DNS spoofing for redirection to fake websites and counter measure techniques

Shona Start
ID w17019752
Northumbria University, KF5007 Security Case Project
Attack Analysis in Ethical Hacking
shona.start@northumbria.ac.co.uk

***Abstract*— If performed correctly DNS spoofing can be a very dangerous form of attack. Once implemented, a DNS spoof can be used to redirect traffic to a dishonest website where an attacker could capture data such as login credentials or financial information. A DNS spoof can even spread between mutable DNS servers making it even more hazardous. To conduct a safe replica of a DNS spoof a honeypot should be implanted, this will stop it spreading across a network. This report will analyse one such replica where Ettercap was used to establish a man in the middle attack and perform a DNS spoof. It will analyse which protocols were targeted by the DNS spoof and determine the exploits in these protocols which allow this to happen. The static ARP counter measures, will then be implemented onto this replica.**

***Index Terms*—DNS, DNS Spoofing, MITM, ARP, ARP spoofing. DNSSEC**

I. Introduction

DNS (Domain Name System) is a vital networking protocol. The modern internet is reliant on this protocol to map domain names to their corresponding IP (Internet Protocol) addresses. ARP is another vital networking protocol, used to map IP addresses to MAC (Media Access Control) addresses. The first section of this paper will be an analysis and discussion on how these protocols function. Although valuable, both DNS and ARP have many known exploits. These exploits allow for many types of attacks on the protocols, such as spoofing attacks. The second section of this report will focus on the software which can be used to perform spoofing attacks and which parts of the protocols are being exploited by this software. Finally, this report will discuss possible ways to protect against spoofing attacks on DNS and ARP, mainly focusing on Static ARP as a counter measure.

II. Protocol Standards

MITM (Man in the Middle) attacks take advantage of the weaknesses in authentication protocols. One of these protocols is ARP (Address Resolution Protocol). ARP is a function of the network layer of the TCP/IP suite; it was designed to approach the issue of networks requiring protocols to coexist on a network. A 10Mbit Ethernet will allow this, however a 10Mbit Ethernet requires 48 bit addresses, such as MAC addresses. This will complicate things because packets can only be transferred using IP addresses [1]. ARP was created to fix this issue as its main function is to map a hosts IP (Internet Protocol) address to a MAC (Media Access Control) address. ARP maintains a table of these mappings, stored in RAM (Random Access Memory), called an ARP table. The table contains mappings of known IP addresses and their corresponding MAC addresses. This means that when sending a packet, if a matching IP address is found in the ARP table it uses this to determine the destination MAC address. However, if a matching entry is not found, an ARP request will be sent out. An ARP request will contain the senders MAC address, sender's IP address, destination IP address and space for the destination MAC address. ARP requests are broadcasts, so are flooded to all ports in the network. Once an ARP request is received, a device must process this to determine if its IP address matches the destination address. The device should only reply if its IP address matches, if it does not, no future action should be taken. If it does match, the device will send an ARP reply. An ARP reply is similar to an ARP request, however, the device will fill in the previously empty MAC address field. [2, 3, 4].

Identification	Control
Question count	Answer count
Authority count	Additional count
Question	
…	
Answer	
…	
Authority	
…	
Additional	
…	

Figure 1: DNS message format

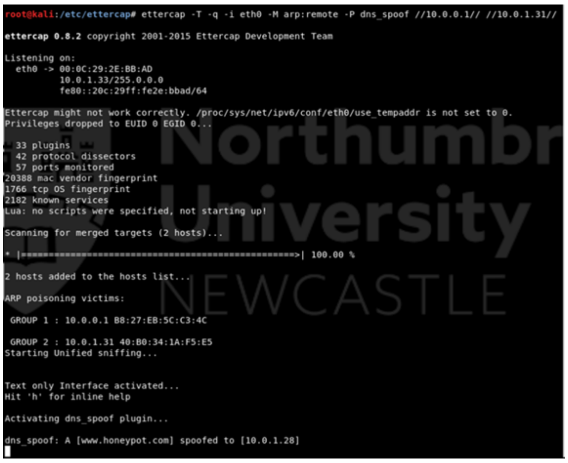
The DNS (Domain Name System) protocols main function is to translate IP addresses into URLs (Uniform Resource Locators). This is a vital service because internet based hosts and entities, such as websites, are identified using IP addresses. However, URLs are more readable and memorable then IP addresses. The destination port of DNS is 53, which works on both TCP and UDP. DNS uses a single message format for all communications, for example: queries, responses and errors.

This message format is shown in Figure 1 [5]. DNS uses a hierarchically distributed database. This database is made up of RR's (Resource Records). RR's are used to define data types, for example: A (address) records, AAAA (IPv6) records, MX (Mail Exchange) records, PT (Pointer) records, NS (Name Server) records, CNAME (Canonical Name) records and SOA (Start of Authority) records etc. RRs are globally accessible from DNS architecture. [6, 7]

DNS uses a hierarchical data structure of domain names. The topmost is root zone which is indicated by a "." dot. Under this, domains are grouped into TDLs (Top Level Domains). TDLs represent the type of organization or the country of origin, for example: .com, .org and .uk. Under the root zone is second level domains, which are the main part of the domain name. The bottom-most is the sub-domains. This structure is used so that each part is in small manageable zones and so that DNS is scalable. Each DNS server maintains a specific database file and will only manage translations for that section of the DNS structure. If a DNS server receives a DNS request for a translation not within its zone, it will forward the request to a DNS server within the correct zone. A DNS server does not need to know about all domains, only those immediately above and below it. [2, 5].

III. Attack software and its effect on the protocols

There are many software products such as Dnsspoof and Ettercap that can be used to perform a DNS spoof. However, this report will mainly be discussing an attack implanted using Ettercap in a VM (Virtual Machine) running Kali Linux. This was connected to a honeypot set up with a DNS server. A VM running Ubuntu Server and Apache was used to host the fake website.



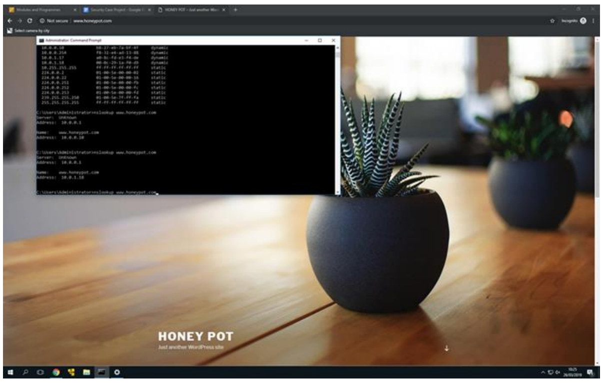
```
root@kali:/etc/ettercap# ettercap -T -q -i eth0 -M arp:remote -P dns_spoof //10.0.0.1// //10.0.1.31//

ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team

Listening on:
  eth0 -> 00:0C:29:2E:BB:AD
          10.0.1.33/255.0.0.0
          fe80::20c:29ff:fe2e:bbad/64

Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth0/use_tempaddr is not set to 0.
Privileges dropped to EUID 0 EGID 0...

  33 plugins
  42 protocol dissectors
  57 ports monitored
20388 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 10.0.0.1 B8:27:EB:5C:C3:4C

 GROUP 2 : 10.0.1.31 40:B0:34:1A:F5:E5
Starting Unified sniffing...


Text only Interface activated...
Hit 'h' for inline help

Activating dns_spoof plugin...

dns_spoof: A [www.honeypot.com] spoofed to [10.0.1.28]
```

Figure 2

Command line was used instead of the Ettercap GUI (Graphical User interface) as it is more efficient. The command shown in Figure 2 instructs Ettercap to activate a MITM attack (-M), perform an ARP poisoning attack (Arp:remote) and perform a DNS spoof (dns_spoof). The targets, being 10.0.0.1 and 10.0.1.31 on the eth0 interface; 10.0.0.1 is the server on the honeypot and 10.0.1.31 is the computer targeted to spoof. All of this is done on quiet mode (-q) and outside of promiscuous mode (-P). Ettercap is performing the attack in promiscuous mode, because only current connections will be targeted, not all the traffic on the wire. Utilising ARP spoofing the MITM will hijack this traffic and redirect it to Ettercap. [8].

ARP spoofing is a type of attack that exploits the ARP protocol. It involves modifying the mapping of an IP address by replying with the attackers MAC address to ARP requests that are meant for another device, for example the default gateway. The broadcaster of the request will add the incorrect MAC address to its ARP table meaning any traffic meant for that device will instead be sent to the attacker. This allows for reading and modification of packets [1, 4]. ARP spoofing is possible because the ARP protocol does not have any security mechanisms; ARP requests and replies do not have any authentication or verification methods and all devices on a network trust all ARP replies. ARP is a stateless protocol, meaning it does not keep any information on requests sent out or replies received. This will mean a device will automatically cache any ARP replies received even if no ARP requests were sent out. Even entries in the ARP table that are not expired will still be overwritten by any new reply packets. This shortcoming is exploited by attackers and allows ARP spoofing and ARP cache poisoning to occur. [1, 3, 9].

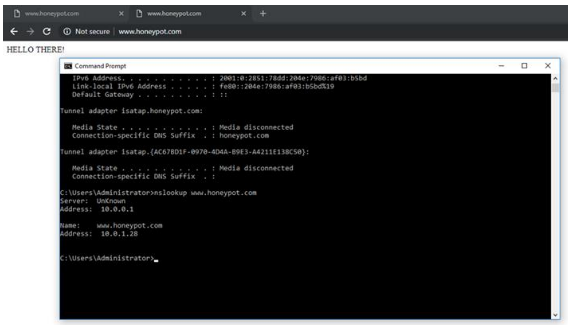
One type of packet ARP spoofing opens for reading and modifying is DNS packets. DNS packets are vulnerable to this because they are transferred without encryption mechanisms and their authentication methods can be tricked [RFC 5452]. DNS spoofing can occur in two ways, either creating malicious responses to DNS requests or tampering with the DNS server's cache. This report will focus on the first.



Not secure | www.honeypot.com
Administrator: Command Prompt
HONEY POT
Just another WordPress site

```
  10.0.0.10             b8-27-eb-7a-bf-4f     dynamic
  10.0.0.254            f8-32-e4-ad-13-88     dynamic
  10.0.1.17             a0-8c-fd-e1-f4-de     dynamic
  10.0.1.18             00-0c-29-1a-f0-d9     dynamic
  10.255.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.0.253           01-00-5e-00-00-fd     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\Administrator>nslookup www.honeypot.com
Server:  Unknown
Address:  10.0.0.1

Name:    www.honeypot.com
Address:  10.0.0.10


C:\Users\Administrator>nslookup www.honeypot.com
Server:  Unknown
Address:  10.0.0.1

Name:    www.honeypot.com
Address:  10.0.1.18


C:\Users\Administrator>nslookup www.honeypot.com
```

Figure 3



```
   IPv6 Address. . . . . . . . . . . : 2001:0:2851:78dd:204e:7986:af03:b5bd
   Link-local IPv6 Address . . . . . : fe80::204e:7986:af03:b5bd%19
   Default Gateway . . . . . . . . . : ::

Tunnel adapter isatap.honeypot.com:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : honeypot.com

Tunnel adapter isatap.{AC678D1F-0970-4D4A-B9E3-A4211E138C50}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\Administrator>nslookup www.honeypot.com
Server:  UnKnown
Address:  10.0.0.1

Name:    www.honeypot.com
Address:  10.0.1.28


C:\Users\Administrator>_
```

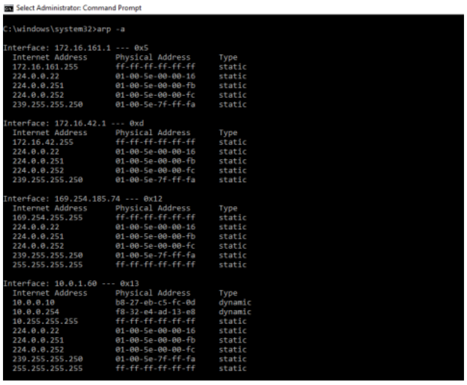
Figure 4

DNS data will only be accepted if it matches set requirements. However, the first matching response is always the one accepted. This means that if an attacker responds before the genuine server, and meets the requirements, the data it has provided will be treated as genuine. One of these requirements is a unique identification number. This ID can be gathered by intercepting DNS requests using the MITM. Once acquired, the attacker can respond to the requester with a corrupt packet masquerading as a genuine reply from the DNS server. This corrupt packet will contain the IP address of the attacker's website and therefore will direct the victim to this website instead of the site they are requesting to visit. Figure 3 and Figure 4 show before and after screencaps of a victim an attack of this manner. Using the command 'nslookup' will show that the IP address of the website has been replaced with the IP address of the attacker's website. The attacker will not be able to access this website until the attack has stopped and their DNS cache is flushed. A DNS cache can be flushed using the command 'ipconfig flushdns'. However, if a victim does not realise they are being attacked or know how to flush their DNS cache, their DNS cache will only flush following the TTL (Time to Live) of the RR (Resource Record). [3, 5]

IV. COUNTERMEASURE TECHNIQUES

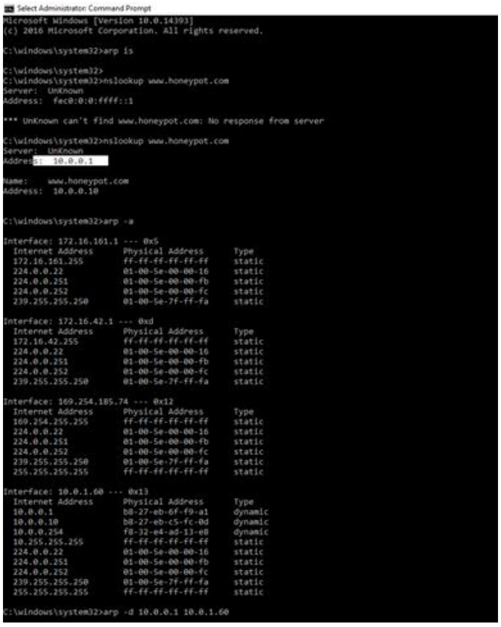
An ARP spoof can easily be detected using software such as XArp, Snort and ArpON. The ARP cache can even be used to detect an ARP spoof, for instance, if two IP (internet) addresses in the cache share the same MAC (physical) address then this will raise suspicion. Shown in Figure 5, the 'arp -a' command can be used to view the ARP cache.

As easy as an ARP spoof is to detect, preventing the attack in the first place is more effective. Static ARP is a counter measure technique that can reduce the risk of successful spoofing. Most networks use dynamic ARP, including the one discussed previously. Entries in this cache can be changed to static ARP, however. Static ARP involves manually mapping the IP addresses and MAC addresses of machines on a network. The Figures 5,6,7, and 8 show how this can be done.



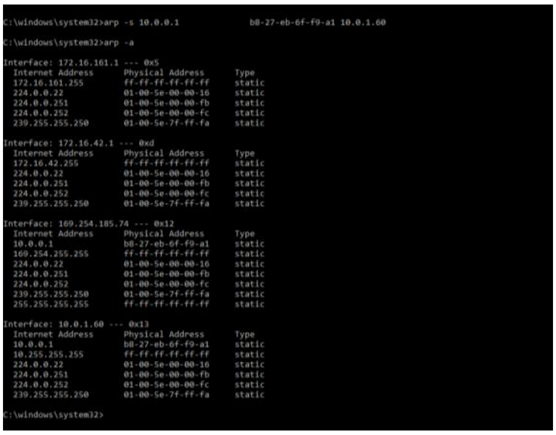
```
C:\windows\system32>arp -a

Interface: 172.16.161.1 --- 0x5
  Internet Address      Physical Address      Type
  172.16.161.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 172.16.42.1 --- 0xd
  Internet Address      Physical Address      Type
  172.16.42.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 169.254.185.74 --- 0x12
  Internet Address      Physical Address      Type
  169.254.255.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 10.0.1.60 --- 0x13
  Internet Address      Physical Address      Type
  10.0.0.10             b8-27-eb-c5-fc-0d     dynamic
  10.0.0.254            f8-32-e4-ad-13-e8     dynamic
  10.255.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Figure 5



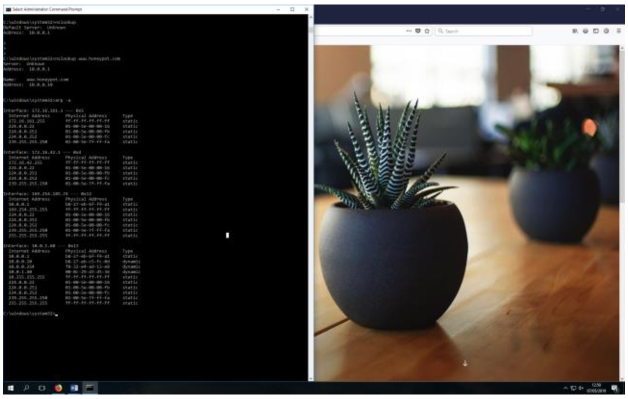
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\windows\system32>arp is

C:\windows\system32>
C:\windows\system32>nslookup www.honeypot.com
Server:  UnKnown
Address:  fec0:0:0:ffff::1

*** UnKnown can't find www.honeypot.com: No response from server

C:\windows\system32>nslookup www.honeypot.com
Server:  UnKnown
Address:  10.0.0.1

Name:    www.honeypot.com
Address:  10.0.0.10


C:\windows\system32>arp -a

Interface: 172.16.161.1 --- 0x5
  Internet Address      Physical Address      Type
  172.16.161.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 172.16.42.1 --- 0xd
  Internet Address      Physical Address      Type
  172.16.42.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 169.254.185.74 --- 0x12
  Internet Address      Physical Address      Type
  169.254.255.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 10.0.1.60 --- 0x13
  Internet Address      Physical Address      Type
  10.0.0.1              b8-27-eb-6f-f9-a1     dynamic
  10.0.0.10             b8-27-eb-c5-fc-0d     dynamic
  10.0.0.254            f8-32-e4-ad-13-e8     dynamic
  10.255.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\windows\system32>arp -d 10.0.0.1 10.0.1.60
```

Figure 6



```
C:\windows\system32>arp -s 10.0.0.1           b8-27-eb-6f-f9-a1 10.0.1.60

C:\windows\system32>arp -a

Interface: 172.16.161.1 --- 0x5
  Internet Address      Physical Address      Type
  172.16.161.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 172.16.42.1 --- 0xd
  Internet Address      Physical Address      Type
  172.16.42.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 169.254.185.74 --- 0x12
  Internet Address      Physical Address      Type
  10.0.0.1              b8-27-eb-6f-f9-a1     static
  169.254.255.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 10.0.1.60 --- 0x13
  Internet Address      Physical Address      Type
  10.0.0.1              b8-27-eb-6f-f9-a1     static
  10.255.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

C:\windows\system32>
```

Figure 7



```
C:\windows\system32>nslookup
Default Server:  UnKnown
Address:  10.0.0.1

>
>
>
C:\windows\system32>nslookup www.honeypot.com
Server:  UnKnown
Address:  10.0.0.1

Name:    www.honeypot.com
Address:  10.0.0.10

C:\windows\system32>arp -a

Interface: 172.16.181.1 --- 0x5
  Internet Address      Physical Address      Type
  172.16.181.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 172.16.42.1 --- 0xd
  Internet Address      Physical Address      Type
  172.16.42.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 169.254.185.78 --- 0x12
  Internet Address      Physical Address      Type
  10.0.0.1              b8-27-eb-6f-f9-a1     static
  169.254.255.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 10.0.1.60 --- 0x13
  Internet Address      Physical Address      Type
  10.0.0.1              b8-27-eb-6f-f9-a1     static
  10.0.0.10             b8-27-eb-c5-fc-0d     dynamic
  10.0.0.254            f8-32-e4-ad-13-a0     dynamic
  10.0.1.80             00-0c-29-d3-d5-30     dynamic
  10.255.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\windows\system32>
```

Figure 8

The command 'arp-a' shows the ARP cache of the machine, from here the ARP entries for this machine can be found. As this is a small experiment, where the machine is only connected to the honeypot, the entry for the honeypot is the only one needed. The IP address (1.0.0.1) for the honeypot can be found using 'nslookup'. Using 'arp -a', as shown in Figure 5, it can be seen that this address is already a dynamic entry in the ARP cache. To change this to a static entry, the dynamic entry must first be removed. This can be accomplished using the command 'arp -d 10.0.0.1 10.0.1.60'. To check if this has worked, the 'arp -a' command can once again be used, Figure 6 shows that the entry for the honeypot has been removed from the cache. A static entry for the honeypot can now be added to the ARP cache using the command 'arp -s 10.0.0.1 b8-27-eb-6f-f9-a1 10.0.1.60' as seen in Figure 7. This will permanently map the IP address of the honeypot to the MAC address of the honeypot, meaning that the machine will ignore all ARP replies that attempt to change the mapping. Figure 8 shows the results of the ettercap attack performed earlier, this time on a machine set up with static ARP. The command 'nslookup shows that the attacker was unable to perform a spoof as the IP address has not been altered. [11, 12]

Static ARP only protects against ARP spoofing, this in turn helps protect against DNS spoofing but it would be more secure to include further counter measures. One such counter measure is DNSSEC (Domain Name System Security Extensions). DNSSEC is an extension of DNS, which is designed to improve the security of the protocol by adding features to protect against fake DNS data. If correctly configured, DNSSEC should protect DNS from spoofing and cache poisoning attacks and it will even help defend against MITM attacks [13]. It uses public-key cryptography to create a digital signature, which must be included in all DNS replies to authenticate the sender of the message. To add a digital signature, DNSSEC adds more types of RRs, including RRSIG (Resource Record Signature), DNSKEY (DNS Public Key), DS (Delegation Signer) and NSEC (NEXT SECURE). Each DNS request will have its own RRSIG, which acts as the digital signature. When a DNS response is received for this request, the DNSKEY record will be used to decrypt the signature and check to see if it matches the signature sent out, if the two are identical then the validation is successful. DNSSEC also adds two header flags to DNS, these are CD (Checking Disabled) and AD (Authenticated Data). The CD flag is used to indicate that a DNS server should not validate the DNS response. The AD flag is used by DNS servers to indicate to RRs that the reply has been authenticated. DNSSEC does not protect against modification of a DNS message header. This means that if the connection between the DNS server and the requester is unsecure, then the header and flags could be modified. [14, 15]

V. Conclusion

ARP and DNS are both vital protocols which are essential to the function of the internet. However, they both have dangerous security weaknesses like a lack of authentication and encryption mechanisms. These exploits open networks using these protocols up to spoofing, as attackers are able to modify the ARP cache and from there alter DNS packets so that victims are directed to malicious websites. Countermeasures such as static ARP and DNSSEC are effective ways of protecting against this. However, these countermeasures also have disadvantages, in further work it would be effective to look more closely at the down sides of these and see if these can be improved upon in any way. For instance, Static ARP could be compared against semi-static ARP (a method in which static ARP entries are configures automatically). Alternatives to these countermeasures could also be investigated such as protection software's like AntiARP. It would also be beneficial to study how these countermeasures would be handle in an alternative environment, for instance on a larger network or against a different attack software, to see if they are more or less effective.

References

[1] D. C. Plummer, "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware," Network Working Group, 1982.

[2] Cisco. (2017, Sept 5). CCNA Routing and Switching: Intruduction to Networking (v5.1) [Ebook]. Available: https://static-course-assets.s3.amazonaws.com/ITN51/en/index.html

[3] S. Gangan, "A Review of Man-in-the-Middle Attacks," Cornell University, Ithaca, New York, 2015. Available: https://arxiv.org/ftp/arxiv/papers/1504/1504.02115.pdf

[4] M. Ataullah and N. Chauhan, "ES-ARP: An efficient and secure Address Resolution Protocol," 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, 2012. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6184794&isnumber=6184719

[5] P. Mockapetris, "Domain Names - Concepts and Facility's," Network Working Group, 1983. Available: https://tools.ietf.org/html/rfc882

[6] M. H. Jalalzai, W. B. Shahid and M. M. W. Iqbal, "DNS security challenges and best practices to deploy secure DNS with digital signatures," 2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, 2015. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7058517&isnumber=7058466

[7] S.P. Singh, “The Use of DNS Resource Records,” Infosys Limited, Maharashtra, India, 2014. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.640.444&rep=rep1&type=pdf

[8] A. Ornaghi, M. Valleri. (2005, Jul. 6). Ettercap Man Page [Online]. Available: https://www.mankier.com/8/ettercap#Project_Stewards

[9] V. Ramachandran, S. Nandi, “Detecting ARP Spoofing: An Active Technique,” Cisco Systems, Inc., Bangalore, India., 2005. Available: https://page-one.springer.com/pdf/preview/10.1007/11593980_18

[10] A. Hubert, “Measures for Making DNS More Resilient against Forged Answers,” Network Working Group, 2009. Available: https://tools.ietf.org/html/rfc5452

[11] S. Hijazi and M. S. Obaidat, "A New Detection and Prevention System for ARP Attacks Using Static Entry," IEEE Systems Journal, 2018. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8540917

[12] T. Chomsiri, "Sniffing Packets on LAN without ARP Spoofing," 2008 Third International Conference on Convergence and Hybrid Information Technology, Busan, 2008. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4682286

[13] D. Atkins, “Threat Analysis of the Domain Name System (DNS),” Network Working Group, 2004. Available: https://tools.ietf.org/html/rfc3833

[14] M. A. Hussain, H. Jin, Z. A. Hussien, Z. A. Abduljabbar, S. H. Abbdal and A. Ibrahim, "DNS Protection against Spoofing and Poisoning Attacks," 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), Beijing, 2016. Available: https://ieeexplore.ieee.org/document/7726376