|  |  |  |  |
| --- | --- | --- | --- |
| Windows and Linux commands and software used in this worksheet are documented below. Each command has a link were additional information can be obtained to further understand the command features. | | | |
| Command/Tool | Description | Link | Appendix |
| BiDiBlah v2.0 | Windows footprinting tool that leverages web search (Google, Yahoo, Windows Live), dictionary DNS enumeration, and reverse lookup to identify all external hosts. | <http://www.sensepost.com/labs/tools/pentest/bidiblah> |  |
| SiteDigger 3.0 | Windows footprinting tool that leverages the Google Hacking Database to identify “Google Dorks”. These are weaknesses that have been cached by Google found by specific Google Queries. | <http://www.mcafee.com/us/downloads/free-tools/sitedigger.aspx> |  |
| dig | A flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried | <http://linux.die.net/man/1/dig> | Windows [Install](#diginstall) |
| whois  whosip  whoiscl | Standard Linux command for querying domain and IP registrant information. | Windows: <http://www.nirsoft.net/utils/whoiscl.html>  Windows: <http://www.nirsoft.net/utils/whosip.html> |  |
|  |  |  |  |
| traceroute | Map network path from workstation to target host using ICMP packates. |  |  |
| tracert | Windows traceroute command. |  |  |
| tcptraceroute | Map network path from workstation to target host using TCP packets. This tool may have more success than traceroute as firewalls can be configured to drop ICMP packets. |  |  |
| tracetcp | Windows tcptraceroute tool. | <http://tracetcp.sourceforge.net/> |  |
| NetCat | Netcat is a featured networking utility which reads and writes data across network connections, using the TCP/IP protocol. It is the swiss army knife of the TCP/IP protocol. | <http://netcat.sourceforge.net/>  Windows: <http://joncraton.org/files/nc111nt.zip> |  |
| stunnel | multiplatform SSL tunneling proxy | <http://www.stunnel.org/> | Linux [Install](#installstunnel) |
| Nmap | Nmap ("Network Mapper") is a free and open source utility for network exploration or security auditing. | <http://www.nmap.org> | Linux [Install](#installnmap) |

Advanced Tools

|  |  |  |  |
| --- | --- | --- | --- |
| scapy | Scapy is a powerful interactive packet manipulation program. It is able to forge or decode packets of a wide number of protocols, send them on the wire, capture them, match requests and replies, and much more. | <http://www.secdev.org/projects/scapy/demo.html> | Linux [Install](#scapyinstall) |
| dnswalk | dnswalk is a DNS debugger. It performs zone transfers of specified domains, and checks the database in numerous ways for internal consistency, as well as accuracy. | <http://sourceforge.net/projects/dnswalk/> | Linux [Install](#installdnswalk) |
| dnsenum | The purpose of Dnsenum is to gather as much information as possible about a domain. | <http://code.google.com/p/dnsenum/> | Linux [Install](#installdnsenum) |
| goog-mail.py | Python script that scrapes Google for email addresses of the supplied domain name. Found on the Backtrack 4 Live CD. | http://www.jedge.com/utilities/goog-mail.py |  |
| dnsmap | Passive DNS network mapper a.k.a. subdomains bruteforcer | <http://dnsmap.googlecode.com/> | Linux [Install](#installdnsmap) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

| **Task** | **Steps and Description** | **Initials** | **Date** | **Linked Results** |
| --- | --- | --- | --- | --- |
| 1 | **Web search (Google, Yahoo, Bing) organization domains to enumerate websites and email addresses.**  **Windows**  A Windows tool from Sensepost called BiDiBlah can be used to scour the web and identifies email addresses and websites from the domain being searched.  **Linux (and Windows with Python installed)**  #goog-mail.py <domain> |  |  | [EV1](#EV1) |
| 2 | **Search web forums and newsgroup postings for email posts related to information technology.**  A generic search of “@<agency\_email\_domain>” can yield results on newsgroups. However a Windows tool from Sensepost called BiDiBlah or the Python script goog-mail.py, used in step 1, identified email addresses from the organization. Search the web with the email addresses found to see if they are related to information technology posts on forums or newsgroups.  In additions create a “users” file from all the email addresses gathered. Each email address is a potential username that can be used to gain access to a system. |  |  | [EV2](#EV2) |
| 3 | **Search job databases**  Just like regular search engines, job search sites could reveal a plethora of information on technology and services running on the target’s internal network. An assessor should carefully review the job postings published by the target on their own website or on other popular job search sites.  Process  • Check for resumes available on the target website  • Check various job databases (i.e. monster, hotjobs, careerbuilder, & dice)  • Search using search engines  • Check for job postings on the target website  • Check for job postings on job sites  • Focus on resumes/ads where technology experience is required |  |  | [EV3](#EV3) |
| 4 | **Run Foundstone Sitedigger tool against agency address to enumerate common Google Dorking web vulnerabilities.** |  |  | [EV4](#EV4) |
| 5 | **Identify authoritative DNS servers for the agency.**  These authoritative name servers can be found by querying the DNS infrastructure. We will query our own dns server and ask it who controls the agency’s address.  Linux  #cat /etc/resolv.conf 🡸 identify your name server for step 2  #dig ns <agency\_domain> @<any\_nameserver>  Windows  C:\>ipconfig /all 🡸 identify your name server for step 2  C:\Tools\dig\dig ns <agency\_domain> @<any\_nameserver> |  |  | [EV5](#EV5) |
| 6 | **Whois Lookup**  The whois utility is used to obtain the registered information for the domain name or ip address space. This will help gather additional information about the auditee/client/target. Whois can be used from the OS command line as well as a number of web services.  **Windows**  C:\>whoiscl <domain\_name>  C:\>nslookup <domain\_name>  C:\>whosip <ip\_address> 🡸 obtained from nslookup command  **Linux**  $whois <domain\_name>  $nslookup <domain\_name>  $whois <ip\_address> 🡸 obtained from nslookup command |  |  | [EV6](#EV6) |
| 7 | **Identify the perimeter of the network segment.**  **Trace ICMP and TCP to web target.** Run a TRACEROUTE to the targets web server and document the results. A properly configured firewall will drop ICMP packets. This means that the last hop to respond back will be the last router BEFORE the firewall. This is useful in knowing the number of hops to the firewall.  **Linux**  #traceroute <webserver\_ip>  #tcptraceroute <webserver\_ip>  **Windows**  C:\>tracert <webserver\_ip>  C:\>tools\tracetcp <webserver\_ip> |  |  | [EV7a](#EV7a)  [EV7b](#EV7b) |
| 8 | **Trace and Graph TCP to select target (i.e. webserver) using SCAPY.**  #scapy  >>> res,unans = traceroute(["**<target\_webserver>**"],dport=[80,443],maxttl=20,retry=-2)  >>> res.graph(target="> /tmp/graph.svg")  The image will be called graph.svg in the /tmp directory. Results may vary depending on the router sitting between your workstation and the internet. Your router may not send the packets back to your machine. It is best to have your machine directly connected to the internet for this test. See Appendix B for example results (both correct and incorrect).  For viewing the graphics file in Windows without downloading special software it is best to convert the file to a PNG. Ensure you have imagemagick installed.  #convert +antialias /tmp/graph.svg /tmp/graph.png |  |  | [EV8](#EV8) |
| 9 | **Identify email servers via DNS query (MX record)**  #dig @<domain\_DNS\_server> -t MX <agency\_domain> |  |  | [EV9](#EV9) |
| 10 | **Query DNS server for common server names.**  **Windows**  The easiest and most complete way to accomplish this is in conjunction with steps 1 and 7 with a tool for Windows from Sensepost called BiDiBlah. BiDiBlah has dictionary files of common server names. These lists are run against the agency DNS server to enumerate additional hosts.  **Linux**  The Linux perl script dnsenum can be used to brute force hosts with the supplied dictionary file.  #perl dnsenum.pl --file dns\_words.txt <domain>  You can also use the compiled program dnsmap  #./dnsmap <domain> -w wordlist.txt |  |  | [EV10](#EV10) |
|  | **Perform reverse lookup against DNS server.**  **Windows**  The easiest and most complete way to accomplish this is in conjunction with steps 1, 7, and 10 with a tool for Windows from Sensepost called BiDiBlah.  After a whois lookup is done on the IP addresses from step 7 that range is then scanned for reverse lookup responses from the agency DNS server.  **Linux**  The Linux perl script dnsenum can be used to reverse lookup domain names via ip address ranges that have been identified.  #perl dnsenum.pl --recursion –-file <word\_list> <domain>  Also, a real easy way to do a reverse lookup given an ip address range is just to use Nmap.  # nmap -R -sL -o reverse.txt --dns-servers <DNS\_Server> <IP\_Address\_Range> |  |  | [EV11](#EV11)  [EV11b](#EV11b) |
|  | **Check target for zone transfer and DNS issues**  dnswalk can be quickly used to identify issues with a DNS record file or if a domain allows zone transfers. I will identify all DNS servers that maintain records of the domain and try to audit each of them.  #./dnswalk <dns\_domain> |  |  | [EV12](#EV12) |
|  | **Perform zone transfer against DNS server.**  #dig @<domain\_DNS\_server> -t AXFR <agency\_domain> |  |  | [EV13](#EV13) |
|  |  |  |  |  |
|  | **Banner grap smtp, http, dns, ftp, and https**  Attempt to connect to any of the hosts identified from previous steps. Attempt to connect to the web servers on port 21 (ftp). Attempt to connect to ports 22 and 23 on all hosts identified.  #echo "" | nc -v -n -w1 <ip\_address> 21-23  #nc <host> 25  HELO .com  #dig @<host>  #(echo HEAD / HTTP/1.0; echo; ) | nc <host> 80  #(echo HEAD / HTTP/1.0; echo; ) | stunnel -c -r <host>:443 |  |  | [EV14](#EV14) |
|  |  |  |  |  |

EV1 –(example) ([Task 1](#task1))

root@e-ubuntu:~/tools# python goog-mail.py agency.state.xx.us

+++++++++++++++++++++++++++++++++++++++++++++++++++++

+ Google Web & Group Results:

+++++++++++++++++++++++++++++++++++++++++++++++++++++

Recruitment@agency.state.xx.us

dro...@agency.state.xx.us

MO-EBO@agency.state.xx.us

jwheeler1@agency.state.xx.us

permits@agency.state.xx.us

GRiederer@agency.state.xx.us

pweykamp@agency.state.xx.us

nymoving@agency.state.xx.us

rstark@agency.state.xx.us

rpeck@agency state xx us

sroden@agency.state.xx.us

gchristian@agency.state.xx.us

vcavaleri@agency.state.xx.us

rpersaud@agency.state.xx.us

nloconnell@agency.state.xx.us

rdimauro@agency.state.xx.us

jrapoli@agency.state.xx.us

jflint@agency.state.xx.us

Jhewitt@agency.state.xx.us

jmhigley@agency.state.xx.us

...MO-RecordsAccess@agency.state.xx.us

Contactsmreuss@agency.state.xx.us

nlynch@agency.state.xx.us

Memberssmunson@agency.state.xx.us

DWOODIN@agency.state.xx.us

walbert@agency.state.xx.us

gsiletzky@agency.state.xx.us

CSchleede@agency.state.xx.us

PlanHELPDESK@agency.state.xx.us

primmer@agency.state.xx.us

wtelovsky@agency.state.xx.us

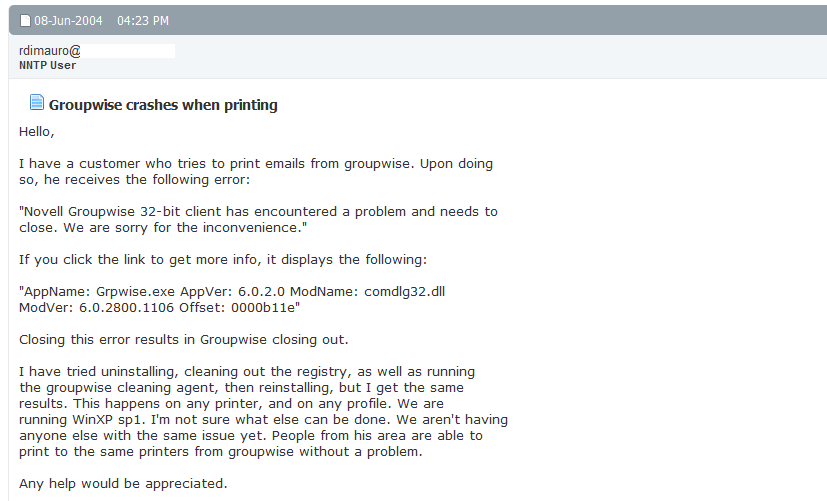
jminotti@agency.state.xx.us

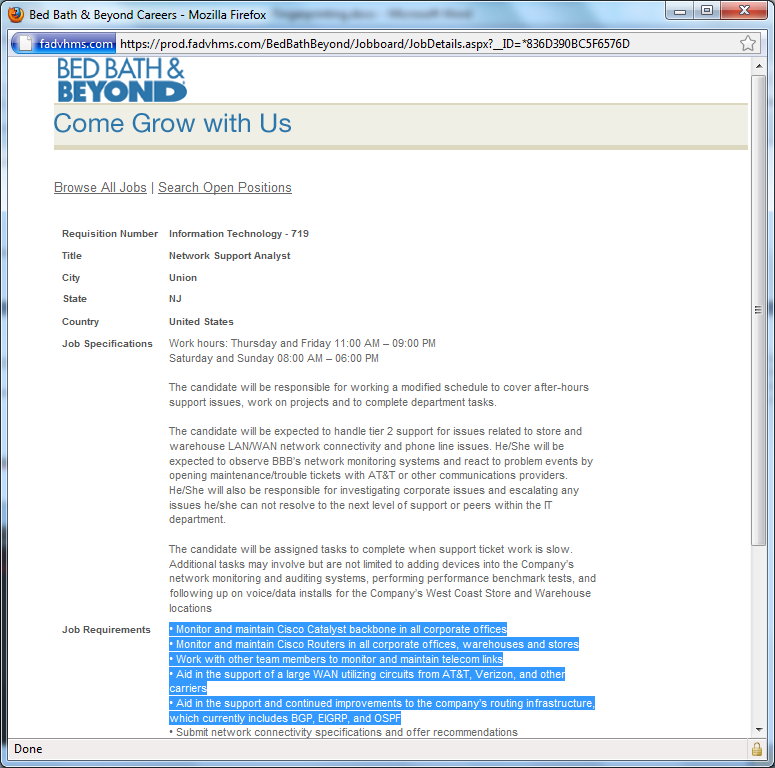
aglynn@agency.state.xx.us

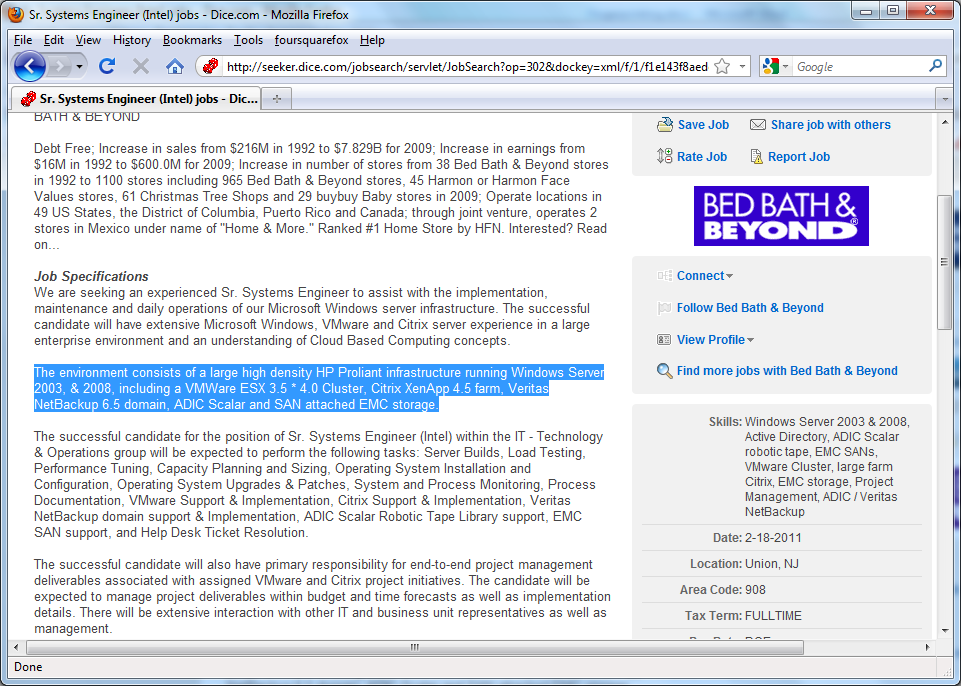
2009-11GBogacz@agency.state.xx.us

rmcdonough@agency.state.xx.us

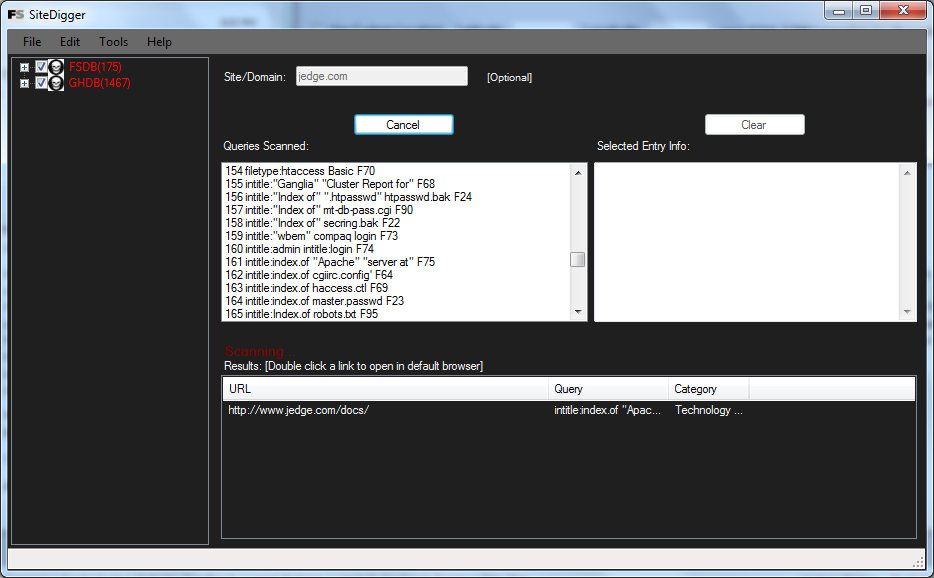
EV2 – Example forum post that may reveal too much information about the organization ([Task 2](#task2))

EV3 – Job Postings from the company website and a job board ([Task 3](#task3))





* EV4 – SiteDigger Screenshot ([Task 4](#task4)).

EV5 – Dig command results ([Task 5](#task5)).

; <<>> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.el5\_5.3 <<>> ns georgia.gov @192.168.0.1

;; global options: printcmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54673

;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:

;georgia.gov. IN NS

;; ANSWER SECTION:

georgia.gov. 28800 IN NS statens2.state.ga.us.

georgia.gov. 28800 IN NS ns1.state.ga.us.

georgia.gov. 28800 IN NS ns3.state.ga.us.

georgia.gov. 28800 IN NS ns2.state.ga.us.

georgia.gov. 28800 IN NS statens1.state.ga.us.

;; Query time: 39 msec

;; SERVER: 192.168.0.1#53(192.168.0.1)

;; WHEN: Fri Feb 18 14:55:39 2011

;; MSG SIZE rcvd: 140EV6 – Whois Lookup ([Task 6](#task6)).

|  |  |
| --- | --- |
| 1. #whois google.com  Registrant:  Dns Admin  Google Inc.  Please contact contact-admin@google.com 1600 Amphitheatre Parkway  Mountain View CA 94043  US  dns-admin@google.com +1.6502530000 Fax: +1.6506188571  Domain Name: google.com  Registrar Name: Markmonitor.com  Registrar Whois: whois.markmonitor.com  Registrar Homepage: http://www.markmonitor.com  Administrative Contact:  DNS Admin  Google Inc.  1600 Amphitheatre Parkway  Mountain View CA 94043  US  dns-admin@google.com +1.6506234000 Fax: +1.6506188571  Technical Contact, Zone Contact:  DNS Admin  Google Inc.  2400 E. Bayshore Pkwy  Mountain View CA 94043  US  dns-admin@google.com +1.6503300100 Fax: +1.6506181499  Created on..............: 1997-09-15.  Expires on..............: 2011-09-13.  Record last updated on..: 2011-02-05.  Domain servers in listed order:  ns2.google.com  ns1.google.com  ns4.google.com  ns3.google.com | 2. #nslookup google.com  Non-authoritative answer:  Name: google.com  Address: 74.125.157.99  Name: google.com  Address: 74.125.157.104  Name: google.com  Address: 74.125.157.147 |
| 3. #whois  NetRange: 74.125.0.0 - 74.125.255.255  CIDR: 74.125.0.0/16  OriginAS:  NetName: GOOGLE  NetHandle: NET-74-125-0-0-1  Parent: NET-74-0-0-0-0  NetType: Direct Allocation  NameServer: NS2.GOOGLE.COM  NameServer: NS3.GOOGLE.COM  NameServer: NS4.GOOGLE.COM  NameServer: NS1.GOOGLE.COM  RegDate: 2007-03-13  Updated: 2007-05-22  Ref: http://whois.arin.net/rest/net/NET-74-125-0-0-1  OrgName: Google Inc.  OrgId: GOGL  Address: 1600 Amphitheatre Parkway  City: Mountain View  StateProv: CA  PostalCode: 94043  Country: US  RegDate: 2000-03-30  Updated: 2009-08-07  Ref: http://whois.arin.net/rest/org/GOGL  OrgTechHandle: ZG39-ARIN  OrgTechName: Google Inc  OrgTechPhone: +1-650-253-0000  OrgTechEmail: arin-contact@google.com  OrgTechRef: http://whois.arin.net/rest/poc/ZG39-ARIN |

EV7a – Traceroute results ([Task 7](#task7))

root@e-ubuntu:~# traceroute www.jedge.com

traceroute to www.jedge.com (74.220.207.132), 30 hops max, 60 byte packets

1 192.168.2.254 (192.168.2.254) 4.715 ms 4.585 ms 7.013 ms

2 132.sub-66-174-175.myvzw.com (66.174.175.132) 72.575 ms 124.967 ms 125.263 ms

3 \* \* \*

4 145.sub-66-174-36.myvzw.com (66.174.36.145) 199.153 ms 199.388 ms 199.943 ms

5 98.sub-66-174-36.myvzw.com (66.174.36.98) 203.174 ms 209.869 ms 218.216 ms

6 6.sub-69-83-33.myvzw.com (69.83.33.6) 227.900 ms 142.326 ms 142.756 ms

7 3.sub-69-83-33.myvzw.com (69.83.33.3) 107.173 ms 107.308 ms 110.093 ms

8 253.sub-69-83-33.myvzw.com (69.83.33.253) 111.813 ms 113.821 ms 114.380 ms

9 12.89.31.61 (12.89.31.61) 117.374 ms \* 116.128 ms

10 \* \* \*

11 fdlfl01jt.ip.att.net (12.122.81.29) 145.832 ms 145.689 ms 145.387 ms

12 192.205.36.254 (192.205.36.254) 148.271 ms 99.677 ms 114.349 ms

13 ae-32-52.ebr2.Miami1.Level3.net (4.69.138.126) 116.053 ms 106.759 ms 103.991 ms

14 ae-2-2.ebr2.Atlanta2.Level3.net (4.69.140.142) 112.014 ms 138.127 ms 138.186 ms

15 ae-72-72.csw2.Atlanta2.Level3.net (4.69.148.250) 121.093 ms ae-62-62.csw1.Atlanta2.Level3.net (4.69.148.238) 173.695 ms 140.634 ms

16 ae-71-71.ebr1.Atlanta2.Level3.net (4.69.148.245) 172.547 ms 172.355 ms ae-61-61.ebr1.Atlanta2.Level3.net (4.69.148.233) 172.235 ms

17 ae-6-6.ebr1.Washington12.Level3.net (4.69.148.106) 180.890 ms 190.928 ms 190.666 ms

18 ae-1-100.ebr2.Washington12.Level3.net (4.69.143.214) 190.494 ms 190.153 ms 216.714 ms

19 4.69.148.49 (4.69.148.49) 167.360 ms 189.146 ms 204.527 ms

20 ae-71-71.csw2.NewYork1.Level3.net (4.69.134.70) 205.225 ms \* \*

21 ae-4-99.edge3.NewYork1.Level3.net (4.68.16.209) 121.124 ms ae-1-69.edge3.NewYork1.Level3.net (4.68.16.17) 144.942 ms ae-3-89.edge3.NewYork1.Level3.net (4.68.16.145) 145.584 ms

22 BLUEHOST-IN.edge3.NewYork1.Level3.net (4.26.35.98) 192.965 ms 191.138 ms 191.613 ms

23 tg2-5.ar01.prov.acedatacenters.com (69.195.64.41) 191.812 ms 200.950 ms 201.490 ms

24 host132.hostmonster.com (74.220.207.132) 203.458 ms 206.762 ms 208.874 ms

EV7b –TCPTraceroute results ([Task 7](#task7))

root@e-ubuntu:~# tcptraceroute www.jedge.com

Selected device eth0, address 192.168.2.103, port 52128 for outgoing packets

Tracing the path to www.jedge.com (74.220.207.132) on TCP port 80 (www), 30 hops max

1 192.168.2.254 2.524 ms 2.166 ms 3.161 ms

2 132.sub-66-174-175.myvzw.com (66.174.175.132) 59.230 ms 54.153 ms 59.083 ms

3 \* \* \*

4 201.sub-69-83-43.myvzw.com (69.83.43.201) 93.733 ms 80.251 ms 81.526 ms

5 98.sub-66-174-36.myvzw.com (66.174.36.98) 79.322 ms 74.339 ms 88.353 ms

6 6.sub-69-83-33.myvzw.com (69.83.33.6) 88.139 ms 82.665 ms 82.259 ms

7 3.sub-69-83-33.myvzw.com (69.83.33.3) 82.295 ms 88.504 ms 95.971 ms

8 253.sub-69-83-33.myvzw.com (69.83.33.253) 108.682 ms 188.511 ms 77.345 ms

9 12.89.31.61 84.693 ms 124.828 ms 120.428 ms

10 \* \* \*

11 fdlfl01jt.ip.att.net (12.122.81.29) 95.569 ms 88.542 ms 83.216 ms

12 192.205.36.254 131.049 ms 93.077 ms 95.506 ms

13 \* ae-32-52.ebr2.Miami1.Level3.net (4.69.138.126) 147.726 ms 98.632 ms

14 ae-2-2.ebr2.Atlanta2.Level3.net (4.69.140.142) 113.210 ms 107.538 ms 107.887 ms

15 ae-72-72.csw2.Atlanta2.Level3.net (4.69.148.250) 112.365 ms 112.815 ms 10 6.933 ms

16 ae-71-71.ebr1.Atlanta2.Level3.net (4.69.148.245) 99.927 ms 107.251 ms 205.416 ms

17 ae-6-6.ebr1.Washington12.Level3.net (4.69.148.106) 114.900 ms 124.174 ms 124.107 ms

18 ae-1-100.ebr2.Washington12.Level3.net (4.69.143.214) 131.660 ms 119.146 ms 122.574 ms

19 4.69.148.49 128.262 ms 199.605 ms 132.331 ms

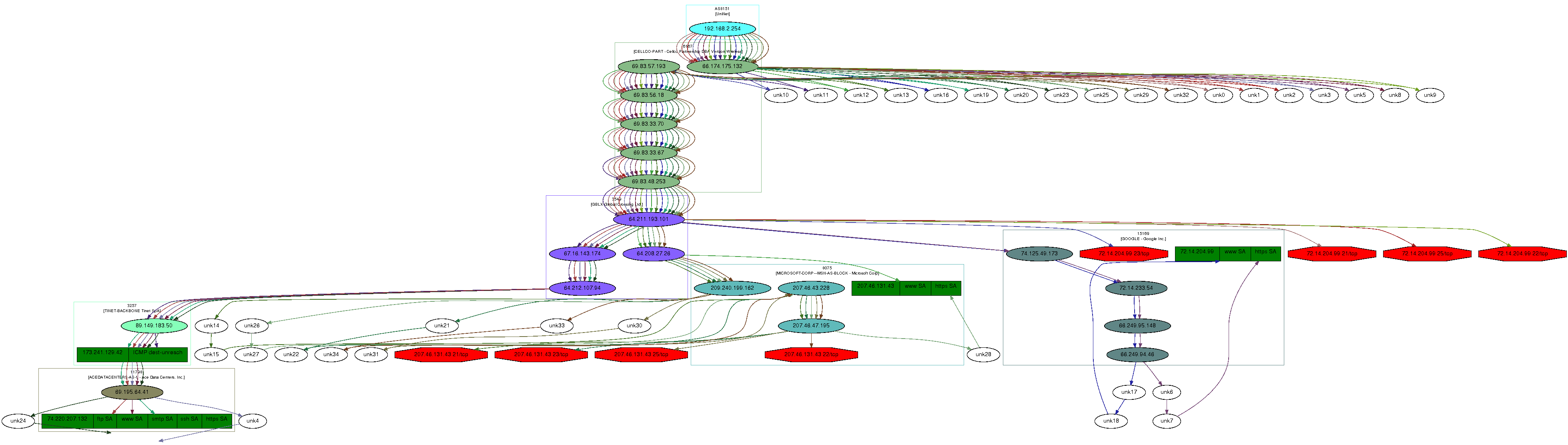
20 ae-81-81.csw3.NewYork1.Level3.net (4.69.134.74) 132.453 ms 121.833 ms 140.402 ms

21 ae-3-89.edge3.NewYork1.Level3.net (4.68.16.145) 128.667 ms 125.030 ms 128.452 ms

22 BLUEHOST-IN.edge3.NewYork1.Level3.net (4.26.35.98) 186.740 ms 184.821 ms 185.135 ms

23 tg2-5.ar01.prov.acedatacenters.com (69.195.64.41) 180.102 ms 186.995 ms 178.154 ms

24 host132.hostmonster.com (74.220.207.132) [open] 179.793 ms 184.481 ms 172.905 msEV8 – SCAPY TCP traceroute results ([Task 8](#task8)).



EV9 – dig command MX query results ([Task 9](#task9)).

root@e-ubuntu:~# dig @ns1.hostmonster.com -t MX jedge.com

; <<>> DiG 9.7.0-P1 <<>> @ns1.hostmonster.com -t MX jedge.com

; (1 server found)

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14199

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:

;jedge.com. IN MX

;; ANSWER SECTION:

jedge.com. 14400 IN MX 0 jedge.com.

;; ADDITIONAL SECTION:

jedge.com. 14400 IN A 74.220.207.132

;; Query time: 189 msec

;; SERVER: 74.220.195.131#53(74.220.195.131)

;; WHEN: Wed Feb 23 08:38:28 2011

;; MSG SIZE rcvd: 59 EV10 – dictionary attack DNS server looking for hosts with common names using dnsmap ([Task 10](#task10)).

root@e-ubuntu:~/tools/dnsmap-0.30# ./dnsmap agency.state.xx.us -w words.txt

dnsmap 0.30 - DNS Network Mapper by pagvac (gnucitizen.org)

[+] searching (sub)domains for agency.state.xx.us using words.txt

[+] using maximum random delay of 10 millisecond(s) between requests

dns1.agency.state.xx.us

IP address #1: 170.3.245.245

dns2.agency.state.xx.us

IP address #1: 170.3.245.246

www.agency.state.xx.us

IP address #1: 170.3.245.30

www1.agency.state.xx.us

IP address #1: 170.3.245.54

www3.agency.state.xx.us

IP address #1: 170.3.8.21

www4.agency.state.xx.us

IP address #1: 170.3.245.14

[+] 6 (sub)domains and 6 IP address(es) found

[+] completion time: 7 second(s)

EV11a – results from reverse lookup using perl script dnsenum.pl ([Task 11](#task11))

----------------------------------------------------

Performing reverse lookup on 1024 ip addresses:

----------------------------------------------------

4.8.168.192.in-addr.arpa. 10800 IN PTR www.agency.state.xx.us.

5.8.168.192.in-addr.arpa. 10800 IN PTR www.agency.state.xx.us.

9.8.168.192.in-addr.arpa. 10800 IN PTR batman.agency.state.xx.us.

10.8.168.192.in-addr.arpa. 10800 IN PTR robin.agency.state.xx.us.

18.8.168.192.in-addr.arpa. 10800 IN PTR mail3.agency.state.xx.us.

19.8.168.192.in-addr.arpa. 10800 IN PTR mail4.agency.state.xx.us.

21.8.168.192.in-addr.arpa. 10800 IN PTR www3.agency.state.xx.us.

25.8.168.192.in-addr.arpa. 10800 IN PTR ftp.agency.state.xx.us.

27.8.168.192.in-addr.arpa. 10800 IN PTR www1.agency.state.xx.us.

27.8.168.192.in-addr.arpa. 10800 IN PTR www2.agency.state.xx.us.

27.8.168.192.in-addr.arpa. 10800 IN PTR www8.agency.state.xx.us.

35.8.168.192.in-addr.arpa. 10800 IN PTR www7.agency.state.xx.us.

1.57.168.192.in-addr.arpa. 10800 IN PTR gw.agency.state.xx.us.

2.245.168.192.in-addr.arpa. 10800 IN PTR www9.agency.state.xx.us.

3.245.168.192.in-addr.arpa. 10800 IN PTR www10.agency.state.xx.us.

4.245.168.192.in-addr.arpa. 10800 IN PTR www11.agency.state.xx.us.

5.245.168.192.in-addr.arpa. 10800 IN PTR www12.agency.state.xx.us.

7.245.168.192.in-addr.arpa. 10800 IN PTR www13.agency.state.xx.us.

10.245.168.192.in-addr.arpa. 10800 IN PTR www15.agency.state.xx.us.

14.245.168.192.in-addr.arpa. 10800 IN PTR www4.agency.state.xx.us.

15.245.168.192.in-addr.arpa. 10800 IN PTR www5.agency.state.xx.us.

44.245.168.192.in-addr.arpa. 10800 IN PTR www20.agency.state.xx.us.

245.245.168.192.in-addr.arpa. 10800 IN PTR dns1.agency.state.xx.us.

246.245.168.192.in-addr.arpa. 10800 IN PTR dns2.agency.state.xx.us.

22 results out of 1024 ip addresses.

EV11b – results from reverse lookup using Nmap ([Task 11](#task11))

Nmap scan report for 192.168.0.14

Nmap scan report for 192.168.0.15

Nmap scan report for 192.168.0.16

Nmap scan report for 192.168.0.17

Nmap scan report for mail3.agency.state.xx.us (192.168.0.18)

Nmap scan report for mail4.agency.state.xx.us (192.168.0.19)

Nmap scan report for 192.168.0.20

Nmap scan report for www3.agency.state.xx.us (192.168.0.21)

Nmap scan report for 192.168.0.22

Nmap scan report for 192.168.0.23

Nmap scan report for 192.168.0.24

Nmap scan report for ftp.agency.state.xx.us (192.168.0.25)

Nmap scan report for 192.168.0.26

Nmap scan report for www8.agency.state.xx.us (192.168.0.27)

Nmap scan report for 192.168.0.28

Nmap scan report for 192.168.0.29

Nmap scan report for 192.168.0.30

Nmap scan report for 192.168.0.31

Nmap scan report for 192.168.0.32

Nmap scan report for 192.168.0.33

Nmap scan report for 192.168.0.34

Nmap scan report for www7.agency.state.xx.us (192.168.0.35)

Nmap scan report for 192.168.0.36

Nmap scan report for 192.168.0.37

Nmap scan report for 192.168.0.38

EV12 – dnswalk sample results ([Task 12](#task12))

root@e-ubuntu:~/tools/dnswalk# ./dnswalk agency.state.xx.us.

Checking agency.state.xx.us.

Getting zone transfer of agency.state.xx.us. from dns1.agency.state.xx.us...failed

FAIL: Zone transfer of agency.state.xx.us. from dns1.agency.state.xx.us failed: connection failed

Getting zone transfer of agency.state.xx.us. from dns2.agency.state.xx.us...failed

FAIL: Zone transfer of agency.state.xx.us. from dns2.agency.state.xx.us failed: connection failed

Getting zone transfer of agency.state.xx.us. from xxxx.xx.xxxx.xx.att.net...done.

SOA=dns1.agency.state.xx.us contact=sleddick.agency.state.xx.us

WARN: agency.state.xx.us A 192.168.245.30: no PTR record

WARN: autodiscover.agency.state.xx.us CNAME adredirect.nysemail.nyenet: unknown host

WARN: ldap.agency.state.xx.us A 192.168.62.6: no PTR record

WARN: smartnet.agency.state.xx.us A 192.168.8.15: no PTR record

WARN: www.agency.state.xx.us A 192.168.245.30: no PTR record

WARN: www1.agency.state.xx.us A 192.168.245.54: no PTR record

WARN: www14.agency.state.xx.us A 192.168.245.12: no PTR record

WARN: www19.agency.state.xx.us A 192.168.245.9: no PTR record

WARN: www21.agency.state.xx.us A 192.168.245.43: no PTR record

2 failures, 9 warnings, 0 errors.

EV13 – Zone transfer results ([Task 13](#task13))

; <<>> DiG 9.7.0-P1 <<>> @xxx.xx.xx.xxxxxx.att.net -t AXFR agency.state.xx.us

; (1 server found)

;; global options: +cmd

agency.state.xx.us. 10800 IN SOA dns1.agency.state.xx.us. sleddick.agency.state.xx.us. 124 10800 3600 432000 86400

agency.state.xx.us. 10800 IN TXT "v=spf1 a:batman.agency.state.xx.us mx include:xxxxxx.state.xx.us ~all"

agency.state.xx.us. 10800 IN A 192.168.245.30

agency.state.xx.us. 600 IN MX 10 mail.xxxxxxxx.xxxxxxxxxx.com.

agency.state.xx.us. 10800 IN NS xxx.xx.xx.xxxxxx.att.net.

agency.state.xx.us. 10800 IN NS xxx.xx.xx.xxxxxx.att.net.

agency.state.xx.us. 10800 IN NS dns1.agency.state.xx.us.

agency.state.xx.us. 10800 IN NS dns2.agency.state.xx.us.

batman.agency.state.xx.us. 10800 IN A 192.168.8.9

dns1.agency.state.xx.us. 10800 IN A 192.168.245.245

dns2.agency.state.xx.us. 10800 IN A 192.168.245.246

web1.agency.state.xx.us. 10800 IN A 192.168.8.4

web2.agency.state.xx.us. 10800 IN A 192.168.8.5

ftp.agency.state.xx.us. 10800 IN A 192.168.8.25

gw.agency.state.xx.us. 10800 IN A 192.168.57.1

ldap.agency.state.xx.us. 10800 IN A 192.168.62.6

lyris.agency.state.xx.us. 10800 IN A 192.168.8.253

mail3.agency.state.xx.us. 10800 IN A 192.168.8.18

mail4.agency.state.xx.us. 10800 IN A 192.168.8.19

www4.agency.state.xx.us. 10800 IN A 192.168.245.14

www5.agency.state.xx.us. 10800 IN A 192.168.245.15

www7.agency.state.xx.us. 10800 IN A 192.168.8.35

www8.agency.state.xx.us. 10800 IN A 192.168.8.27

agency.state.xx.us. 10800 IN SOA dns1.agency.state.xx.us. sleddick.agency.state.xx.us. 124 10800 3600 432000 86400

;; Query time: 142 msec

;; SERVER: xxx.xxx.128.106#53(xxx.xxx.128.106)

;; WHEN: Wed Feb 23 09:04:27 2011

;; XFR size: 37 records (messages 1, bytes 1020)

EV14 – Banner grabbing results ([Task 14](#task14))

root@bt:~# echo "" | nc -v -n -w1 74.220.207.132 21-23

(UNKNOWN) [74.220.207.132] 23 (telnet) : Connection timed out

(UNKNOWN) [74.220.207.132] 22 (ssh) open

SSH-2.0-OpenSSH\_5.5

Protocol mismatch.

(UNKNOWN) [74.220.207.132] 21 (ftp) open

220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------

220-You are user number 6 of 1000 allowed.

220-Local time is now 19:15. Server port: 21.

220-This is a private system - No anonymous login

220-IPv6 connections are also welcome on this server.

220 You will be disconnected after 15 minutes of inactivity.

500 ?

root@bt:~# (echo HEAD / HTTP/1.0; echo; ) | nc www.microsoft.com 80

HTTP/1.1 200 OK

Cache-Control: no-cache

Content-Length: 1020

Content-Type: text/html

Last-Modified: Mon, 16 Mar 2009 20:35:26 GMT

Accept-Ranges: bytes

ETag: "67991fbd76a6c91:0"

Server: Microsoft-IIS/7.5

VTag: 438572940500000000

P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI TELo OUR SAMo CNT COM INT NAV ONL PHY PRE PUR UNI"

X-Powered-By: ASP.NET

Date: Fri, 25 Feb 2011 02:13:21 GMT

Connection: keep-alive

root@e-ubuntu:~# (echo HEAD / HTTP/1.0; echo; ) | stunnel -c -r www.google.com:443

HTTP/1.0 302 Found

Location: https://encrypted.google.com/

Cache-Control: private

Content-Type: text/html; charset=UTF-8

Set-Cookie: PREF=ID=de532827bb23a262:FF=0:TM=1298600075:LM=1298600075:S=\_a8OugcwR2uNgvvg; expires=Sun, 24-Feb-2013 02:14:35 GMT; path=/; domain=.google.com

Set-Cookie: NID=44=NWggLgtBWc7uPUYXm1lDZiLaBUpKu3J4JRbf7xW6tSmw41LQnW3J0d1JxWHpuFhxn0L2FA--q9Y7yB30XI9BLXAOPbCBnUmUNg5Jr5JxFKRcc8RT3wX0FZut643eBgBW; expires=Sat, 27-Aug-2011 02:14:35 GMT; path=/; domain=.google.com; HttpOnly

Date: Fri, 25 Feb 2011 02:14:35 GMT

Server: gws

Content-Length: 226

X-XSS-Protection: 1; mode=block

**Appendix A: Installation Help**

### Windows DIG installation

dig is the standard tool for advanced DNS queries. A Windows version is available as part of the BIND port. To install it on Windows:

* Go to <ftp://ftp.isc.org/isc/bind9/>
* Download the latest version of BIND (in Zip format)
* Open the archive in Windows
* Extract dig.exe and \*.dll to c:\Tools\dig
* From the Windows Command Prompt change to the c:\Tools\dig directory and run dig

If you want the documentation page, extract dig.html to somewhere that you can find it.

Now you will be able to use dig from your command prompt in Windows. It is faster and more sophisticated than nslookup.

Get the quick help options with "dig -h".

### Linux SCAPY installation

## Ubuntu 10.04 LTS

$sudo apt-get install scapy python-pygraphviz python-pythonmagick python-pyx python-gnuplot

accept all dependencies

## RHEL5 and CentOS-5 (python 2.4)

$wget <http://packages.sw.be/scapy/scapy-1.0.5-1.el5.rf.noarch.rpm>

$sudo rpm -ivh scapy-1.0.5-1.el5.rf.noarch.rpm

$sudo yum install gnuplot gd tcpdump libpcap ImageMagick

$sudo ln -s /usr/sbin/tcpdump /usr/bin/tcpdump

**Install Python PyX**

$wget <http://optusnet.dl.sourceforge.net/sourceforge/pyx/PyX-0.10.tar.gz>

$tar xvzf PyX-0.10.tar.gz

$cd PyX-0.10

$python setup.py build

$sudo python setup.py install

**Install Gnuplot Python Wrapper**

$wget http://cdnetworks-us-2.dl.sourceforge.net/project/numpy/NumPy/1.5.1/numpy-1.5.1.tar.gz

$tar zxvf numpy-1.5.1.tar.gz

$cd numpy-1.5.1

$python setup.py build

$sudo python setup.py install

$cd ..

$wget http://cdnetworks-us-2.dl.sourceforge.net/project/gnuplot-py/Gnuplot-py/1.8/gnuplot-py-1.8.zip

$unzip gnuplot-py-1.8.zip

$cd gnuplot-py-1.8

$python setup.py build

$sudo python setup.py install

**Install Python Crypto Wrapper**

$wget http://ftp.dlitz.net/pub/dlitz/crypto/pycrypto/pycrypto-2.3.tar.gz

$tar zxvf pycrypto-2.3.tar.gz

$cd pycrypto-2.3

$python setup.py build

$sudo python setup.py install

$wget http://www.secdev.org/projects/scapy/files/ethertypes

$sudo mv ethertypes /etc

### Linux dnswalk installation (Ubuntu 10.4 LTS)

apt-get install libnet-dns-perl libdigest-hmac-perl libdigest-sha1-perl libnet-ip-perl

mkdir –p ~/tools/dnswalk

wget http://cdnetworks-us-2.dl.sourceforge.net/project/dnswalk/dnswalk/2.0.2/dnswalk-2.0.2.tar.gz

tar zxvf dnswalk-2.0.2.tar.gz

rm dnswalk-2.0.2.tar.gz

### Linux dnsenum installation (Ubuntu 10.4 LTS)

$sudo apt-get install libnet-netmask-perl libxml-writer-perl libnet-whois-raw-perl libcarp-assert-more-perl libcarp-assert-perl libhttp-server-simple-perl libio-socket-ssl-perl libnet-libidn-perl libnet-ssleay-perl libtest-longstring-perl libwww-mechanize-perl libnet-xwhois-perl libnet-whois-ripe-perl

$mkdir ~/tools

$cd ~/tools

$wget http://dnsenum.googlecode.com/files/dnsenum1.2.1.tar.gz

$tar zxvf dnsenum1.2.1.tar.gz

$cd dnsenum1.2.1

$wget http://dnsenum.googlecode.com/files/dnsbig.txt

### Linux dnsmap installation (Ubuntu 10.4 LTS)

$mkdir ~/tools

$cd ~/tools

$wget http://dnsmap.googlecode.com/files/dnsmap-0.30.tar.gz

$tar zxvf dnsmap-0.30.tar.gz

$cd dnsmap-0.30

$make

### Linux stunnel 3 installation (Ubuntu 10.4 LTS)

When stunnel 4.0 was released, the entire interface changed from where you can type all the details on the command line to one where all the details must be placed within a configuration file. This will not work for the purposes we need. Ubuntu only offers stunnel4. Instructions below will get the latest version of Stunnel 3 up and running.

Download the latest stunnel version 3

http://www.stunnel.org/download/stunnel/src/stunnel-3.26.tar.gz

$wget ftp://ftp.stunnel.org/stunnel/obsolete/3.x/stunnel-3.22.tar.gz

$tar zxvf stunnel-3.22.tar.gz

$cd stunnel-3.22

$./configure --prefix=/usr --bindir=/usr/bin --sbindir=/usr/bin

$make

When asked enter the following information (or whatever you agency information is)

Country Name (2 letter code) [PL]:US

State or Province Name (full name) [Some-State]:Georgia

Locality Name (eg, city) []:Atlanta

Organization Name (eg, company) [Stunnel Developers Ltd]:DOAA

Organizational Unit Name (eg, section) []:ISAAS

Common Name (FQDN of your server) [localhost]:audits.state.ga.us

$sudo make install

## Installing NMAP

$mkdir ~/source

$cd ~/source

$wget http://nmap.org/dist/nmap-5.51.tar.bz2

$tar jxvf nmap-5.51.tar.bz2

$cd nmap-5.51

$./configure

$make

$sudo make install

Appendix B – SCAPY results

Incorrect Results: your router is not passing the trace packets back to your workstation.

>>> res,unans = traceroute(["www.jedge.com","www.google.com","www.microsoft.com"],dport=[21,22,23,25,80,443],maxttl=20,retry=-2)

Begin emission:

\*\*\*\*\*\*Finished to send 360 packets.

Begin emission:

\*\*\*...............................Finished to send 354 packets.

Begin emission:

\*\*\*...................\*..\*....................\*..............................\*............................Finished to send 351 packets.

...........Begin emission:

\*\*\*Finished to send 344 packets.

Begin emission:

\*\*Finished to send 341 packets.

Begin emission:

\*\*Finished to send 339 packets.

Begin emission:

Finished to send 337 packets.

Begin emission:

Finished to send 337 packets.

Received 164 packets, got 23 answers, remaining 337 packets

209.85.157.99:tcp21 209.85.157.99:tcp22 209.85.157.99:tcp23 209.85.157.99:tcp25 209.85.157.99:tcp443 209.85.157.99:tcp80 65.55.12.249:tcp21 65.55.12.249:tcp22 65.55.12.249:tcp23 65.55.12.249:tcp25 65.55.12.249:tcp443 65.55.12.249:tcp80 74.220.207.132:tcp21 74.220.207.132:tcp22 74.220.207.132:tcp23 74.220.207.132:tcp25 74.220.207.132:tcp443 74.220.207.132:tcp80

1 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11 192.168.0.1 11

17 - - - - 209.85.157.99 SA - - - - - - - - - - - - -

19 - - - - 209.85.157.99 SA 209.85.157.99 SA - - - - - - - - - - - -

20 - - - - 209.85.157.99 SA 209.85.157.99 SA - - - - - - - - - - - -

The highlighted text is the router separating the workstation from the internet. If your results show this then the test did not work.

* [Link](http://www.jedge.com/docs/scapy_1.txt) to a text version of the output.
* [Link](http://www.jedge.com/images/scapy_1.png) to a screenshot.

Correct Results: workstation is connected directly to the internet.

>>> res,unans = traceroute(["www.jedge.com","www.google.com","www.microsoft.com"],dport=[21,22,23,25,80,443],maxttl=20,retry=-2)

Begin emission:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Finished to send 360 packets.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Begin emission:

\*\*\*\*\*\*\*Finished to send 148 packets.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Begin emission:

\*\*\*Finished to send 113 packets.

Begin emission:

\*\*Finished to send 110 packets.

\*Begin emission:

\*\*Finished to send 107 packets.

Begin emission:

\*\*Finished to send 105 packets.

\*Begin emission:

Finished to send 102 packets.

\*\*Begin emission:

\*Finished to send 100 packets.

Begin emission:

Finished to send 99 packets.

\*Begin emission:

Finished to send 98 packets.

Begin emission:

Finished to send 98 packets.

Received 262 packets, got 262 answers, remaining 98 packets

207.46.131.43:tcp21 207.46.131.43:tcp22 207.46.131.43:tcp23 207.46.131.43:tcp25 207.46.131.43:tcp443 207.46.131.43:tcp80 72.14.204.99:tcp21 72.14.204.99:tcp22 72.14.204.99:tcp23 72.14.204.99:tcp25 72.14.204.99:tcp443 72.14.204.99:tcp80 74.220.207.132:tcp21 74.220.207.132:tcp22 74.220.207.132:tcp23 74.220.207.132:tcp25 74.220.207.132:tcp443 74.220.207.132:tcp80

1 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11 192.168.2.254 11

2 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11 66.174.175.132 11

4 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11 69.83.57.193 11

5 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11 69.83.56.18 11

If you results show all the different hops the packet traversed they you are all set!

* [Link](http://www.jedge.com/docs/scapy_2.txt) to a text document showing the results in a better format.
* [Link](http://www.jedge.com/images/scapy_2.png) to a screen shot of the results in a terminal window.