Forum of Incident Response and Security Teams

Improving Security Together

IPv6 Security Training

Resources for the IPv6 Lab

Foto: https://unsplash.com/@ibrahimboran

Author: Frank Herberg
frank.herberg@switch.ch

Document Type:	Documentation
Version:	V1.1
Created:	23.11.12
Last changes:	14.06.19
Copyright:	FIRST Inc.

Content

1 Useful Commands

1.1. Linux

Show interface configuration

```
ifconfig
ifconfig eth0
```

Show routing table

```
route -6
ip -6 route
ip -6 route get 2001:: (show route for destination 2001::)
ip -6 route list cache (show Destination Cache (Route Cache))
```

Show neighbor cache

```
ip -6 neigh show
```

Show address selection policy information

```
cat /etc/gai.conf
ip -6 addrlabel show
```

Configuring IPv6 addresses

```
ip -6 addr list / show [dev eth0]
ip -6 addr add 2001::1/64 dev eth0
ip -6 addr del 2001::1/64 dev eth0
ip -6 maddr
```

Configuring IPv6 routes

```
ip -6 route add ::/0 via 2001::2 dev eth0
ip -6 route del ::/0 via 2001::2 dev eth0
ip -6 route flush default
ip -6 route flush 2001::
```

Show neighbor discovery configuration

```
ip -6 ntable show [dev eth0]
```

Show multicast groups joined by the host

```
ip -6 maddr show
```

Configure IPv6 neighbours (IPv6 address => MAC address)

```
ip -6 neigh add fe80::3 lladdr 00:11:22:33:44:55 dev eth0
ip -6 neigh del fe80::3 lladdr 00:11:22:33:44:55 dev eth0
```

Enabling IPv6 (older versions)

```
modprobe ipv6
```

Enable/disable ipv6 routing

```
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
echo 0 > /proc/sys/net/ipv6/conf/all/forwarding
```

1.2. Windows

Show network configuration information

```
ipconfig /all
netsh interface ipv6 show interface
```

Show list of neighbors

```
netsh interface ipv6 show neighbors
```

Show configured IPv6 addresses

```
netsh interface ipv6 show address
```

Show configured IPv6 routes

```
route print -6
netsh interface ipv6 show route
(Default Route has destination ::/0)
```

Show advertised prefixes

```
netsh interface ipv6 show siteprefixes
netsh interface ipv6 show potentialrouters
```

Set IPv6 address

```
netsh interface ipv6 set address interface="LAN Connection" address=2a01::1/64
store=active/persistent
```

Set route

```
netsh interface ipv6 set route ::/0 "LAN Connection" 2a02::2 0 2 yes 5000 5000
```

Show MTU of interfaces

```
netsh interface ipv6 show subinterfaces
```

Lookup neighbor cache

```
netsh interface ipv6 show neighbors
```

Set neighbor

```
netsh interface ipv6 set neighbors interface="LAN Connection" address="fec0::2"
neighbor="12-34-56-78-9a-bc" store=active/persistent
```

Enable forwarding

```
netsh interface ipv6 set interface interface="LAN Connection" forwarding=enabled
store=active/persistent
```

Enable RA

```
netsh interface ipv6 set interface interface="LAN Connection" advertise=enable
siteprefixlength=64 advertisedefaultroute store=active/persistent
```

Register to DNS

```
netsh interface ipv6 set dnsservers name="LAN Connection" register=primary
```

Disable Pricacy Extensions (7, Vista, 2008)

```
netsh interface ipv6 set privacy state=disabled store=persistent
Reboot
```

Enable SLAAC EUI-64 address generation

```
netsh interface ipv6 set global randomizeidentifiers=disabled store=persistent
```

Status of Teredo / ISATAP / 6to4 Tunnel

```
netsh interface ipv6 show teredo
netsh interface ipv6 isatap show mode|state|router
netsh interface ipv6 6to4 show interface|relay|routing|state
```

2. Wireshark Tips

1.1 Adjust GUI

Adjust Wireshark-Fontsize: CTRL+"+"/"-"
Switch off Paket Bytes View: View Paket Bytes = off

1.2 Filter Examples

ilter	xpression
v6	`pv6`
MPv6	`cmpv6`
MPv6 Types (134 for RAs)	`cmpv6.type == 134`
thernet Type (0x86dd for IPv6)	`th.type == 0x86dd`
rotocol Type (41 for tunnels)	`p.proto == 41`
ource Address	`pv6.src == [address]`
gic operations	`nd, and not, or`

Tabelle 1

Example:

If you want to see only IPv6 Router Solicitations and -Advertisements for example, use filter string

```
icmpv6.type == 133 or icmpv6.type == 134
```

Example:

If you want to see only IPv6 RS, RA, NS, NA for example, use filter string

```
icmpv6.type >= 133 and icmpv6.type <= 136
```

3. THC IPv6 Attack Toolkit

This chapter documents „The Hackers Choice IPv6 toolkit“. It is based on the non-free version 1.9 as of March 2012. See thc.org for more info.

3.1. Overview of the Tools

Tool	Description	Example
alive6; alive26	effective alive scanning, detects all systems listening to this address – use where ping6 ff02::1 fails – because not all systems answer multicast pings, alive6 sends ICMP6 echo request and faulty ICMP6 paket, sends per default to ff02::1; see also fake_mld6	**scan ip pattern:** alive26 –F eth1 ip:ip:ip:8000-8010:0-1:0-3 **scan list of systems:** alive26 –I found.txt –o alive.txt -M –D –v –d eth0 **scan hosts behind fw:** -f **scan link local:** –l **test source routing:** alive26 eth1 target(self) bounce-over-router
covert_send6; covert_send6d	future use	
denial6	collection of denial-of-service tests	
detect-new-ip6	detect new ip6 devices which join the network, execute a script then	
detect_sniffer6	Tests if systems on the local LAN are sniffing. Works against Windows, Linux, OS/X and *BSD.	detect_sniffer6 eth1
dnsdict6	parallized dns ipv6 dictionary bruteforcer	dnsdict6 –l –d berkeley.edu
dos-new-ip6	detect new ip6 devices and tell them that their chosen IP collides on the network (DAD DOS)	dos-new-ip6 eth1
dump_router6	dump all local routers and their information	
exploit6	known ipv6 vulnerabilities to test against a target	
fake_advertise6	fake Neighbor Advertisement	
fake_dhcps6	Rogue DHCPv6 server. Use to configure an address and set a DNS server.	
fake_dns6d	Fake DNS server that serves the same ipv6 address to any lookup request.	
fake_dnsupdate6	Update DNS entries of other clients	fake_dnsupdate6 <server> <client-entry> <ip6-addr>
fake_mipv6	steal a mobile IP to yours if IPSEC is not	

	needed for authentication	
`fake_mld6; fake_mld26`	add or remove entries in a multicast group of your choice on the net; 26: same but for MLDv2	`fake_mld6 eth1 query` to get all link local hosts on the local network (reliably)
`fake_mldrouter6`	fake MLD router messages	
`fake_router6; fake_router26`	announce yourself as a router on the network, with the highest priority, advertise prefixes and DNS servers. *with RA Guard Evasion options (add headers)*	`fake_router6 eth1 2004::/48` will add global addresses to all systems (see ifconfig)
`flood_advertise6`	flood a target with random neighbor advertisements	
`flood_dhcpc6`	stalls a DHCP6 service and depletes its pool	
`flood_mld6; flood_mld26`	flood multicast join/leave messages	
`flood_mldrouter6`	flood multicast router announcements	
`flood_router6; flood_router26`	flood a target with random router advertisements, sends multiple prefixes in one RA, will crash Windows systems and Juniper Netscreen; additional features to bypass RA-Guard (multiple extension headers)	`flood_router26 eth1`
`flood_solicitate6`	flood ICMPv6 neighbour solicitations	
`fragmentation6`	tries various fragmentation implementation checks	
`fuzz_ip6`	fuzzes various IPv6 packets and headers	
`implementation6`	performs various implementation checks on ipv6	
`implementation6d`	listen daemon for implementation6 to check behind a FW	
`kill_router6`	Announce (to ff02:1) that a router is going down (RA with Router Lifetime 0) to delete it from the routing tables. Using '*' as router-address, this tool will sniff the network for RAs and immediately send a kill packet.	`kill_router6 eth1 '*'`
`ndpexhaust6`	exhaust memory by flooding with non-existent targets	
`node_query6`	sends an ICMPv6 node query request to the	`node_query6 <INTERFACE>`

	target and dumps the reply (only some OS)	<TARGET>
parasite6	ND spoofer, puts you as man-in-the-middle same as ARP MITM (MITM if you forward, otherwise DOS)	parasite6 –lR eth1 <fake-mac>
randicmp6	send all ICMP types+codes	
redir6	redirect traffic to you intelligently (man-in-the-middle) through spoofing of ICMP6 redirects	only works on one target
rsmurf6	remote smurfer (ping with a fake sender addr to multicast addr) to DOS local LAN (only linux at the moment)	rsmurf eth1 ff02::1
sendpees6	DOS SeND implementations , generates a neighbor solicitation requests with a lot of CGAs (crypto stuff ;-) to keep the CPU busy	
sendpeesmp6	DOS SeND implementations	
smurf6	local smurfer, ping-flood smurf a target	
thcping6	sends a hand crafted ping6 packet	
toobig6	decrease mtu (min 1280), same intelligence as redir6	toobig6 eth0 victim dst 1280
trace6	very fast traceroute6 with supports ICMP6 echo request and TCP-SYN	-t trys to detect tunnels

3.2. The THC IPv6 Tools

3.2.1. alive26

Shows alive addresses in the segment. If you specify a remote router, the packets are sent with a routing header prefixed by fragmentation.
Syntax: alive26 [-i file] [-o file] [-DM] [-p] [-F] [-e opt] [-s port,..] [-a port,..] [-u port,..] [-W TIME] [-dlrvS] interface [unicast-or-multicast-address [remote-router]]
Options:

```
-i file     check systems from input file
-o file     write results to output file
-M          enumerate hardware addresses (MAC) from input addresses (slow!)
-D          enumerate DHCP address space from input addresses
-p          send a ping packet for alive check (default)
-e dst,hop  send an errornous packets: destination (default), hop-by-hop
-s port,port,..  TCP-SYN packet to ports for alive check
-a port,port,..  TCP-ACK packet to ports for alive check
-u port,port,..  UDP packet to ports for alive check
-d          DNS resolve alive ipv6 addresses
-n number   how often to send each packet (default: local 1, remote 2)
-W time     time in ms to wait after sending a packet (default: 1)
-S          slow mode, get best router for each remote target or when proxy-NA
-l          use link-local address instead of global address
-v          verbose (twice: detailed information, thrice: dumping all packets)
```

Target address on command line or in input file can include ranges in the form

of ff02::3-fff or ff0e::0-ffff:0:0-ffff, etc.

3.2.2. alive6

Shows alive addresses in the segment. If you specify a remote router, the packets are sent with a routing header prefixed by fragmentation.
Syntax: alive6 [-dlmrS] [-W TIME] [-i FILE] [-o FILE] [-s NUMBER] interface [unicast-or-multicast-address [remote-router]]
Options:

```
-i FILE      check systems from input file
-o FILE      write results to output file
-m           enumerate from hardware addresses in input fule
-l           use link-local address instead of global address
-d           resolve alive ipv6 addresses
-W TIME      time in ms to wait after sending a packet (default: 10)
-S           slow mode, get best router for each remote target or when proxy-NA
-n NUMBER how often to send each packet (default: 1)
-s NUMBER scan type, bit-wise add: 1-ping, 2-invalid header,
            4-invalid hop-by-hop, 8-udp dns, 16-tcp ack highport,
            32-tcp syn ssh, 64-tcp syn web, 128-tcp syn ssl; default: 5
```

3.2.3. denial6

Performs various denial of service attacks on a target. If a system is vulnerable, it can crash or be under heavy load, so be careful! If no test-case-number is supplied, the list of available test-cases will be shown.
Syntax: denial6 interface destination test-case-number

3.2.4. detect-new-ip6

This tool detects new ipv6 addresses joining the local network. If a script is supplied, it is executed with the detected IPv6 address.
Syntax: detect-new-ip6 interface [script]

3.2.5. detect_sniffer6

Tests if systems on the local LAN are sniffing. Works against Windows, Linux, OS/X and *BSD. If no target is given, the link-local-all-nodes address is used.
Syntax: detect_sniffer6 interface [target6]

3.2.6. dnsdict6

Enumerates a domain for DNS entries, it uses a dictionary file if supplied or a built-in list otherwise. This tool is based on dnsmap by gnucitizen.org.
Syntax: dnsdict6 [-d46] [-s|-m|-l|-x] [-t THREADS] [-D] domain [dictionary-file]
Options:

```
-4       also dump IPv4 addresses
-t NO    specify the number of threads to use (default: 8, max: 32).
-D       dump the selected built-in wordlist, no scanning.
-d       display IPv6 information on NS and MX DNS domain information.
-[smlx] choose the dictionary size by -s(mall=50), -m(edium=796) (DEFAULT)
          -l(arge=1416), or -x(treme=3211)
```

3.2.7. dnsdictalive.sh

Performs dns bruteforcing and alive scanning on the domain. If the targets are behind a firewall, use the -F option
Syntax: /usr/local/bin/dnsdictalive.sh [-F] [-t|-T] [-u] domain.com

3.2.8. dos_mld.sh

If specified, the multicast address of the target will be dropped first.
All multicast traffic will cease after a while.
Specify -2 to use MLDv2.
Syntax: /usr/local/bin/dos_mld.sh [-2] interface [target-link-local-address multicast-address]

3.2.9. dos-new-ip6

This tools prevents new ipv6 interfaces to come up, by sending answers to duplicate ip6 checks (DAD). This results in a DOS for new ipv6 devices.
Syntax: dos-new-ip6 interface

3.2.10. dump_router6

Dumps all local routers and their information.
Syntax: dump_router6 interface

3.2.11. exploit6

Performs exploits of various CVE known IPv6 vulnerabilities on the destination
Note that for exploitable overflows only 'AAA...' strings are used.
If a system is vulnerable, it will crash, so be careful!
Syntax: exploit6 interface destination [test-case-number]

3.2.12. fake_advertise6

Advertise ipv6 address on the network (with own mac if not defined) sending it to the all-nodes multicast address if no target specified. Option -H adds a hop-by-hop header, -F a one shot fragment header, -D adds a large destination header which fragments the packet.
Syntax: fake_advertise6 [-DHF] interface ip-address-advertised [target-address [mac-address-advertised [source-ip-address]]]

3.2.13. fake_dhcps6

Fake DHCPv6 server. Use to configure an address and set a DNS server.
Syntax: fake_dhcps6 interface network-address/prefix-length dns-server [dhcp-server-ip-address [mac-address]]

3.2.14. fake_dns6d

Fake DNS server that serves the same ipv6 address to any lookup request. You can use this together with parasite6 if clients have a fixed DNS server.
Note: very simple server. Does not honor multiple queries in a packet, nor NS, MX, etc. lookups.
Syntax: fake_dns6d interface ipv6-address [fake-ipv6-address [fake-mac]]

3.2.15. fake_dnsupdate6

Syntax: fake_dnsupdate6 dns-server full-qualified-host-dns-name ipv6address
Example: fake_dnsupdate6 dns.test.com myhost.sub.test.com ::1

3.2.16. fake_mipv6

If the mobile IPv6 home-agent is mis-configured to accept MIPV6 updates without IPSEC, this will redirect all packets for home-address to care-of-address.
Syntax: fake_mipv6 interface home-address home-agent-address care-of-address

3.2.17. fake_mld26 / fake_mld6

Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your choice. Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
fake_mld**2**6 uses the MLDv**2** protocol. Only a subset of what the protocol is able to do is possible to implement via a command line. Code it if you need something.
Syntax: fake_mld6| fake_mld26 [-l] interface add|delete|query [multicast-address [target-address [ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

3.2.18. fake_mldrouter6

Announce, delete or soliciate MLD router - yourself or others.
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
Syntax: fake_mldrouter6 [-l] interface advertise|solicitate|terminate [own-ip [own-mac-address]]

3.2.19. fake_router26

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
Syntax: fake_router26 [-E type] [-A network/prefix] [-R network/prefix] [-D dns-server] [-s sourceip] [-S sourcemac] [-ardl seconds] [-Tt ms] interface
Options:

```
-A network/prefix add autoconfiguration network (up to 16 times)
-a seconds          valid lifetime of prefix -A (defaults to 99999)
-R network/prefix   add a route entry (up to 16 times)
-r seconds          route entry lifetime of -R (defaults to 4096)
-D dns-server       specify a DNS server (up to 16 times)
-d seconds          dns entry lifetime of -D (defaults to 4096
-M mtu              the MTU to send, defaults to the interface setting
-s sourceip         the source ip of the router, defaults to your link local
-S sourcemac        the source mac of the router, defaults to your interface
-l seconds          router lifetime (defaults to 2048)
-T ms               reachable timer (defaults to 0)
-t ms               retrans timer (defaults to 0)
-E type             Router Advertisement Guard Evasion option. Types:
           H            simple hop-by-hop header
           1            simple one-shot fragment. hdr. (can add multiple)
           D            insert a large destin. hdr. so that it fragments
                    Examples: -E H111, -E D
```

3.2.20. fake_router6

Announce yourself as a router and try to become the default router. If a non-existing link-local or mac address is supplied, this results in a DOS. Option -H adds hop-by-hop, -F fragmentation header and -D dst header.

Syntax: fake_router6 [-HFD] interface network-address/prefix-length [dns-server [router-ip-link-local [mtu [mac-address]]]]

3.2.21. flood_advertise6

Flood the local network with neighbor advertisements.
Syntax: flood_advertise6 interface

3.2.22. flood_dhcpc6

DHCP client flooder. Use to deplete the IP address pool a DHCP6 server is offering. Note: if the pool is very large, this is rather senseless.
By default the link-local IP MAC address is random, however this won't work
in some circumstances. -n will use the real MAC, -N the real MAC and
link-local address. -1 will only solicate an address but not request it.
If -N is not used, you should run parasite6 in parallel.
Use -d to force DNS updates, you can specify a domain name on the commandline.
Syntax: flood_dhcpc6 [-n|-N] [-1] [-d] interface [domain-name]

3.2.23. flood_mld26

Flood the local network with MLDv2 reports.
Syntax: flood_mld26 interface

3.2.24. flood_mld6

Flood the local network with MLD reports.
Syntax: flood_mld6 interface

3.2.25. flood_mldrouter6

Flood the local network with MLD router advertisements.
Syntax: flood_mldrouter6 interface

3.2.26. flood_router26

Flood the local network with router advertisements. Each packet contains 17 prefix and route enries -F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
Syntax: flood_router26 [-HFD] interface

3.2.27. flood_router6

Flood the local network with router advertisements.
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
Syntax: flood_router6 [-HFD] interface

3.2.28. flood_solicitate6

Flood the network with neighbor solicitations.
Syntax: flood_solicitate6 interface [target]

3.2.29. fragmentation6

Performs fragment firewall and implementation checks, incl. denial-of-service.

-f activates flooding mode, no pauses between sends; -p disables first and final pings, -n number specifies how often each test is performed
Syntax: fragmentation6 [-fp] [-n number] interface destination [test-case-no]

3.2.30. fuzz_ip6

Fuzzes an icmp6 packet
Syntax: fuzz_ip6 [-x] [-t number | -T number] [-p number] [-IFSDHRJ] [-1|-2|-3|-4|-5|-6|-7] interface unicast-or-multicast-address [address-in-data-pkt]
Options:

```
 -1          fuzz ICMP6 echo request (default)
 -2          fuzz ICMP6 neighbor solicitation
 -3          fuzz ICMP6 neighbor advertisement
 -4          fuzz ICMP6 router advertisement
 -5          fuzz multicast listener report packet
 -6          fuzz multicast listener done packet
 -7          fuzz multicast listener query packet
 -8          fuzz multicast listener v2 report packet
 -9          fuzz multicast listener v2 query packet
 -x          tries all 256 values for flag and byte types
 -t number   continue from test no. number
 -T number   only performs test no. number
 -p number   perform an alive check every number of tests (default: none)
 -n number   how many times to send each packet (default: 1)
 -I          fuzz the IP + ICMP header too
 -F          add one-shot fragmentation, and fuzz it too (for 1)
 -S          add source-routing, and fuzz it too (for 1)
 -D          add destination header, and fuzz it too (for 1)
 -H          add hop-by-hop header, and fuzz it too (for 1 and 5-9)
 -R          add router alert header, and fuzz it too (for 5-9 and all)
 -J          add jumbo packet header, and fuzz it too (for 1)
You can only define one of -1 ... -7, defaults to -1.
```

3.2.31. implementation6

Performs some ipv6 implementation checks, can be used to test firewalls too.
Syntax: implementation6 interface destination [test-case-number]

3.2.32. implementation6d

Identifies test packets by the implementation6 tool, useful to check what packets passed a firewall.
Syntax: implementation6d interface

3.2.33. kill_router6

Announce that a target a router going down to delete it from the routing tables.
If you supply a '*' as router-address, this tool will sniff the network for RAs and immdedeantely send the kill packet.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
Syntax: kill_router6 [-HFD] interface router-address [srcmac [dstmac]]

3.2.34. ndpexhaust6

Randomly pings IPs in target network
Syntax: ndpexhaust6 interface destination-network [sourceip]

3.2.35. parasite6

This is an "ARP spoofer" for IPv6, redirecting all local traffic to your own
system (or nirvana if fake-mac does not exist) by answering falsely to
Neighbor Solitication requests
Option -l loops and resends the packets per target every 5 seconds.
Option -R will also try to inject the destination of the solicitation
NS security bypass: -F fragment, -H hop-by-hop and -D large destination header
Syntax: parasite6 [-lRFHD] interface [fake-mac]

3.2.36. randicmp6

Sends all ICMPv6 type and code combinations to destination.
Set source ipv6 address to spoof.
Syntax: randicmp6 interface destination [source]

3.2.37. redir6

Implant a route into src-ip, which redirects all traffic to target-ip to
new-ip. You must know the router which would handle the route.
If the new-router-mac does not exist, this results in a DOS.
If the TTL of the target is not 64, then specify this is the last option.
Syntax: redir6 interface src-ip target-ip original-router new-router [new-router-mac] [hop-limit]

3.2.38. rsmurf6

Smurfs the local network of the victim. Note: this depends on an
implementation error, currently only verified on Linux.
Evil: "ff02::1" as victim will DOS your local LAN completely
Syntax: rsmurf6 interface victim-ip

3.2.39. sendpees6

Send SEND neighbor solicitation messages and make target to verify a Iota CGA and RSA signatures
Syntax: sendpees6 <inf> <key_length> <prefix> <victim>

3.2.40. sendpeesmp6

Send SEND neighbor solicitation messages and make target to verify a Iota CGA and RSA signatures
Syntax: sendpeesmp6 <inferface> <key_length> <prefix> <victim>
Example: sendpeesmp6 eth0 2048 fe80:: fe80::1

3.2.41. smurf6

Smurf the target with icmp echo replies. Target of echo request is the
local all-nodes multicast address if not specified
Syntax: smurf6 interface victim-ip [multicast-network-address]

3.2.42. thcping6

Craft your special icmpv6 echo request packet.

You can put an "x" into src6, srcmac and dstmac for an automatic value.
Syntax: thcping6 [-af] [-H o:s:v] [-D o:s:v] [-F dst] [-t ttl] [-c class] [-l label] [-d size] interface src6 dst6 [srcmac [dstmac [data]]]
Options:

```
  -a               add a hop-by-hop header with router alert option.
  -H o:s:v         add a hop-by-hop header with special content
  -D o:s:v         add a destination header with special content
  -f               add a one-shot fragementation header
  -F ipv6address   use source routing to this final destination
  -t ttl           specify TTL (default: 64)
  -c class         specify a class (0-4095)
  -l label         specify a label (0-1048575)
  -d data sizedefine the size of the ping data buffer
o:s:v Syntax: option-no:size:value, value is in hex, e.g. 1:2:feab
```

3.2.43. toobig6

Implants the specified mtu on the target.
If the TTL of the target is not 64, then specify this is the last option.
Syntax: toobig6 interface target-ip existing-ip mtu [hop-limit]

3.2.44. trace6

A basic but very fast traceroute6 program.
If no port is specified, ICMP6 ping requests are used, otherwise TCP SYN packets to the specified port. -d resolves the ip6 addresses to DNS.
Option -t enables tunnel detection mechanism, -s src6 specifies the source ip6
Maximum hop reach: 24
Syntax: trace6 [-d] [-t] [-s src6] interface targetaddress [port]

2. Appendix

2.1 Address Types

see RFC 4291 (IP Version 6 Addressing Architecture) and RFC 5156 (Special-Use IPv6 Addresses)

Address Type	Range	Alternative representation or Example	Remark
Unspecified Address	0:0:0:0:0:0:0:0	::	DAD source addr
Loopback Address	0:0:0:0:0:0:0:1	::1	
IPv4-mapped IPv6 address	::FFFF:ddd.ddd.ddd.dd	0000:0000:0000:0000:0000:FFFF:129.144.52.38	(::d.d.d.d deprecta in RFC 4291)
Link-Local unicast	FE80::/64	FE80:0000:0000:0000 + Interface ID	mandatory, local (subnet)
Unique Local Unicast addresses (ULA)	FC00::/7 FD00::/7	can be used for local net like 192.168.x.x, *Global-I* (FD00+32bit generated*) + *16bit Subnet ID* + Interfac ID e.g. FD9E:21A7:A92C:*2323*::1	RFC 4193, *merger collisions possible, can be registered here: http://www.sixxs.r tools/grh/ula/
Global Unicast	everything else	2001:0DB8:0:CD30:123:4567:AB:CDEF	

Multicast	FF00::/8	FF + 4bit flags + 4bit scop + 112 bits group identifie	RFC 4291, see examples below
All nodes multicast address	FF02::1/128	"Multicast Ping": ping6 -I en0 ff02::1	equivalent to ipv4 broadcast, RA destination addr
All routers multicast address	FF02::2/128	link local scope all route multicast	RS destination add
All DHCP Servers and Relay Agents multicast address	FF02::1:2/128	all DHCP Server and Relays listen at this link local scope multicast address	used by DHCP clien
All DHCP Servers multicast address	FF05::1:3	Relay Agents can use this addr to talk to all DHCP servers site-scoped	used by DHCP relay agents
Solicited node multicas address	FF02::1:FF*ab:cdef*	FF02::1:FF00:0000/104 + *la 24 Bit* of the correspondin address, e.g. FF02::1:FF*AB:CDEF*	RFC 4291, used for DAD and NS (Name Resolution)
Source specific multica	FF3x:/32	x=scope	RFC 3569 informational
Anycast		for redundancy, load balancing (DNS over UDP, multiple Router, Mobile) Subnet Prefix + Interface + 7bit Anycast ID (e.g. reserved Anycast ID 7E for Mobile IPv6 Home Agent)	taken from the unicast address spaces RFC 2526
Subnet Router Anycast		n bits Subnet-Prefix+ 128-r bits "0"	mandatory router anycast address
6to4	2002::/16	Overall-Prefixlength is /4 (16Bit+32Bit IPv4 address)	RFC 3056
Teredo	2001::/32	2001:0000 + Teredo-server-ipv4-address + Flags + Client-UDP-Teredo-Port + Client-ipv4-address	RFC 4380 (inverted v4-addresses and port)
Documentation	2001:DB8::/32	used for documentation purposes	RFC 3849
(6bone)	was 3FFE/16		
ISATAP		64bit global unicast+00:0***0/2***:5E:FE+32bi v4-address	RFC 5214
NAT64	64:ff9b::/96		RFC 6146

2.2 Helpful URLs

http://ipv6-test.com/
http://www.heise.de/netze/tools/meine-ip-adresse/
http://ipv6test.google.com/

2.3 FAQ

- How to stay root in sudo environment: sudo -i, until exit
- How to run cmd.exe as Administrator: Copy link to desktop, right-click, Run as Administrator
- Attack tool doesn't work (with or without error mssage)? ➔ Make sure you're root
- Error: could not capture on interface eth0 with string ip6 and icmp6 and dst ff02::2 ➔become root: sudo -i
- detect-new-ip doesn't work ➔ enable promisc. mode (ifconfig eth0 promisc)
- Error: kill_router6 mtu errormessage when sending RAs ➔A bug, it only works if ethernet interface is eth0 (maybe fixed in 2.0?)

2.4 Install thc-Tools

- You can download the latest free version from http://thc.org.
- The toolkit currently only runs on Linux
- Install libraries: libssl-dev, libpcap-dev

```
tar -xzf thc-ipv6-1.9.tar.gz
cd thc-ipv6-1.9/
make
sudo make install
make clean
```

3. Network Configuration Form

	Linux	Windows
Name of the Ethernet-Interface connected to LAN (e.g. eth0):		
Configured IPv6 address(es) for this interface:		
• Link local	fe80	fe80
• Global Unicast static	2001 fd	2001 fd
• Global Unicast random (privacy)	2001 fd	2001 fd
IPv6 address(es) of router (default gateway):		
MAC address of interface		
Interface MTU		

fc/fd = ULA, see 2.1