AKAMAI THREAT ADVISORY

Threat Advisory: mDNS Reflection DDoS

Risk Factor: Medium
TLP: Green

Author: Wilber Mejia

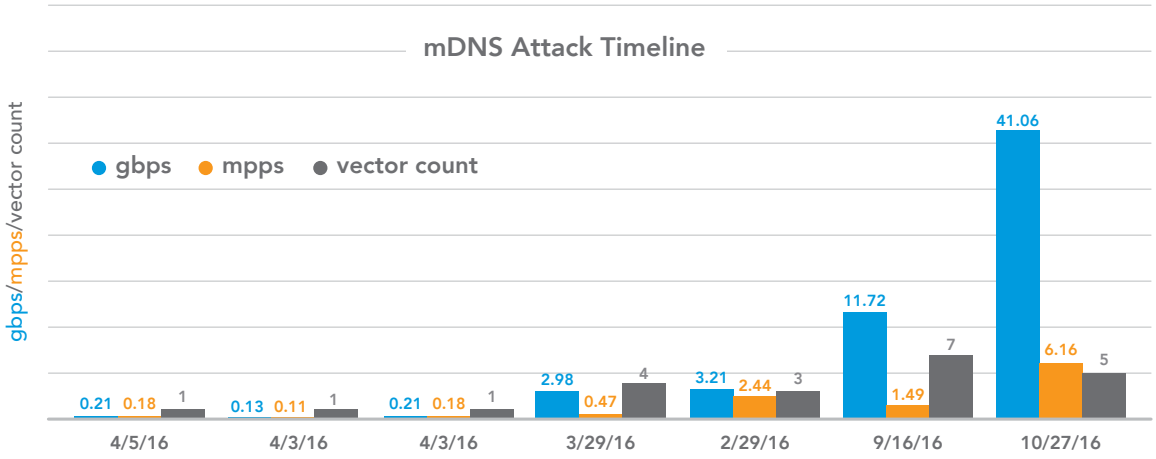
Akamai
FASTER FORWARD

1.0 / Overview / Toward the end of Q3 2015, Akamai SIRT began observing limited use of DDoS attacks fueled by Multicast Domain Name System (mDNS) capable devices. The potential for mDNS to become a vector for use in reflection and amplification DDoS attacks was also disclosed on **March, 2015**.

This advisory details the concept and techniques of the mDNS reflection attack vector as well as how to mitigate it. This attack vector is possible through the availability of source devices that expose mDNS, which is expected on port 5353, over the Internet.

As of October 2016, Akamai has detected and successfully mitigated seven mDNS DDoS attacks against targets in the Gaming and Software & Technology industry verticals. So far, the longevity of this attack may be in question, based on characteristics observed in real-world attacks.

2.0 / Attack Timeline / The mDNS attack vector has seen sporadic use since it was first observed in September 2015. As indicated in the timeline below, there was roughly a five-month gap between the initial attack and the next one; this was possibly the result of initial testing of attack scripts. It wasn't until the end of March, early April 2016 that the attack vector began to see more consistent use. However, the attacks using mDNS as a single vector have not been as powerful as other reflection vectors to date.



2.1 / Highlighted Attack Attributes / In the following section, we highlight one attack that occurred in 2016. Of the seven mitigated attacks thus far, none have been against the same target.

- Peak bandwidth: 2.9 Gigabits per second (Gbps)
- Peak packets per second: 465.2 Thousand Packets per second
- Attack Vector: mDNS
- Source port: 5353 (mDNS)
- Destination port: Random

```
17:07:32.071836 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)

17:07:32.071857 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)

17:07:32.071873 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 4/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local.,
PTR _http._tcp.local., PTR _rfb._tcp.local. (143)

17:07:32.072187 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 3/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local.,
PTR _ipp._tcp.local. (123)

17:07:32.072274 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 3/0/0
PTR _workstation._tcp.local., PTR _https._tcp.local., PTR _http._tcp.local. (119)

17:07:32.072279 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 4/0/0
PTR _pdl-datastream._tcp.local., PTR _printer._tcp.local.,
PTR _ipp._tcp.local., PTR _http._tcp.local. (143)

17:07:32.072295 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 8/0/0
PTR _workstation._tcp.local., PTR _webdavs._tcp.local.,
PTR _webdav._tcp.local., PTR _smb._tcp.local., PTR _sftp._tcp.local.,
PTR _http._tcp.local., PTR _afpovertcp._tcp.local.,
PTR _device-info._tcp.local. (235)

17:07:32.072361 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 7/0/0
PTR _workstation._tcp.local., PTR _ftp._tcp.local., PTR _edcp._udp.local.,
PTR _afpovertcp._tcp.local., PTR _device-info._tcp.local.,
PTR _smb._tcp.local., PTR _http._tcp.local. (209)
```

The last noted mDNS attack event occurred on October 27, 2016. This was a multi vector DDoS attack consisting of a SYN Flood, UDP Flood, UDP Fragment, DNS Flood, and mDNS Flood peaking at 41 Gbps.

- Event Time Start: Oct 27, 2016 06:01:00 UTC
- Peak bandwidth: 41 Gbps
- Peak packets per second: 6 Million Packets per second
- Attack Vector: SYN Flood, UDP Flood, UDP Fragment, DNS Flood, mDNS Flood
- Source port: Random
- Destination port: Random

mDNS Flood

```
06:05:36.522973 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)

06:05:36.522976 IP Z.Z.Z.Z.53301 > X.X.X.X.80: Flags [R],
seq 194989615, win 0, length 0

06:05:36.523036 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 4/0/0
PTR _workstation._tcp.local., PTR _http._tcp.local.,
PTR _device-info._tcp.local., PTR _smb._tcp.local. (144)

06:05:36.523040 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _ssh._tcp.local. (97)

06:05:36.523102 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 3/0/0
PTR _workstation._tcp.local., PTR _https._tcp.local., PTR _http._tcp.local. (119)

06:05:36.523114 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 3/0/0
PTR _workstation._tcp.local., PTR _ssh._tcp.local., PTR _sftp-ssh._tcp.local. (121)

06:05:36.523147 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 4/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local.,
PTR _ssh._tcp.local., PTR _sftp-ssh._tcp.local. (147)

06:05:36.523168 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 3/0/0
PTR _workstation._tcp.local., PTR _https._tcp.local., PTR _http._tcp.local. (119)

06:05:36.523221 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _ssh._tcp.local. (97)

06:05:36.523285 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 3/0/0
PTR _workstation._tcp.local., PTR _ssh._tcp.local., PTR _sftp-ssh._tcp.local. (121)

06:05:36.523313 IP Z.Z.Z.Z.5353 > X.X.X.X.80: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _sftp-ssh._tcp.local. (102)

06:05:36.523322 IP Z.Z.Z.Z.58727 > X.X.X.X.80: Flags [R.],
seq 2670641259, ack 2670641259, win 1400, length 0
```

SYN Flood

```
6:01:35.894589 IP Z.Z.Z.Z.28970 > X.X.X.X.80: Flags [SEW],
seq 4054777856, win 0, length 0

06:01:35.894639 IP Z.Z.Z.Z.1028 > X.X.X.X.80: Flags [SEW],
seq 1405550592, win 0, length 0

06:01:35.894646 IP Z.Z.Z.Z.50021 > X.X.X.X.80: Flags [SEW],
seq 3309240320, win 0, length 0

06:01:35.894655 IP Z.Z.Z.Z.40602 > X.X.X.X.80: Flags [SEW],
seq 2493120512, win 0, length 0

06:01:35.894658 IP Z.Z.Z.Z.3847 > X.X.X.X.80: Flags [SEW],
seq 1046675456, win 0, length 0

06:01:35.894658 IP Z.Z.Z.Z.21163 > X.X.X.X.80: Flags [SEW],
seq 2160787456, win 0, length 0
```

UDP Flood

```
06:08:36.313095 IP Z.Z.Z.Z.42862 > X.X.X.X.80: UDP, length 1

06:08:36.313768 IP Z.Z.Z.Z.648 > X.X.X.X.80: UDP, length 11

06:08:36.314002 IP Z.Z.Z.Z.60399 > X.X.X.X.80: UDP, length 6

06:08:36.322169 IP Z.Z.Z.Z.19246 > X.X.X.X.80: UDP, length 9

06:08:36.327485 IP Z.Z.Z.Z.22952 > X.X.X.X.80: UDP, length 11

06:08:36.327764 IP Z.Z.Z.Z.55095 > X.X.X.X.80: UDP, length 9
```

UDP fragment

```
06:06:43.088487 IP Z.Z.Z.Z > X.X.X.X: udp

06:06:43.088489 IP Z.Z.Z.Z > X.X.X.X: udp

06:06:43.088501 IP Z.Z.Z.Z > X.X.X.X: udp

06:06:43.088504 IP Z.Z.Z.Z > X.X.X.X: udp

06:06:43.088507 IP Z.Z.Z.Z > X.X.X.X: udp
```

DNS Reflection

```
06:16:34.508072 IP Z.Z.Z.Z.53 > X.X.X.X.23130: 28397| 20/0/1 MX stagg.cpsc.gov. 5,
MX hormel.cpsc.gov. 5, TXT “v=spf1 ip4:x.x.x.x ip4:x.x.x.x ip4:X.X.X.X mx a:list.
cpsc.gov -all”, A x.x.x.x, AAAA 2600:803:240::2, DNSKEY, DNSKEY, DNSKEY, DNSKEY,
Type51, RRSIG[|domain]

06:16:34.508077 IP Z.Z.Z.Z.53 > X.X.X.X.23130: 28397| 20/0/1 MX hormel.cpsc.gov. 5,
MX stagg.cpsc.gov. 5, TXT “v=spf1 ip4:X.X.X.X ip4:X.X.X.X ip4:X.X.X.X mx a:list.
cpsc.gov -all”, A x.x.x.x, AAAA 2600:803:240::2, DNSKEY, DNSKEY, DNSKEY, DNSKEY,
Type51, RRSIG[|domain]

06:16:34.508409 IP Z.Z.Z.Z.53 > X.X.X.X.10157: 32564 14/2/0 MX stagg.cpsc.gov. 5, MX
hormel.cpsc.gov. 5, DNSKEY, DNSKEY, DNSKEY, DNSKEY, RRSIG,[|domain]
```

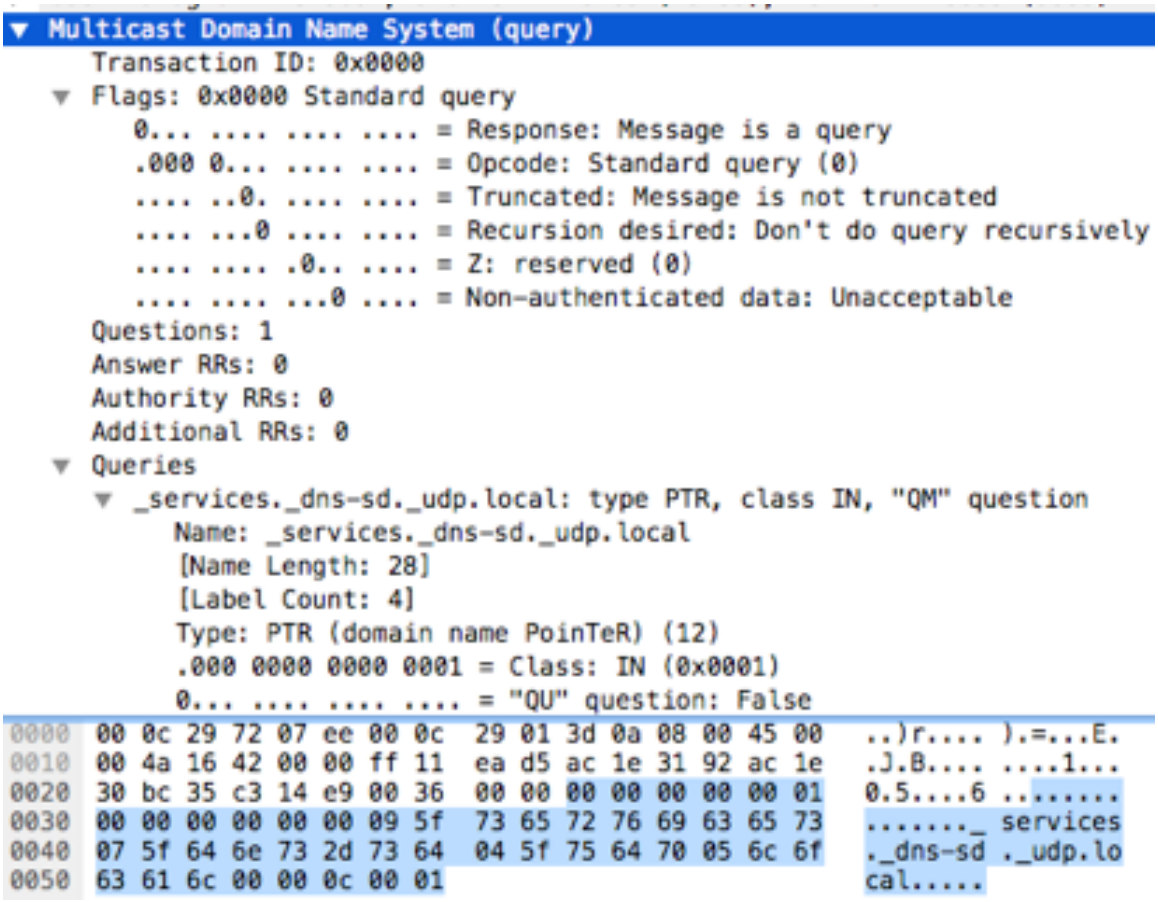
2.2/ Attack & mDNS Overview / Multicast DNS(mDNS) is a proposed standard protocol released in 2013 as RFC6762. It facilitates the discovery of devices and services, ideally in small networks, without the need for any or minimal user interaction. Because of this, mDNS can also be a suitable component for Zero Configuration Networking (zeroconf). mDNS shares much of the same structure as regular DNS packets, a likely reason for its quick adoption as a DDoS attack vector.

Of course, with the simplicity of a protocol designed to allow a device to be plugged in and ready to go comes some risk. A vulnerability (VU#550620) on mDNS was found by Chad Seaman where mDNS would allow responses to queries originating from outside the local network. These responses then would allow it to disclose information about the affected device, such as its software and services, as well as other potentially sensitive information, suchlike hostname, internal network configuration settings, model number, etc. This feedback can allow a malicious actor to use any mDNS hosts that reply to unicast queries over their Internet-connected interface to participate in Distributed Denial of Service (DDoS) Reflection / Amplification attacks. More information regarding the vulnerability can be found **here**.

Using a similar concept, the mDNS attack script created by malicious actors sends a specialized unicast query for eliciting a response from mDNS devices. Defined in RFC6763, the service enumeration query is useful for returning all advertised service types on a network. The attack script will generate a payload query of 46 bytes, as shown in the next figure, sending a DNS query for "_services._dns-sd._udp.local" to the vulnerable host; this is meant to return all known services back to the requesting devices. These advertised services can also be queried individually with varying response sizes, offering more opportunity for attackers. However, this tactic would require a more sophisticated attack tool to easily leverage these additional response payloads in a DDoS attack.

ATTACK SCRIPT UDP PAYLOAD DATA STRUCTURE:

```
0000000: 0000 0000 0001 0000 0000 0000 095f 7365  ............._se
0000010: 7276 6963 6573 075f 646e 732d 7364 045f  rvices._dns-sd._
0000020: 7564 7005 6c6f 6361 6c00 000c 0001       udp.local.....
```



The mDNS response payloads observed in DDoS attacks so far have been limited in size. The largest contained 428 bytes of data, 12 bytes fewer than a single NTP monlist data response packet. However, the typical mDNS response size observed has been around the 100-200 byte range. This means the vector is more commonly maxing out at about 4.35x amplification, when considering just the 46 byte request payload and the 200 byte payload response.

2.3 / Sample Attack Script Usage And Signatures / The attack script created for mDNS is a modified version of the many scripts available now for UDP reflection and amplification attacks. Its usage is similar as well to the other scripts: simply provide a target IP, target port, list of mDNS devices on the Internet, threads, packet throttle rate, and finally attack run time. The script will then impersonate the target IP when sending the malicious 46 byte queries observed in the next figure through tcpdump.

```
18:37:12.565541 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)

18:37:12.565908 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
```

```
18:37:12.566332 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
18:37:12.566726 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
18:37:12.567137 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
18:37:12.567507 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
18:37:12.567895 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
18:37:12.568274 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
18:37:12.568672 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
18:37:12.569069 IP Z.Z.Z.Z.13763 > X.X.X.X.5353: 0
PTR (QM)? _services._dns-sd._udp.local. (46)
```

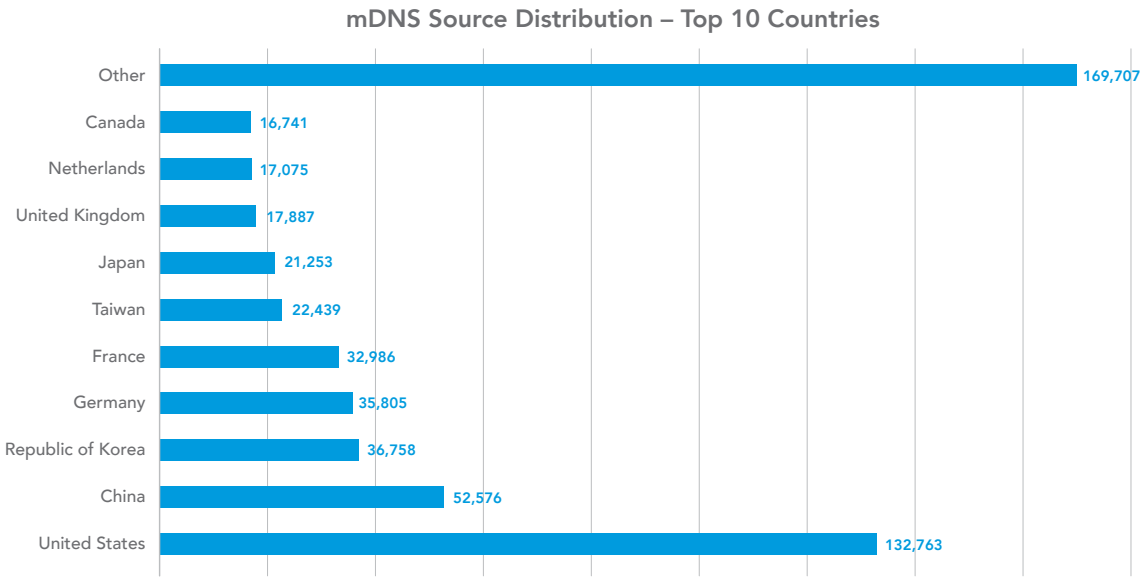
Sample of attack tool request

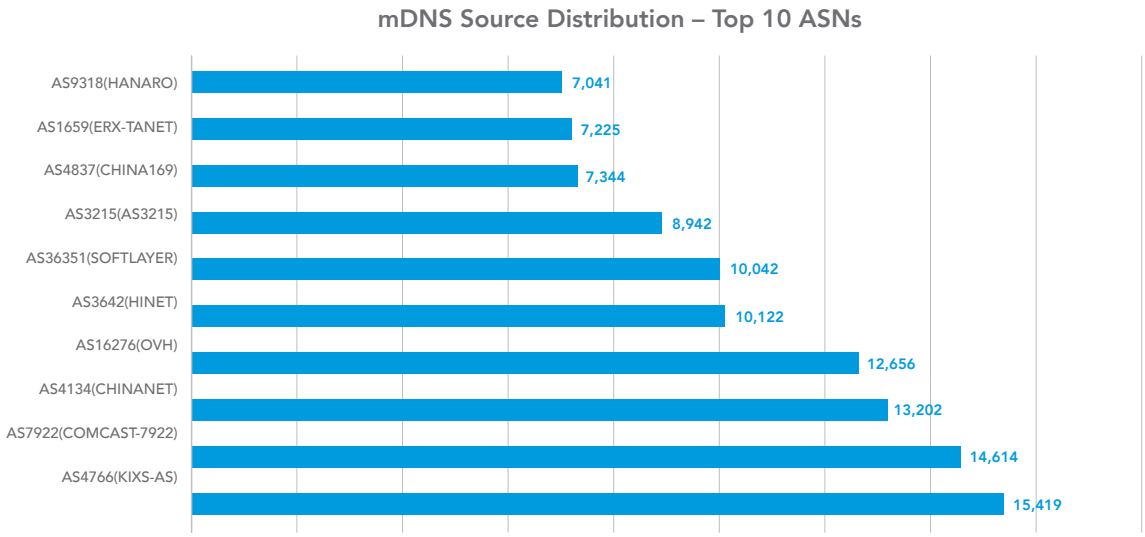
In our lab test, the queries were sent to a Linux server setup with an mDNS listener. The basic response can be seen in the next figure.

```
18:38:10.252994 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.253256 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.253694 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.254043 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.254394 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.254739 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.255098 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.255458 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.255823 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
18:38:10.256193 IP Z.Z.Z.Z.5353 > X.X.X.X.47070: 0*- 2/0/0
PTR _workstation._tcp.local., PTR _udisks-ssh._tcp.local. (104)
```

Sample of attack tool response

On November 4, 2016, The Shadowserver Foundation completed a mDNS (5353/UDP) scan where 526,245 unique IPs responded to mDNS query. The following graph below represents the top 20 countries and ASNs with mDNS accessible according to the latest scan.





Top ten source countries that participated on mDNS reflection against Akamai's customers. The majority of the attacking IP addresses originated from USA and China.

3.0 / Mitigation / This is one of those protocols that is designed for use within a local network. As such, there should be no reason for exposure of mDNS devices over the Internet. If required, a good practice would be to filter any incoming queries and allow known sources only. An IDS such as Snort can further be used to detect these queries and further mitigate the use of your devices in DDoS attacks. A sample snort detection rule is below.

```
alert udp $EXTERNAL_NET !5353 -> $HOME_NET 5353 \
(msg: "mDNS DDoS Abuse request"; \
flow: to_server; \
content: "|00 00 00 00 00 01 00 00 00 00 00 00 09 5f 73 65 72 76 69 63 65 73 07 5f
64 6e 73 2d 73 64 04 5f 75 64 70 05 6c 6f 63 61 6c 00 00 0c 00 01|"; dsize:46<>46; \
classtype:Reflection-Abuse; \
sid: 201600004; rev:1;)
```

4.0 / Conclusion / The mDNS attack vector has seen limited use to date. Like SSDP, this vector would most likely be fueled by devices within a home network. Early attacks have been underpowered but scanning of mDNS port 5353 has been observed on a daily basis over our mitigation platform. Once more devices are enumerated, there is the possibility of larger attacks. Also, like SSDP, based on the unnecessary exposure of this protocol, it is expected that mDNS may not thrive as a DDoS attack vector, as ISPs introduce filtering of this port to home users. Unfortunately, even after years of use, SSDP continues to be leveraged on a daily basis and is still capable of substantial DDoS attacks. This means mDNS attacks may become more powerful before any significant proactive filtering is applied.

References

- http://www.ietf.org/rfc/rfc6762.txt
- https://tools.ietf.org/html/rfc6762
- https://github.com/chadillac/mdns_recon
- https://mdns.shadowserver.org/