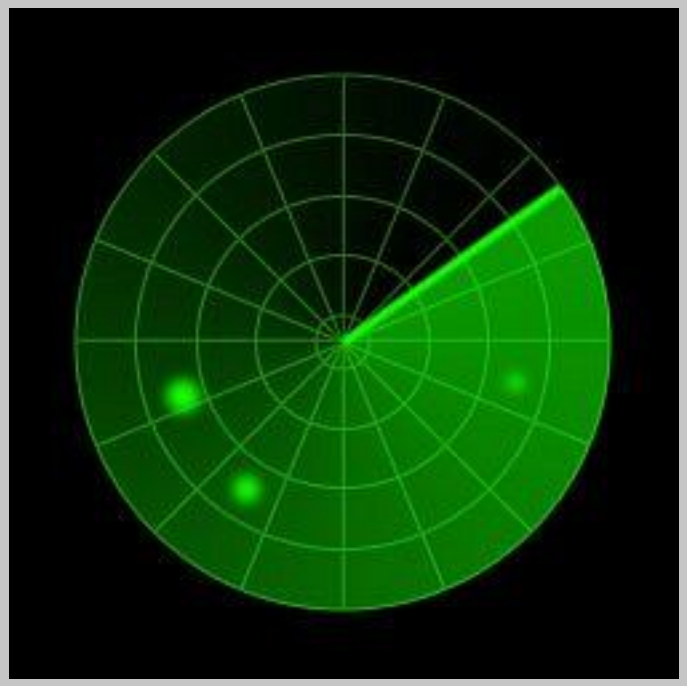
Port Scanning Without Sending Packets



DefCon 19, Las Vegas 2011



Hellfire Security

Gregory Pickett, CISSP, GCIA, GPEN
Chicago, Illinois

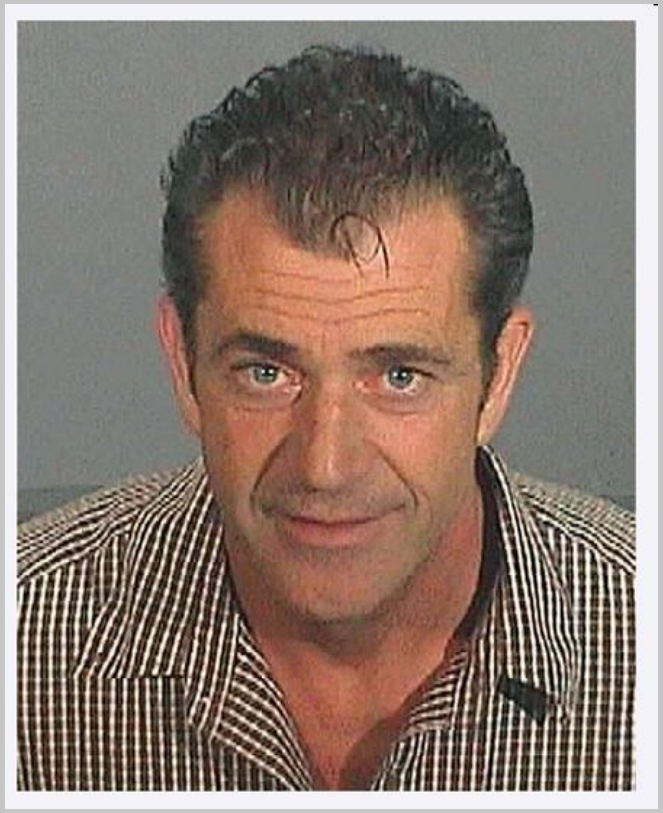
gregory.pickett@hellfiresecurity.com



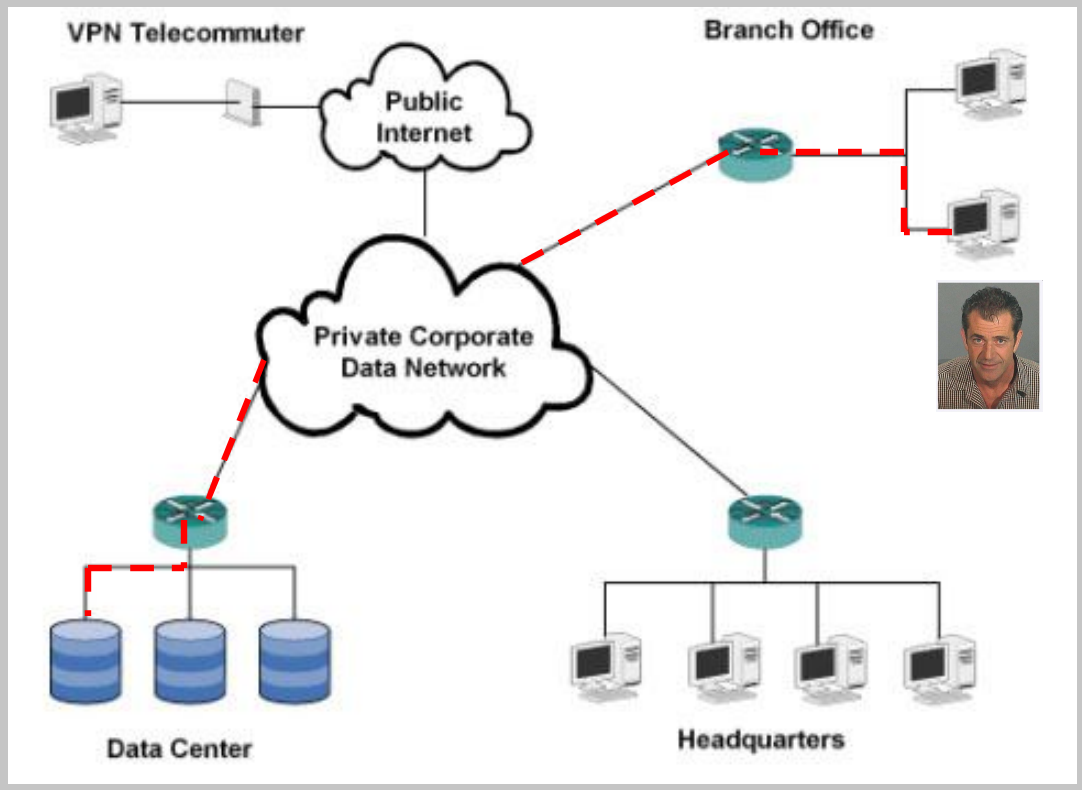
Overview

- How This All Started
- It's Not A Magic Trick
- Loose Lips Sink Ships
- Catch Me If You Can
- Back To The Future

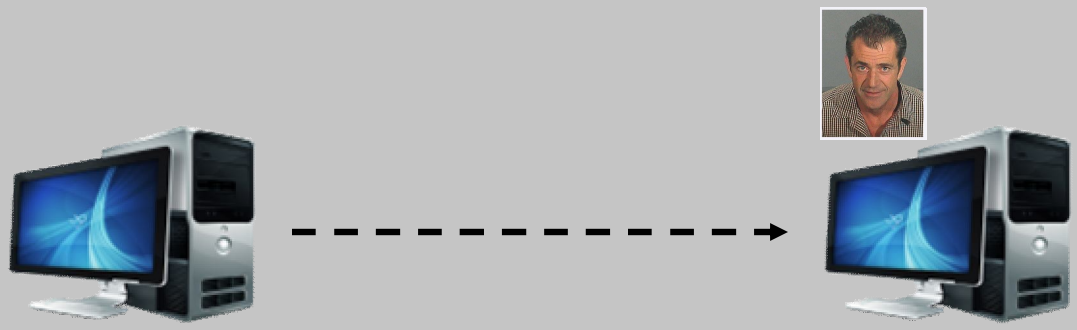
Suppose You Have This Guy On Your Network . . .



Suppose You Have This Guy On Your Network . . .



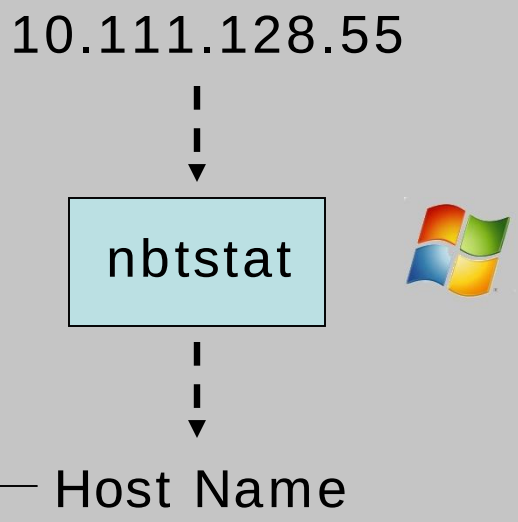
Suppose You Have This Guy On Your Network …



Host
Name?

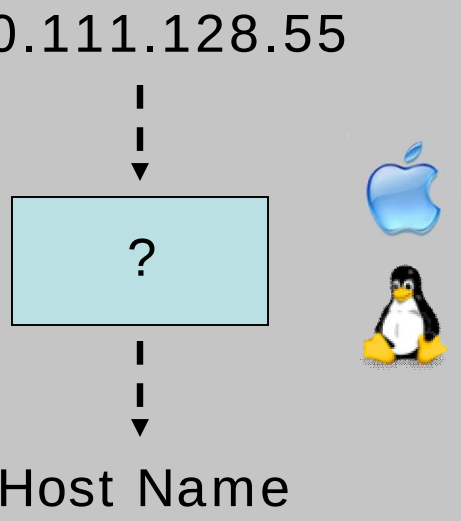
Suppose You Have This Guy On Your Network …

- Characterize
- Profile
  - Asset or Intruder
  - Role
  - Function
- Determination

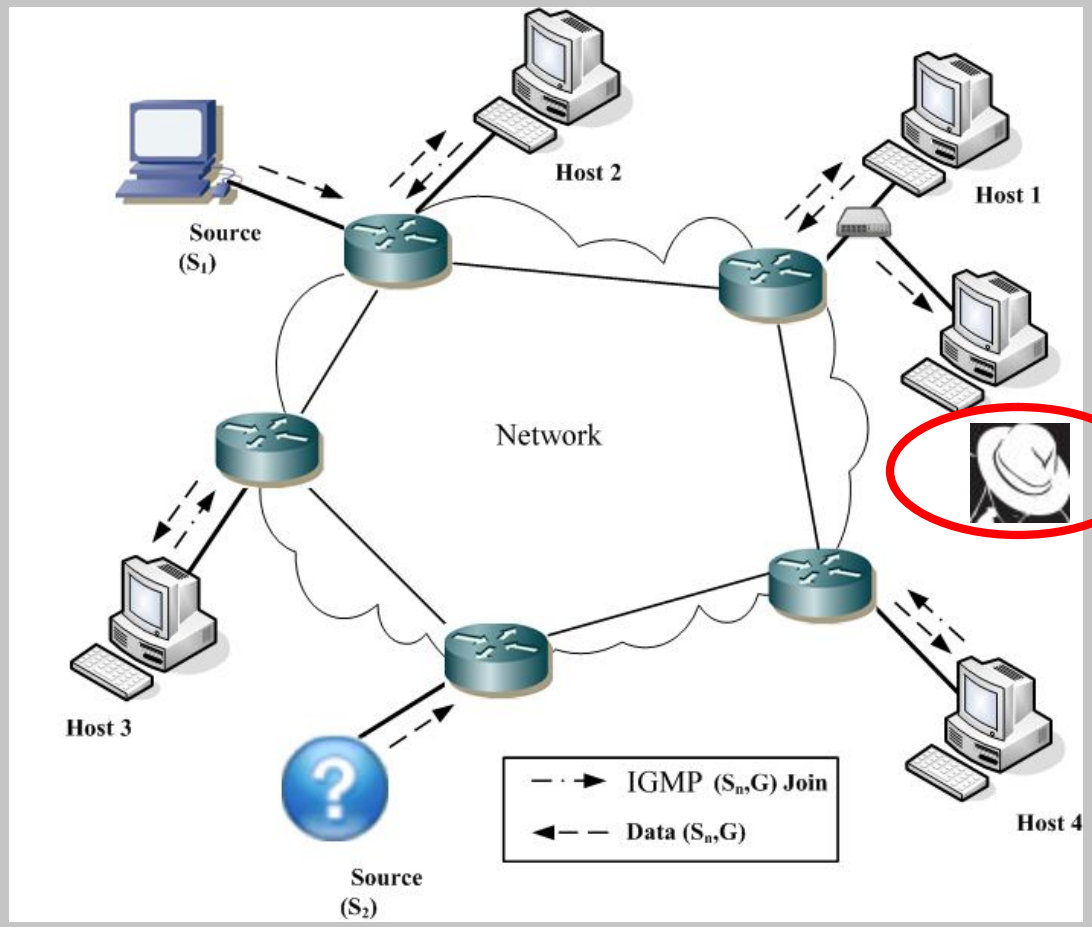


Suppose You Have This Guy On Your Network …

- Characterize
- Profile
  - Asset or Intruder
  - Role
  - Function
- Determination

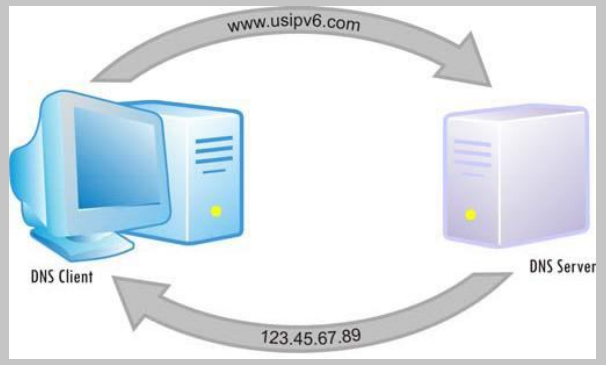


What is all this multicast?



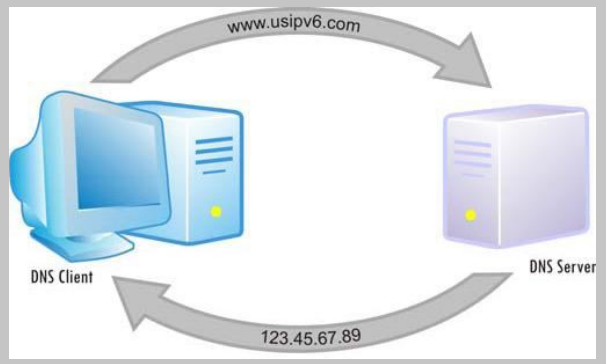
It's Multicast DNS (mDNS)!

- Purpose
  - Name Resolution (Peer-to-Peer)
- History
  - AppleTalk Name Binding Protocol
  - Zero Configuration Networking
- Development
  - Multicast DNS
  - DNS-Service Discovery



Features

- Messages
  - Same formats and operating semantics as conventional DNS
  - Based on “local” domain
  - Shared and unique records
- Operations
  - Queries and responses sent to 224.0.0.251
  - Utilizes UDP port 5353 for both resolvers and responders



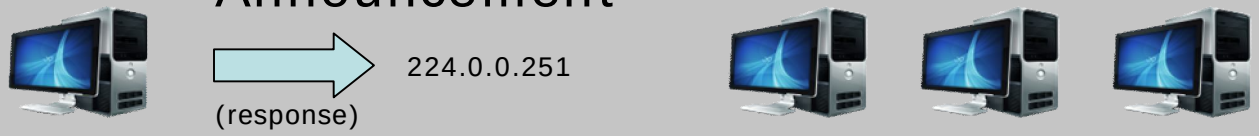
Usage

- Startup -

Probe

- For those resource records that it desires to be unique on the local link
- Proposed questions in the Authority Section as well
- Any “Type” record

Announcement



- All shared and unique records in answer section
- Unique have their cache-flush bit set
- Repeated any time should rdata change
- Unsolicited response

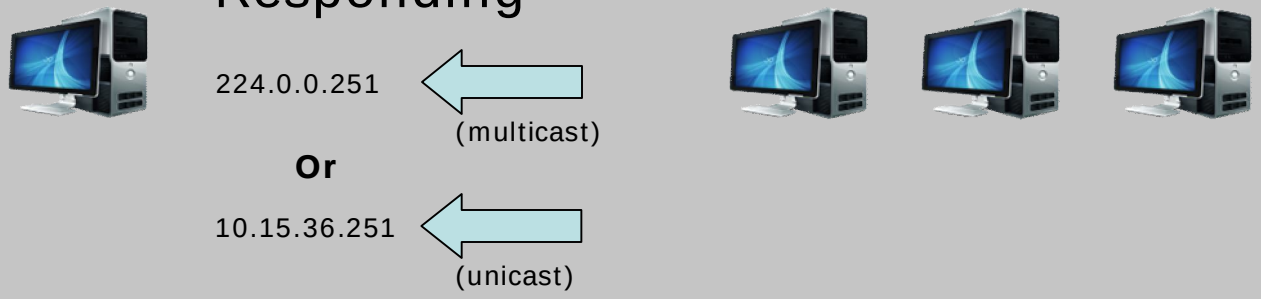
Usage

- Resolution -

Querying

- One-shot queries, and continuous ongoing queries
- Source port determines compliance level of the resolver
- Fully compliant resolvers can receive more than one answer
- Known answer suppression
- Truncation is used for large known answer set

Responding



- Mutlicast or unicast response per the query parameter
- Unicast queries are always treated as having the “QU” bit set
- Cache-flush bit indicates an authoritative answer
- No queries in any response

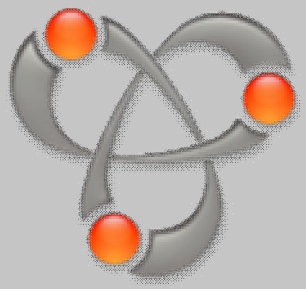
Usage

- Resolution -

- Used for changes on “Shared” records
- Not needed for unique records because of the cache-flush bit

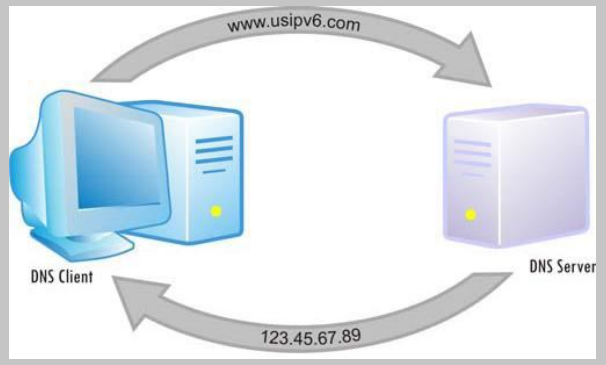
Implementations

- Apple
  - Rendezvous
  - Bonjour
    - Apple
    - Windows
- Avahi
  - Linux
- Others



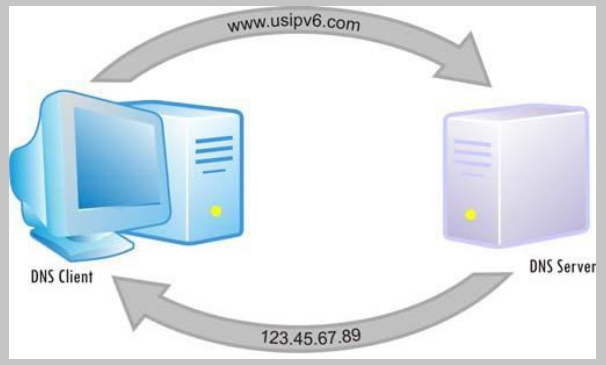
Names

- “PTR” Record
  - 135.148.16.172.in-addr.arpa
  - 7.A.F.A.E.B.E.F.F.F.A.4.6.2.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.E.F.ip6.arpa
- “A” Record
  - NPIBB0A88.local
- “AAAA” Record
  - NPIBB0A88.local



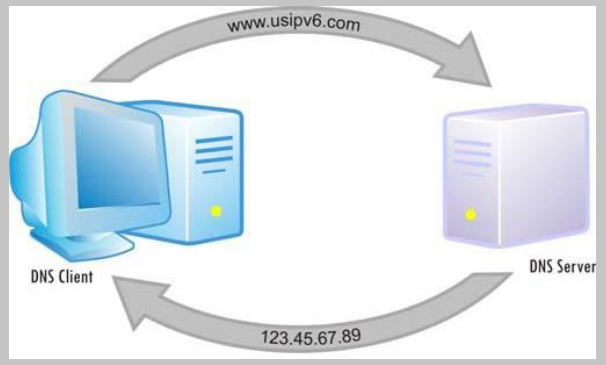
Services

- “PTR” Record
  - _ipp._tcp.local
- “SRV” Record
  - HP Color LaserJet 4700 [10080F]._ipp._tcp.local
  - HP Color LaserJet 4700 [96E411]._ipp._tcp.local
  - HP Color LaserJet 4700 [96E411]._ipp._tcp.local



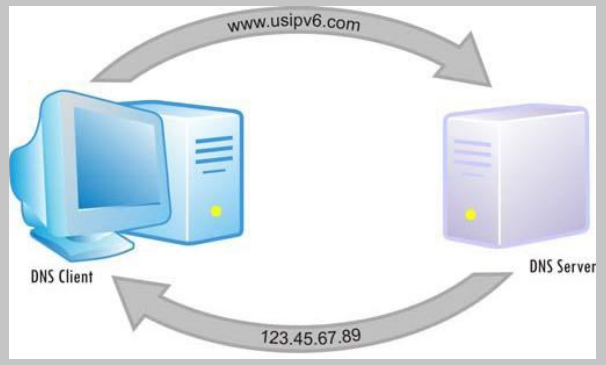
Other

- “TXT” Record
  - HP Color LaserJet 4700 [808EDF]._ipp._tcp.local
- “HINFO” Record
  - timur.local
  - localhost.local

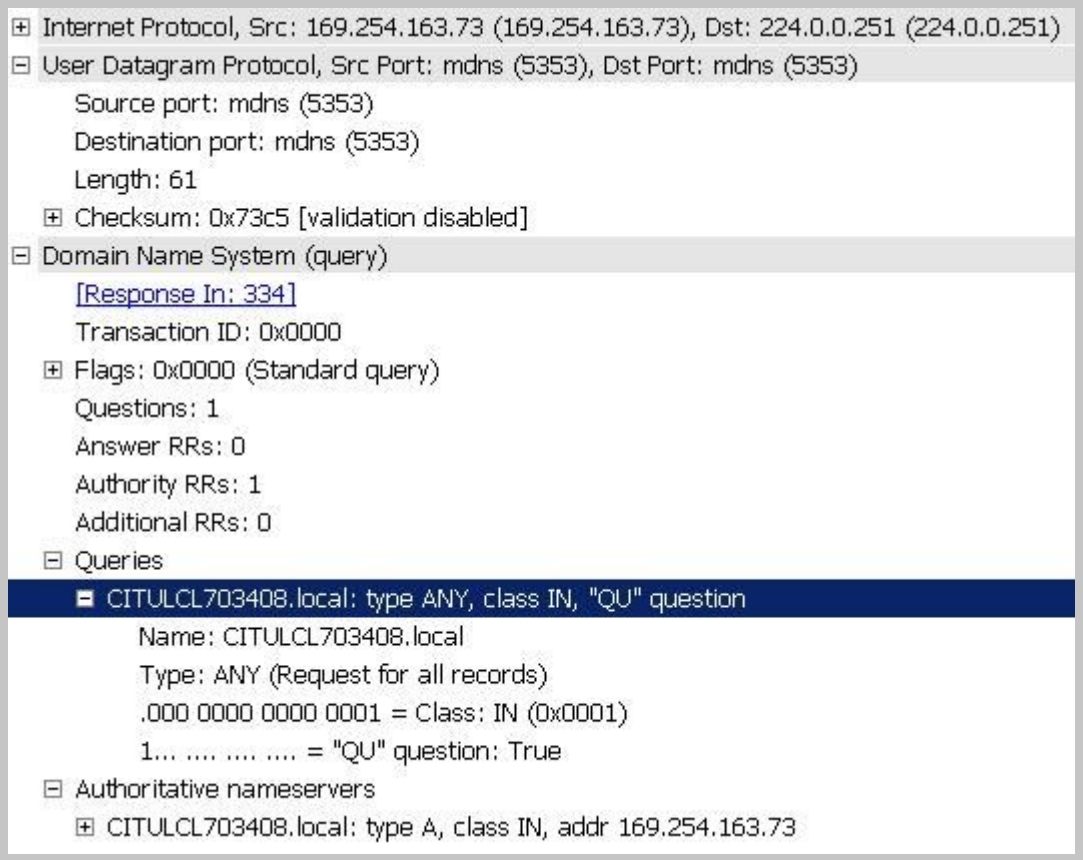


DNS-Service Discovery

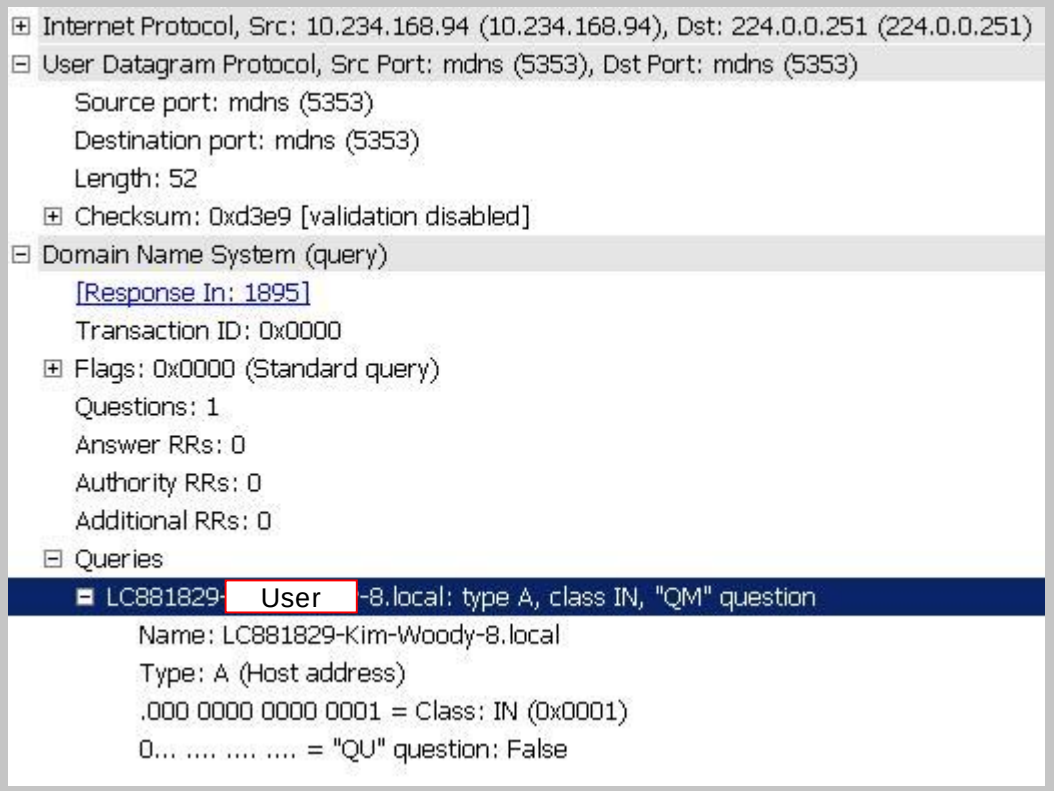
- Works over standard and multicast DNS
- Fully Compliant
- Continuous Querying
- Shared “PTR” records
- Unique “SRV” and “TXT” records



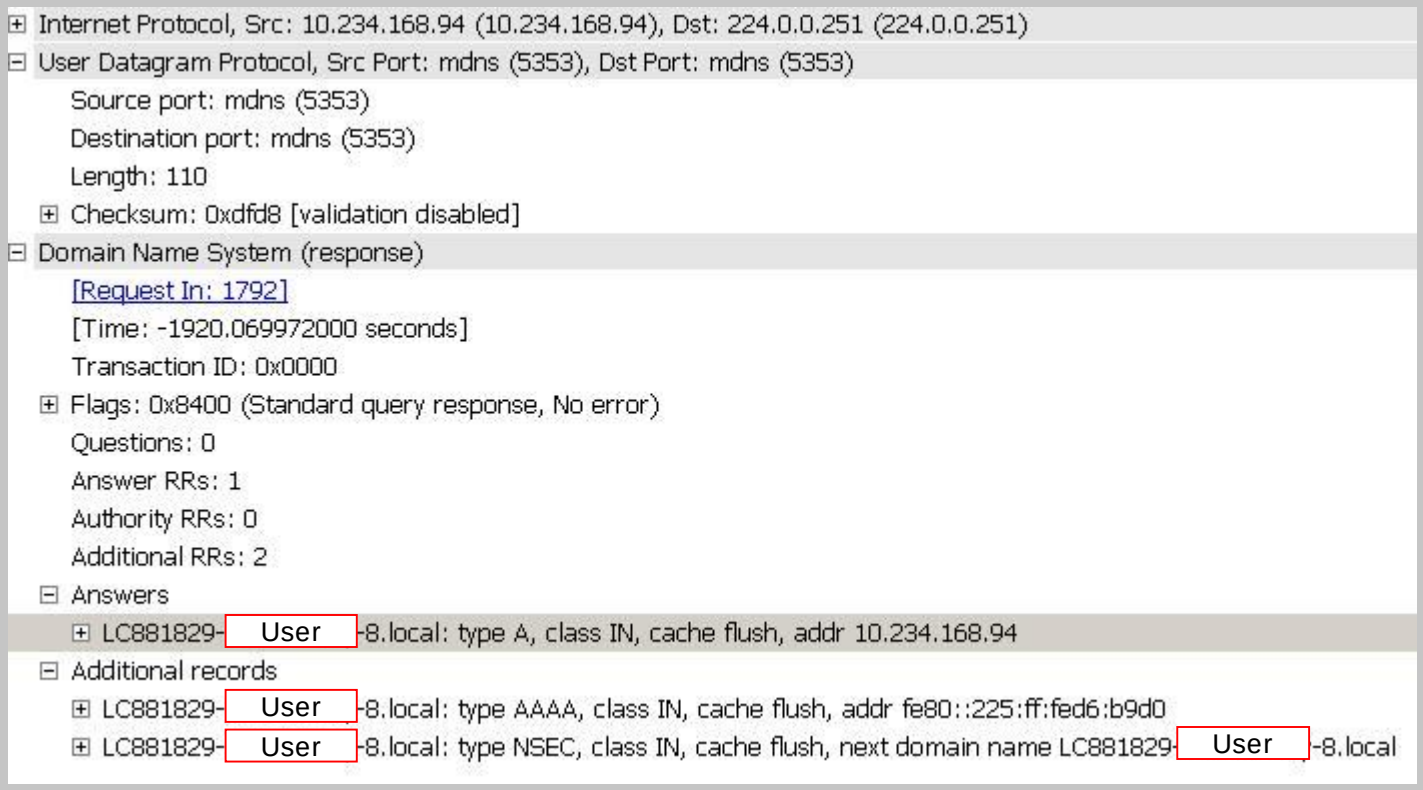
Probe


Internet Protocol, Src: 169.254.163.73 (169.254.163.73), Dst: 224.0.0.251 (224.0.0.251)
User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
Source port: mdns (5353)
Destination port: mdns (5353)
Length: 61
Checksum: 0x73c5 [validation disabled]
Domain Name System (query)
[Response In: 334]
Transaction ID: 0x0000
Flags: 0x0000 (Standard query)
Questions: 1
Answer RRs: 0
Authority RRs: 1
Additional RRs: 0
Queries
CITULCL703408.local: type ANY, class IN, "QU" question
Name: CITULCL703408.local
Type: ANY (Request for all records)
.000 0000 0000 0001 = Class: IN (0x0001)
1... .... = "QU" question: True
Authoritative nameservers
CITULCL703408.local: type A, class IN, addr 169.254.163.73

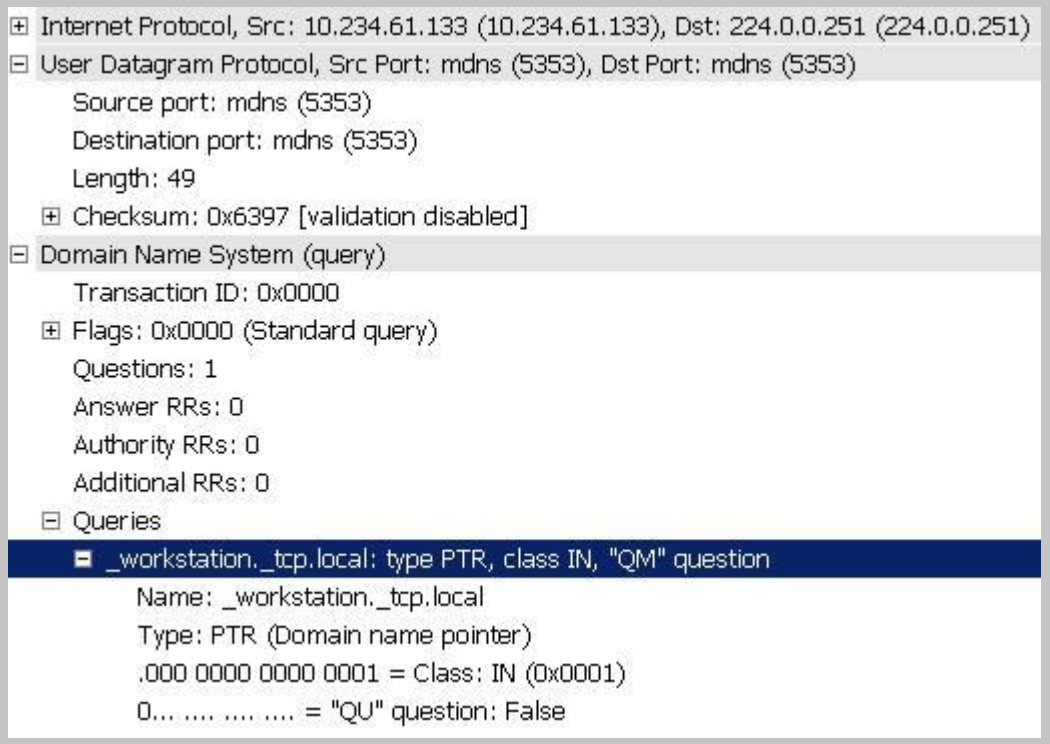
Query, "A" Record



Response, "A" Record



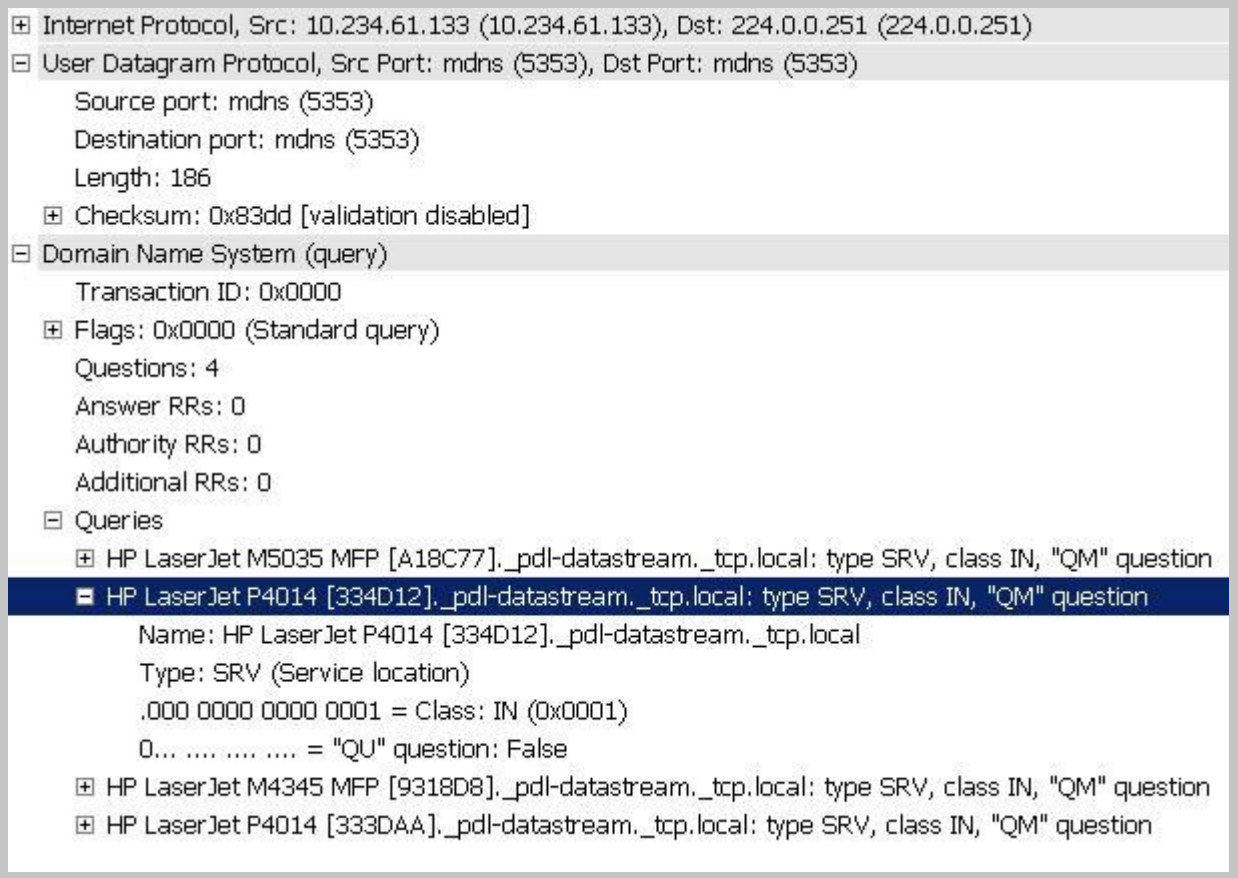
Query, "PTR" Record



Response, "PTR" Record

- ⊞ Internet Protocol, Src: 10.234.61.133 (10.234.61.133), Dst: 224.0.0.251 (224.0.0.251)
- ⊟ User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
  - Source port: mdns (5353)
  - Destination port: mdns (5353)
  - Length: 138
  - ⊞ Checksum: 0xd159 [validation disabled]
- ⊟ Domain Name System (response)
  - [Request In: 2627]
  - [Time: -300.088553000 seconds]
  - Transaction ID: 0x0000
  - ⊞ Flags: 0x8400 (Standard query response, No error)
  - Questions: 0
  - Answer RRs: 4
  - Authority RRs: 0
  - Additional RRs: 0
  - ⊟ Answers
    - ⊞ _workstation._tcp.local: type PTR, class IN, timur [00:1c:c4:ad:2b:1c]._workstation._tcp.local
    - ⊞ timur [00:1c:c4:ad:2b:1c]._workstation._tcp.local: type TXT, class IN, cache flush
    - ⊞ timur [00:1c:c4:ad:2b:1c]._workstation._tcp.local: type SRV, class IN, cache flush, priority 0, weight 0, port 9, target timur.local
    - ⊞ timur.local: type A, class IN, cache flush, addr 10.234.61.133

Query, "SRV" Record



Response, “SRV” Record

- ⊞ Internet Protocol, Src: 10.234.63.142 (10.234.63.142), Dst: 224.0.0.251 (224.0.0.251)
- ⊟ User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
  - Source port: mdns (5353)
  - Destination port: mdns (5353)
  - Length: 149
  - ⊞ Checksum: 0x36bf [validation disabled]
- ⊟ Domain Name System (response)
  - Transaction ID: 0x0000
  - ⊞ Flags: 0x8400 (Standard query response, No error)
  - Questions: 0
  - Answer RRs: 1
  - Authority RRs: 0
  - Additional RRs: 2
  - ⊟ Answers
    - ⊞ HP LaserJet P4014 [334D12]._pdl-datastream._tcp.local: type SRV, class IN, cache flush, priority 0, weight 0, port 9100, target P06LC882298.local
  - ⊟ Additional records
    - ⊞ P06LC882298.local: type A, class IN, cache flush, addr 10.234.63.142
    - ⊞ P06LC882298.local: type AAAA, class IN, cache flush, addr fe80::f6ce:46ff:fe33:4d12

Grabbing Information from an mDNS Responder

- mDNSHostName
  - Parameters (-t:Target)
  - Reverse lookup of the IPv4 address
  - Operates using a unicast legacy query to UDP port 5353 of the target
- mDNSLookup
  - Parameters [-t:Target] [-q:Question] [-r:Record Type]
  - Submits the question as given
  - Also operates using a unicast legacy query to UDP port 5353 of the target …

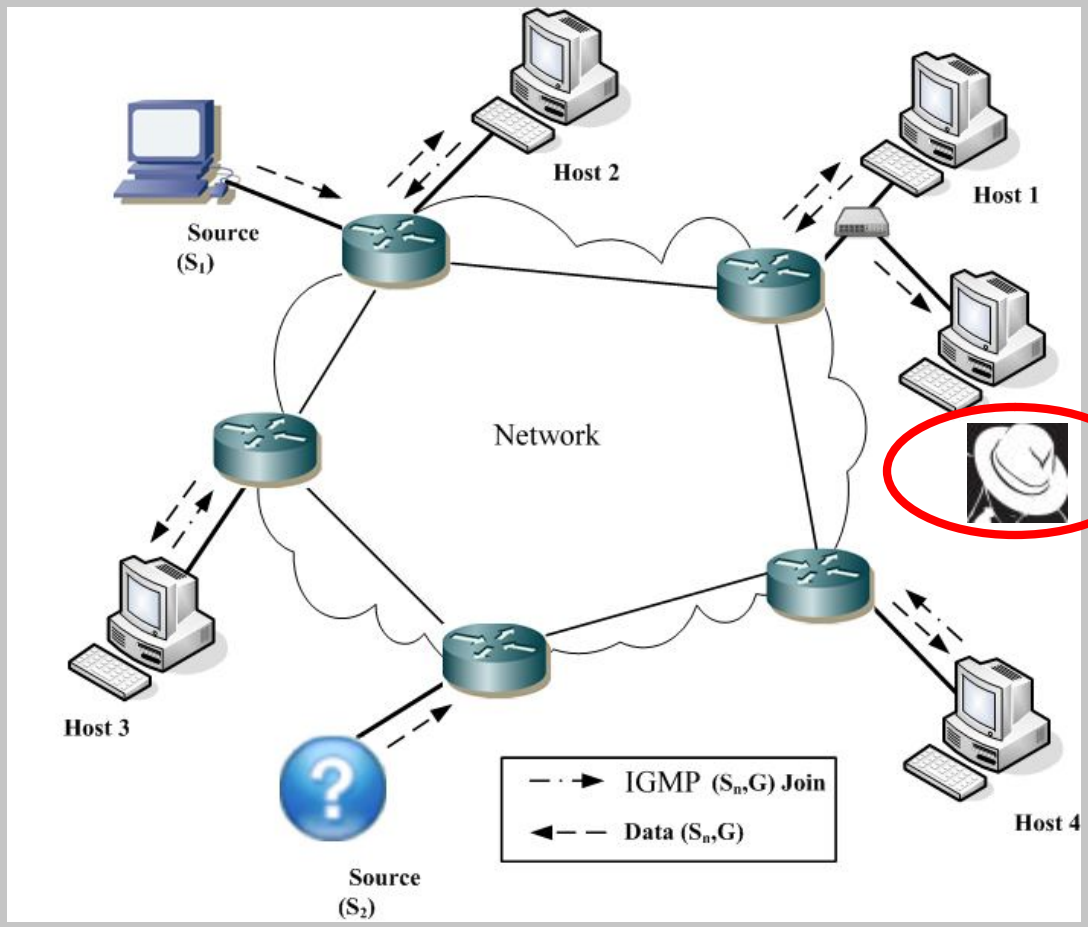
Demonstration

But wait …

Isn't this just flowing to my interface on it's own?

OK … I could do some really cool things with this!

What could I do?



Me!

Information Gathering

Host → Thank you!

Host → Thank you!

Service → Thank you!

Service → Thank you!

Service → Thank you!

Service → Thank you!

Host → Thank you!

Service → Thank you!

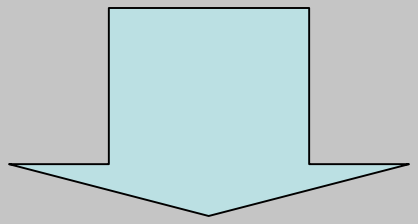
Service → Thank you!

Requirements

- Must have active responders (someone offering)
- Connected to same switch as other resolvers (someone asking)

Or

Join yourself (if you must) to the multicast group



- Works best on a busy network … because you need hosts out there asking a lot of questions so that you can collect the most answers!

First Cool thing … Host Discovery!

- mDNSDiscovery
  - Parameters [-t:Range]
  - Reports on any host communicating to 224.0.0.251
  - Doesn't join the group … only picks up traffic for the multicast group that is forwarded to all ports by the switch

Demonstration

End result?

Completely silent, passive host discovery

Network Security Guy!

Why don't you go active so I can catch you!

But wait, there's more …

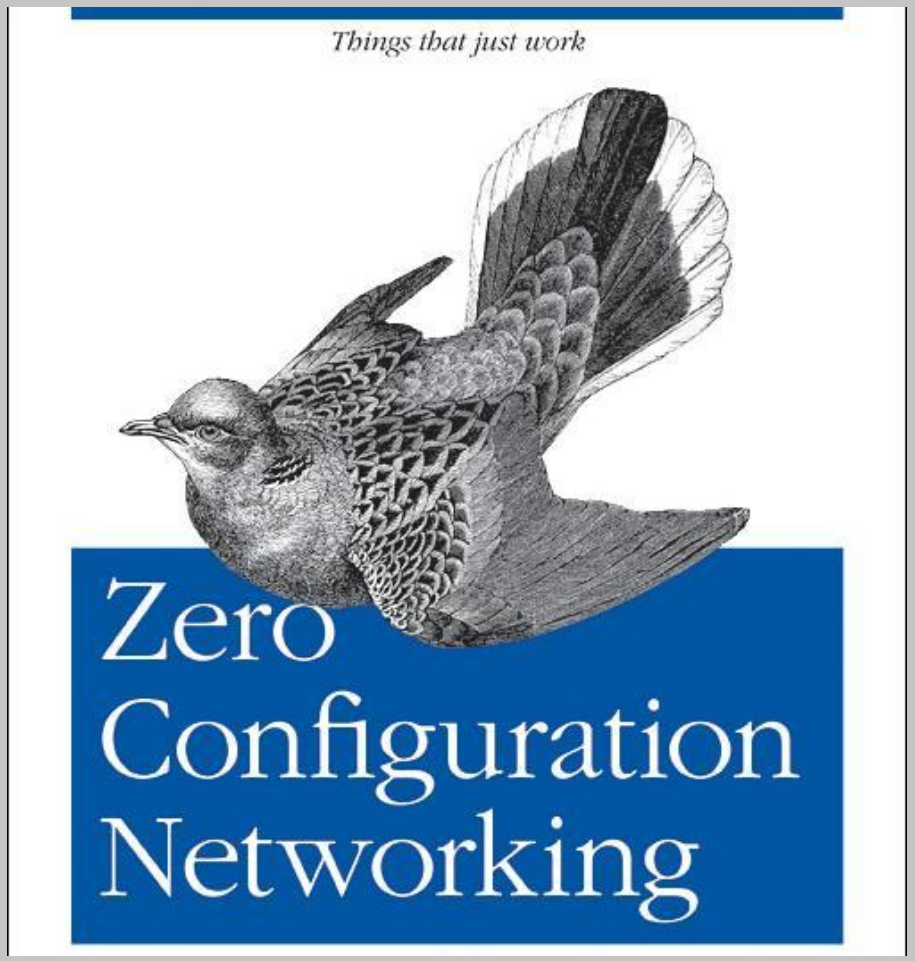
Second Cool thing … Port Scanning!

- Legitimate hosts performing (in essence) port scans with one packet
- Couldn't I perform a port scan with no packets?

That's right … two, two products in one!

Is it magic?

It's "Zero Configuration" Networking!



So Let's Do This …

- DNS-Service Discovery occurs continuously over the network
- Listen for it over multicast DNS on the local link
- Don't rely on known service records … it's too limiting
- When a host responds to a discovery request … report all the SRV record ports in it's replies as ports open on that host

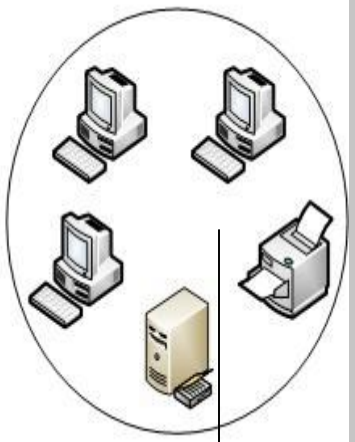
So Let's Do This …

- mDNSScan
  - Parameters [-t:Range] [-p:Ports]
  - Currently 22 services over 18 ports have been seen and identified using this method
  - Many more are possible based on the exhaustive list available
  - Doesn't join the group either …

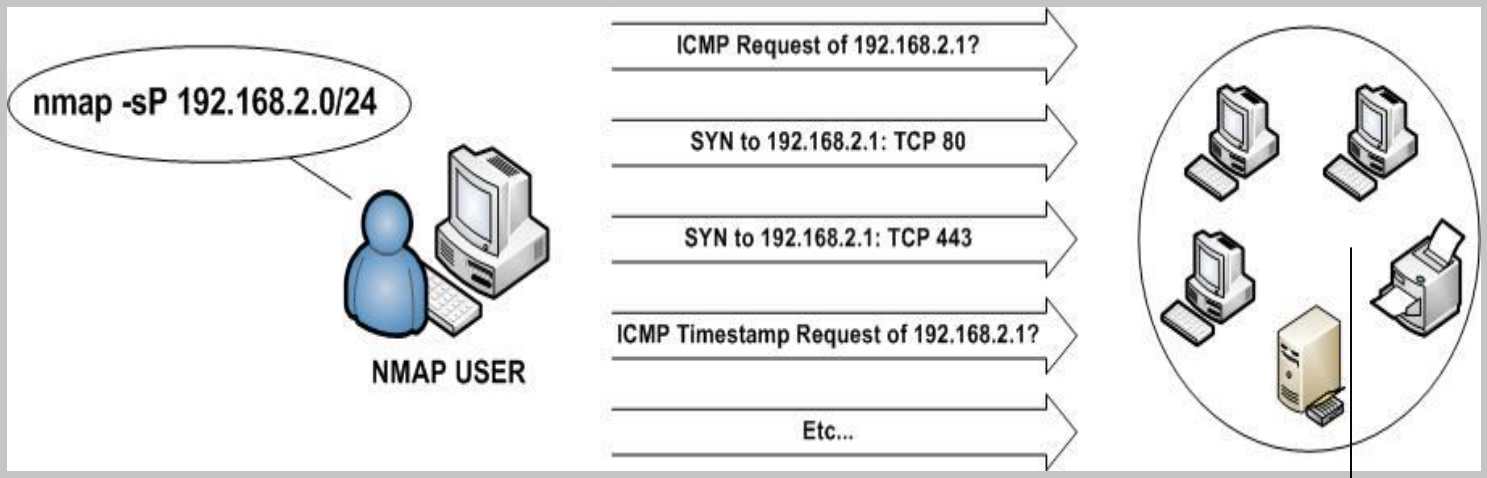
Demonstration

This is what our sensors see …

NMAP USER

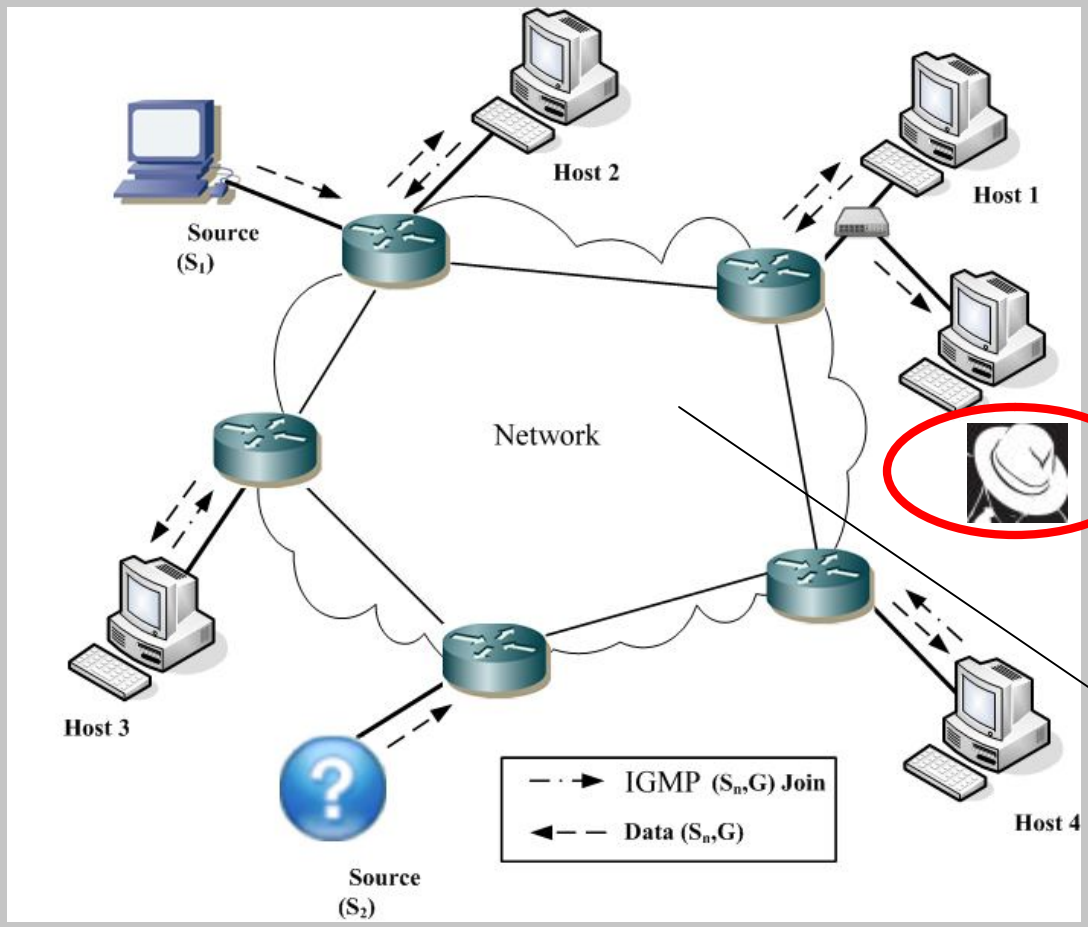


… in a typical active scan

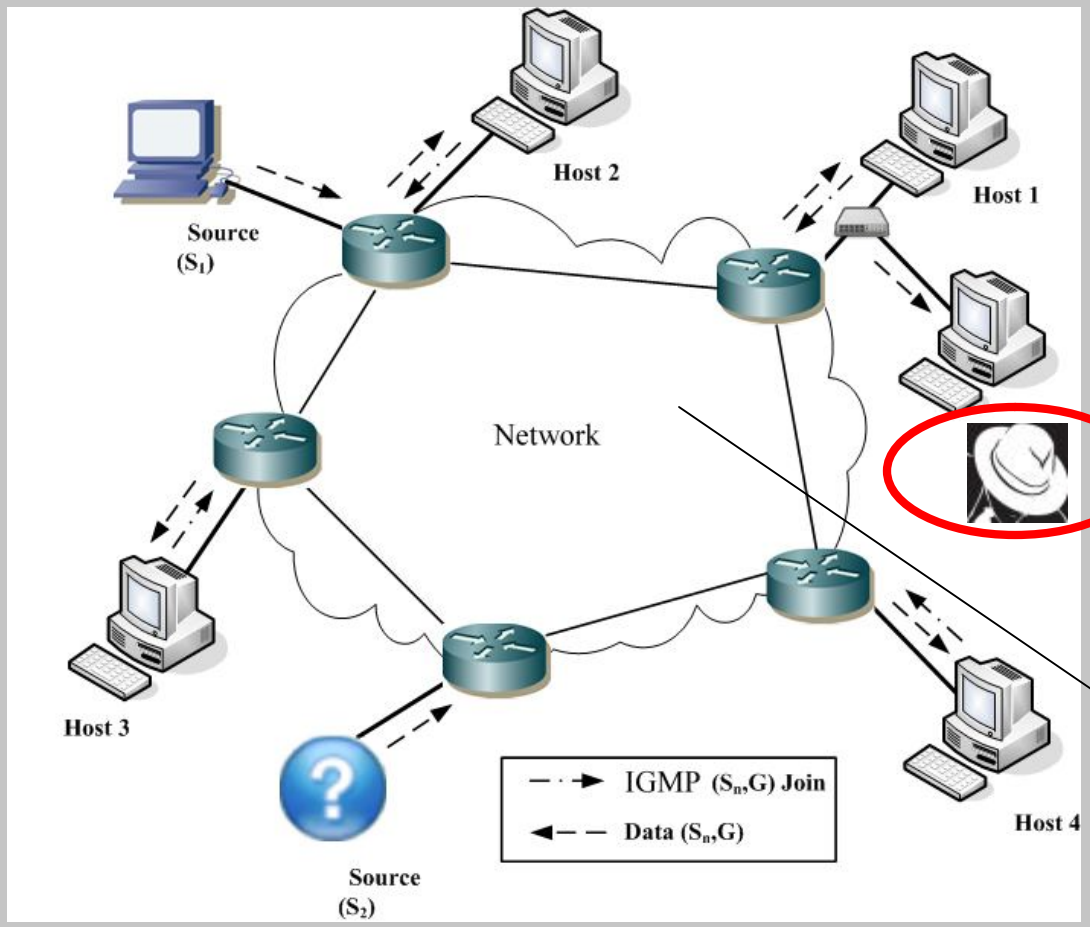


Time	Tag Name	Event Count
2011-05-26 15:03:05 CDT	TCP_Probe_Telnet	1
2011-05-26 15:03:05 CDT	TCP_Probe_Ftp	1
2011-05-26 15:03:06 CDT	TCP_Probe_POP3	1
2011-05-26 15:03:06 CDT	TCP_Port_Scan	1
2011-05-26 15:03:06 CDT	TCP_Probe_HTTP	1
2011-05-26 15:03:06 CDT	TCP_Probe_Finger	1
2011-05-26 15:03:06 CDT	TCP_Probe_LinuxConf	1
2011-05-26 15:03:06 CDT	TCP_Probe_DNS	1
2011-05-26 15:03:07 CDT	TCP_Probe_SunRPC	1
2011-05-26 15:03:07 CDT	TCP_Probe_POP3	1
2011-05-26 15:03:08 CDT	TCP_Port_Scan	1
2011-05-26 15:03:08 CDT	TCP_Probe_Imap4	1
2011-05-26 15:03:08 CDT	TCP_Probe_NNTP	1
2011-05-26 15:03:09 CDT	TCP_Probe_HTTP	1
2011-05-26 15:03:09 CDT	TCP_Probe_Rlogin	1

And what do our network sensors see …

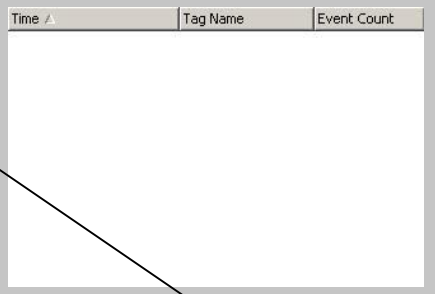


… during this passive scan



Nothing!

Me!



What does this mean?

Completely silent, passive port scans

Network Security Guy!

We are still unhappy!

OK, what else?

- Unique Implementations
- Unique Records
- Unique Sets
- Could this be used to fingerprint?

Yes … yes, it could

- Linux
  - _services._dns-sd._udp.local → Avahi
  - _workstation._tcp.local (SRV)→ Linux
- Apple
  - _services._dns-sd._udp.local → Bonjour
  - _afpovertcp._tcp.local (SRV,TXT) → Apple
  - _device-info._tcp.local (TXT)

Yes … yes, it could

- Printers
  - _ipp._tcp.local (SRV, TXT) → Printer
  - _printer._tcp.local (SRV, TXT) → Printer
  - _pdl-datastream._tcp.local (SRV, TXT) → Printer
- Network Attached Storage (Seagate)
  - _blackarmor4dinfo._udp.local (SRV,TXT) → NAS, Seagate
  - _blackarmor4dconfig._tcp.local (SRV, TXT)
- IP Cameras (Axis)
  - _axis-video._tcp.local (SRV) → IP Camera, Axis

Profiling, "TXT" Records

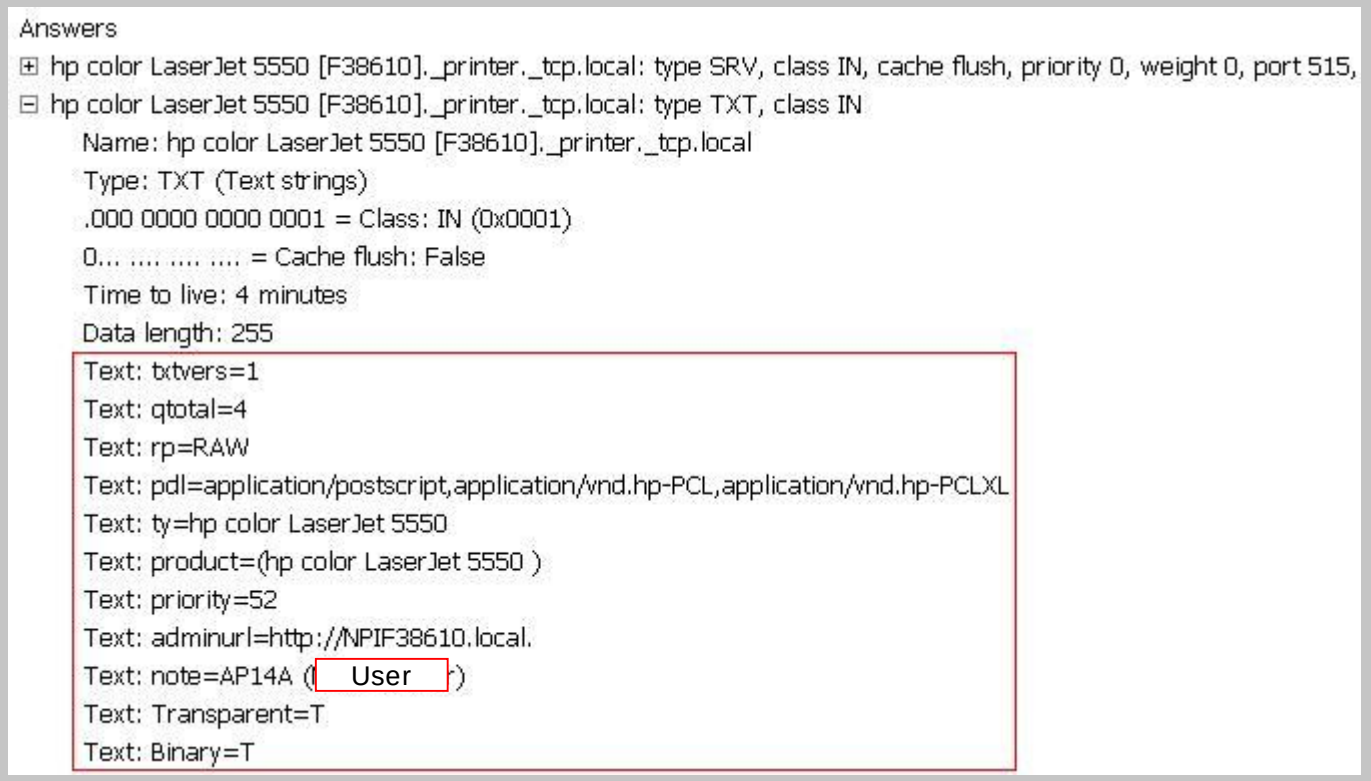
```
Answers
⊟ localhost [00:00:00:00:00:00]._workstation._tcp.local: type TXT, class IN, cache flush
      Name: localhost [00:00:00:00:00:00]._workstation._tcp.local
      Type: TXT (Text strings)
      .000 0000 0000 0001 = Class: IN (0x0001)
      1... .... .... .... = Cache flush: True
      Time to live: 1 hour, 15 minutes
      Data length: 40
      Text: org.freedesktop.Avahi.cookie=2802281168
```

Linux

```
Answers
⊞ _afpovertcp._tcp.local: type PTR, class IN, CITULCL881919._afpovertcp._tcp.local
⊟ CITULCL881919._device-info._tcp.local: type TXT, class IN
      Name: CITULCL881919._device-info._tcp.local
      Type: TXT (Text strings)
      .000 0000 0000 0001 = Class: IN (0x0001)
      0... .... .... .... = Cache flush: False
      Time to live: 1 hour, 15 minutes
      Data length: 20
      Text: model=MacBookPro6,2
⊞ _rfb._tcp.local: type PTR, class IN, CITULCL881919._rfb._tcp.local
Additional records
⊞ CITULCL881919._afpovertcp._tcp.local: type TXT, class IN, cache flush
⊞ CITULCL881919._afpovertcp._tcp.local: type SRV, class IN, cache flush, priority 0,
```

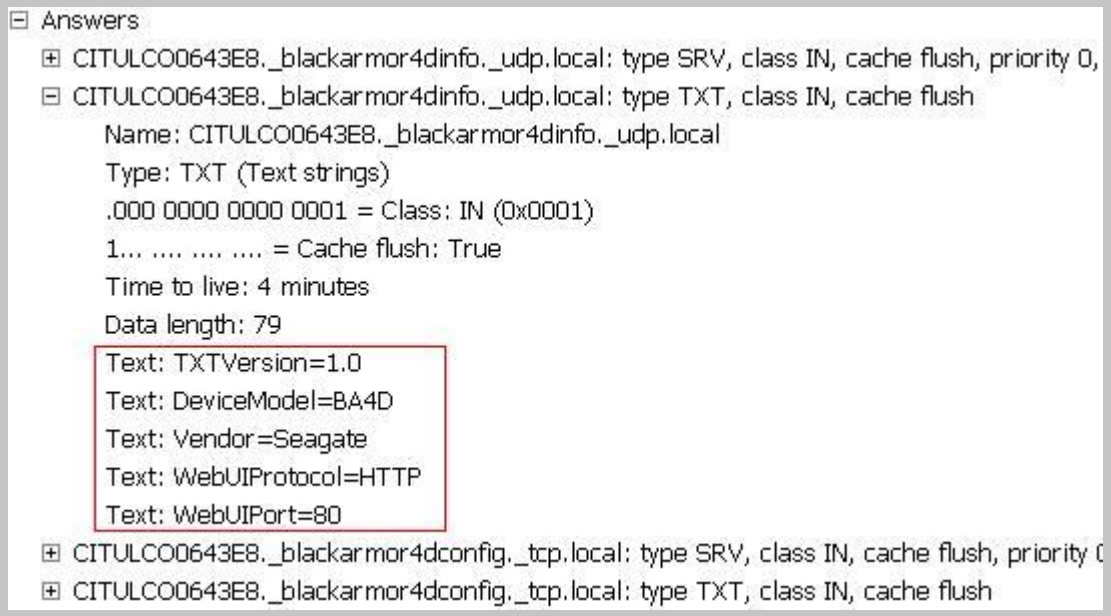
Apple

Profiling, "TXT" Records



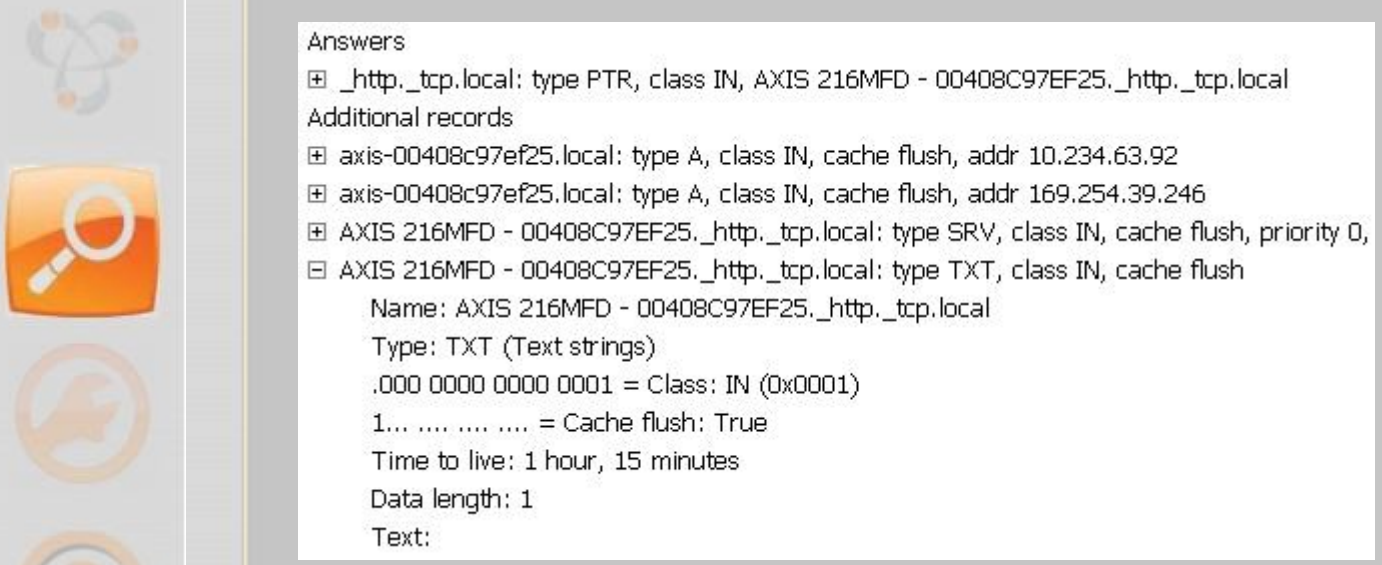
Printer

Profiling, “TXT” Records



Network Attached Storage (Seagate)

Profiling, “TXT” Records



IP Camera (Axis)

Someday … mDNSFingerprint

- Build database of identifying record sets
- Collect all incoming records and organize by host
- Match against database and extract configuration information
- Return identity and configuration information for each host

Limitations

- Multicast
  - Routers between the recipient and the source must be multicast enabled
- mDNS
  - Querying (Link-Local Response Only)
    - Responses only accepted from local-link
    - Responses only sent to the local-link
  - Listening (Layer-2 Boundaries)
    - Broadcast Domain
    - VLAN containment

Sensors

- Intrusion Detection/Prevention Systems
- Etherape
- Netflow/StealthWatch

Detect

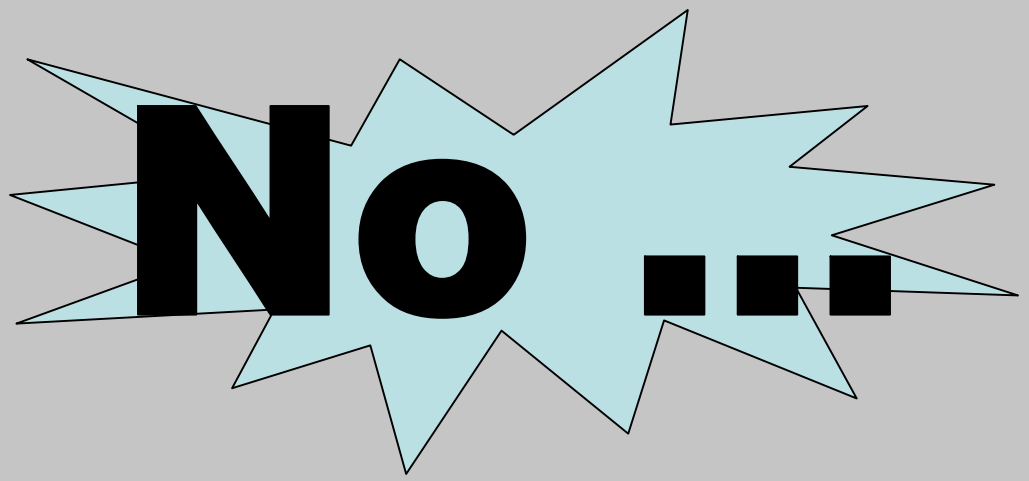
Other detection possibilities

- Monitoring
  - IGMP (group membership)
  - mDNS (responders)
- Management Applications?

Defenses (Host)

- Anti-Virus/Anti-Spyware/Anti-Spam
- Intrusion Prevention System
- Firewall and Port Blocking
- Application Control
- Device Control
- Others

Do these help any?



Defenses (Network)

- Firewalls/Access Control Lists
- Network Access Control
- VLANs

How about these?

Not really ...

What can we do then?

IGMP

- Implement IGMP snooping
- Authenticate group membership (IGAP)
- Track members (Membership reports)

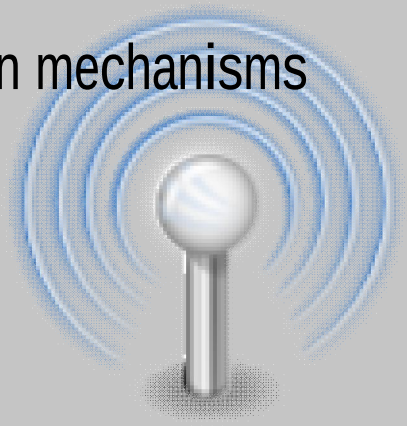
What can we do then?

Multicast DNS

- Locate mDNS responders
- Disable the service
- Harden the box … in particular the services that are offered
- Sanitize records

Plan of Attack

- Hunt down mDNS responders with these tools
- Remove them or harden them
- Implement any controls you have for multicast in your environment
  - IGMP snooping/MLDv2
  - IGAP or IPv6 multicast authentication mechanisms

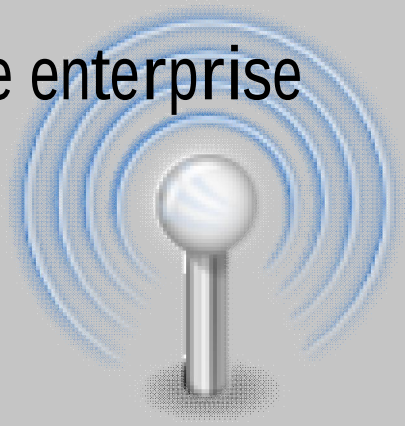


Other Protocols

- Simple Service Discovery Protocol (SSDP)
  - Microsoft's Answer to "Zero Configuration" networking
  - HTTP-Based but also multicasted
  - Methods: NOTIFY, M-SEARCH
- Link Local Multicast Name Resolution (LLMNR)
  - Another Microsoft solution
  - DNS-Based but also multicasted
- Both less developed, but still in use

Final Thoughts

- Hosts are now actively advertising their available attack surfaces to anyone listening on the network
- Great for passive information gathering
- Can be controlled to limit your exposure
- But ultimately …This is not for the enterprise



Demonstration

Tools

- mDNSHostName v1.00 for Windows
  MD5: e97b2c8325a0ba3459c9a3a1d67a6306
- mDNSLookup v1.00 for Windows
  MD5: f489dd2a9af1606dd66a4a6f1f77c892
- mDNSDiscovery v1.00 for Windows
  MD5: e6c8c069989ec0f872da088edbbb1074
- mDNSScan v1.00 for Windows
  MD5: eb764b7f0ece697bd8abbea6275786dc

Updates → http://mdnstools.sourceforge.net/

Links

- http://www.multicastdns.org/
- http://www.dns-sd.org/
- http://www.ietf.org/id/draft-cheshire-dnsext-multicastdns-14.txt
- http://www.ietf.org/id/draft-cheshire-dnsext-dns-sd-10.txt
- http://www.ietf.org/id/draft-cheshire-dnsext-special-names-01.txt
- http://www.rfc-editor.org/rfc/rfc3927.txt
- http://www.bleepsoft.com/tyler/index.php?itemid=105
- http://www.dns-sd.org/ServiceTypes.html
- http://www.zeroconf.org/
- http://avahi.org/
- http://meetings.ripe.net/ripe-55/presentations/strotmann-mdns.pdf
- http://www.mitre.org/work/tech_papers/2010/09_5245/09_5245.pdf


Q&A