPT dest-addr <standby cluster control
interface IP> 255.255.255.255

ip filter chain OUTPUT rule append tail target ACCEPT dest-addr <normal cluster control
interface IP> 255.255.255.255
```

Append a rule to drop UDP traffic at port 5353



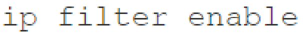
```
ip filter chain INPUT rule append tail target DROP protocol udp dest-port 5353
```

Set OUTPUT policy and FORWARD policy to accept



```
ip filter chain OUTPUT policy ACCEPT
ip filter chain FORWARD policy ACCEPT
```

Enable the IP filter (ACL) on each device



```
ip filter enable
```