Name: Thomas Finn

Email: [thomasjf@usc.edu](mailto:thomasjf@usc.edu)

User: usc430af

QUESTIONS

Part 1: Understanding DNS

1. The dig command sent a DNS query to some DNS server and received a response from that server. What is the IP address at that server?
   * The IP address at the server is 10.1.1.3.
2. The DNS response includes a status. What was the status of the response?
   * The status is ‘NOEEROR’ which means the zone is being served from the requested authority without issues.
3. The response identified the IP address of www.google.com. What is the reported IP address of [www.google.com](http://www.google.com)?
   * The reported IP address of [www.google.com](http://www.google.com) is 10.1.2.155.
4. How long will the www.google.com IPv4 address be cached?
   * It will be cached for 10 seconds.
5. The response listed the nameserver that is authoritative for www.google.com. What is the authoritative name server for google.com?
   * The name server for google.com is ns.google.com
6. Finally, the response listed the IPv4 address for the google.com nameserver. For each such server what is its IPv4 address?
   * For ns.google.com, the IP address is 10.1.2.3.

Part 2: The Big Picture

1. State the source MAC and IP addresses as well as destination MAC and IP addresses for a packet going from the client to the cache.
   * Source MAC: 00:11:43:d5:f4:c2  
     Source IP: 10.1.1.2  
     Destination MAC: 00:04:23:ae:ce:74  
     Destination IP: 10.1.1.3
2. Does the packet travel through the attacker box?
   * No, there is a direct link from the client to the cache.
3. State the source MAC and IP addresses as well as destination MAC and IP addresses for a packet going from the cache to the authoritative server.
   * Source MAC: 00:04:23:ae:ce:75  
     Source IP: 10.1.2.2  
     Destination MAC: 00:04:23:ae:cc:86  
     Destination IP: 10.1.2.3
4. Does the packet travel through the attacker box?
   * Yes, it is very possible for the attacker to redirect the traffic through itself since the attacker is attached to the lan that is between the cache and server.

Part 3: Using Ettercap

1. The command you used.
   * sudo ettercap -T -q -M ARP /10.1.2.2//
2. What each option in the command means.
   * -T tells Ettercap to use the text interface and allow command prompt input  
     -q uses quiet mode for the console interface and is useful for converting pcap files to ettercap log files.  
     -M uses the Man in the Middle method and requires an argument to specify what type of MITM attack is being used. Types include arp, icmp, dhcp, and port. The aim of the attack is to hijack packets and redirect them to ettercap.  
     /10.1.2.2// defines the ip address for the lan network that holds both auth and the attacker. This terminology tells Ettercap to target this IP as the victim.
3. State the source MAC and IP addresses as well as destination MAC and IP addresses for a packet going from the cache to the authoritative server.
   * Source MAC: 00:04:23:ae:ce:75  
     Source IP: 10.1.2.2  
     Destination MAC: 00:0e:0c:66:84:01  
     Destination IP: 10.1.2.3
4. Does the packet travel through the attacker box?
   * Yes the packet travels first to the attack box.
5. If your answers to 3 and 4 differ from answers to 3 and 4 in Part 2, explain why.
   * The answer does change in the fact that the packet sent from the cache to auth goes to the attacker’s MAC address. The ettercap attack spoofs the IP but injects its own MAC as the destination address so the ARP returns an incorrect MAC to the interface in charge of getting the packet to the auth server.
6. The complete command (or steps in the GUI) used to have ettercap forge a DNS message and any necessary configuration files.
   * Edited ettercaps etter.dns file using following command:   
      sudo nano /etc/ettercap/etter.dns  
     Added following line to redirect to attacker IP:  
      [www.google.com](http://www.google.com) A 10.1.2.4
   * Launched ettercap attack using following command:  
      sudo ettercap --plugin dns\_spoof -T -q -M ARP /10.1.2.2//
7. What malicious things could an attacker do by changing the IP address in a DNS response going to the client?
   * By changing the IP address to a site that the attacker owns, the attacker can thus spy on the client’s traffic, intercept and modify packets in a man in the middle attack, drop packets meant for the target to create a denial of service attack. Imagine if the attacker redirected the client to a look alike page. The attacker could write a script that fetches the requested page and listens in on the login automatically thus stealing the clients confidential data.

Part 4: Implementing DNSSEC

1. The signed response obtained on the client machine.

|  |
| --- |
| root@client:/users/usc430af# dig +dnssec www.google.com  ; <<>> DiG 9.10.3-P4-Ubuntu <<>> +dnssec www.google.com  ;; global options: +cmd  ;; Got answer:  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62659  ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 14, ADDITIONAL: 1  ;; OPT PSEUDOSECTION:  ; EDNS: version: 0, flags: do; udp: 4096  ;; QUESTION SECTION:  ;www.google.com. IN A  ;; ANSWER SECTION:  www.google.com. 87 IN A 216.58.217.196  ;; AUTHORITY SECTION:  . 246034 IN NS h.root-servers.net.  . 246034 IN NS a.root-servers.net.  . 246034 IN NS b.root-servers.net.  . 246034 IN NS g.root-servers.net.  . 246034 IN NS d.root-servers.net.  . 246034 IN NS j.root-servers.net.  . 246034 IN NS k.root-servers.net.  . 246034 IN NS e.root-servers.net.  . 246034 IN NS c.root-servers.net.  . 246034 IN NS m.root-servers.net.  . 246034 IN NS f.root-servers.net.  . 246034 IN NS l.root-servers.net.  . 246034 IN NS i.root-servers.net.  . 246034 IN RRSIG NS 8 0 518400 20190424050000 20190411040000 25266 . F3eloidQmp8AGj0oclupkEGLQVwqI2b1UuO6ucd9Bo4hth7enhCvg4Fw Yk//pax5N5254J2H8hcdRNMmp7LM4bc5vzBcwcRJ/18/LB1GNM6QATJe XK/fzj+iacGPo7D7WiXlbBgP21UkkYGLokyyI9aSIbF2StnUMpsq/Sep 8+tVyUhgFx6Ymd68PpZLTOFDl95bZGz2vSm8rqp8WR+VYptqY6Dmtg9N rGjPK3mN/hnT+traUfE9ZQ0+48CIf0a29zYkCqlY/AGWCRrT09wHFImZ wyJplm0pFUOH13bydKF6C12PfNhTTWxboJWERP2RaDjgwWz/thFNQtNC jI/PJg==  ;; Query time: 7 msec  ;; SERVER: 10.1.1.3#53(10.1.1.3)  ;; WHEN: Fri Apr 12 23:41:20 PDT 2019  ;; MSG SIZE rcvd: 556 |

1. Detailed description of all the steps you took to implement DNSSEC. Make sure to list all commands you typed and all configuration changes you made.
   * On auth node (as root):
     1. cd /etc/bind/
     2. zonesigner -genkeys -endtime +2678400 google.com
     3. nano named.conf.options  
        Add lines: dnssec-enable yes;  
         dnssec-validation yes;  
         dnssec-lookaside auto;
     4. nano named.conf.local  
        Replace line: google.com → google.com.signed
     5. service bind9 reload
     6. service bind9 restart
     7. Dig +dnssec [www.google.com](http://www.google.com) A
     8. cp \*.key /users/usc430af
   * On cache node (as root):
     1. mv \*.key /etc/bind/
     2. cd /etc/bind/
     3. nano named.conf.options  
        Add lines: dnssec-enable yes;  
         dnssec-validation yes;  
         dnssec-lookaside auto;
     4. service bind9 restart
     5. dig +dnssec [www.google.com](http://www.google.com) A
   * On client node:
     1. Dig +dnssec [www.google.com](http://www.google.com) A → Receives signed response