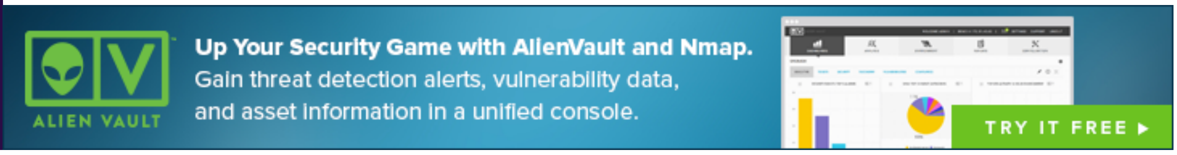
ALIEN VAULT
Up Your Security Game with AlienVault and Nmap.
Gain threat detection alerts, vulnerability data,
and asset information in a unified console.
TRY IT FREE ▸

Nmap Security Scanner
Intro
Ref Guide
Install Guide
Download
Changelog
Book
Docs

Security Lists
Nmap Announce
Nmap Dev
Bugtraq
Full Disclosure
Pen Test
Basics
More

Security Tools
Password audit
Sniffers
Vuln scanners
Web scanners
Wireless
Exploitation
Packet crafters
More

Site News
Advertising
About/Contact

Site Search
Sponsors:

Nmap Development mailing list archives

By Date By Thread Search

[patch] support for ip options

From: "majek04" <nmap () forest one pl>
Date: Fri, 7 Jul 2006 03:42:22 +0200 (CEST)

```
Hi guys.

Ip options support for nmap can be found here:
http://ai.pjwstk.edu.pl/~majek/private/nmap/g14-ipopts-3640.diff
(patch is for nmap-4.20ALPHA4)

This patch adds new interesting option to nmap.



BASIC USAGE for "--ip-options"
For the simplest usage we created some templates.
They will build ip options for you and you don't need
to know the internals.
Supported ip options:
"R"      -> record route (9 slots availble)
"T"      -> record internet timestamps (9 slots)
"U"      -> record timestamps and ip addresses (4 slots)
"L <hop_ip> ..." -> loose source routing (8 slots)
"S <hop_ip> ..." -> strict source routing (8 slots)

ADVANCED USAGE
There is also possibility of building custom ip options.
Consider reading rfc 791.
Supported values in parameter:
        "\xFF"  -> hex value
        "\255"  -> decimal value
        "\x00*16" -> copying char many times



BASIC EXAMPLES:
RECORD ROUTE:
./nmap -n -sP --packet-trace --ip-options "R" scanme.insecure.org
RCVD (0.2790s) ICMP 205.217.153.62 > <censored> Echo reply (type=0/code=0)
ttl=46 id=26078 iplen=68 ipopts={ RR{ <3 hops censored> 212.191.126.4
213.248.77.247 213.248.64.225 213.248.64.254 213.248.70.225 64.125.23.14#}
EOL}

These results are very different from ip's recorded by traceroute.
(because are recorded on different interfaces?)
<3 hops censored>
 4  212.191.224.73  4.770 ms  5.421 ms  5.199 ms
 5  213.248.77.213  16.477 ms  17.059 ms  16.887 ms
 6  80.91.249.197  16.494 ms  17.055 ms  16.547 ms
 7  213.248.65.153  23.255 ms  22.696 ms  23.263 ms
 8  80.91.249.216  23.249 ms  22.768 ms  22.854 ms
 9  129.250.9.221  23.395 ms  23.229 ms  23.096 ms


INTERNET TIMESTAMP (flags = 0)
./nmap -n -sP --packet-trace  --ip-options "T"  7thguard.pl
RCVD (0.1170s) ICMP 217.73.31.27 > <censored> Echo reply (type=0/code=0)
```

```
ttl=58 id=16559 iplen=68 ipopts={ TM{[5 hosts not recorded] 2293415244
2390625 1806394 1806403 1806411 1806412 1806412 1806412 1806412#}}

Okay, maybe these timestamps (milliseconds from midnight) aren't very
interesting. But number of not recorded hosts is quite interesting.
We can count number of hops to target using this formula
nhops = (<guess_this>*15 + <hosts_not_recorded> +
<hosts_recorded(usually=9)>)/2
In these case we will have (0*15+5+9)/2=7 hops.

INTERNET TIMESTAMP (flags = 1 <record timestamp and ip>)
./nmap -n -sP --packet-trace  --ip-options "U"  7thguard.pl
RCVD (0.0710s) ICMP 217.73.31.27 > <censored> Echo reply (type=0/code=0)
ttl=58 id=16560 iplen=64 ipopts={ TM{[9 hosts not recorded]
<cens>.1.2@2293429127 <cens>.1.1@2529464 <cens>.255.218@1945236
193.87.3.226@1945235#}}

Now we have timestamps with ip's. Nothing important.


LOOSE SOURCE ROUTE:
We can select middle hops.
./nmap -n -sS -p139 --packet-trace  --ip-options "S 192.168.1.1 10.0.0.3"
10.0.0.5
RCVD (0.0100s) ICMP 10.0.0.5 > 192.168.1.4 Echo reply (type=0/code=0)
ttl=126 id=38262 iplen=44 ipopts={ NOP LSRR{ 10.0.0.3 192.168.1.1
192.168.1.1#}}

STRICT SOURCE ROUTE:
Again, we can select middle hops.
RCVD (0.0120s) ICMP 10.0.0.5 > 192.168.1.4 Echo reply (type=0/code=0)
ttl=126 id=38255 iplen=44 ipopts={ NOP SSRR{ 10.0.0.3 192.168.1.1
192.168.1.1#}}




EXAMPLE BASIC OS DETECTION (windows/linux):
Let's send some bogus ip option:
./nmap -n -sO -p1 -PE --max-retries 0 --packet-trace  --ip-options
"\1\8\3\4" <target>
SENT (0.0070s) ICMP 192.168.1.4 > 10.0.0.3 Echo request (type=8/code=0)
ttl=48 id=11837 iplen=32 ipopts={ NOP ??{\x08\x03\x04}}

Linux responce:
RCVD (0.0080s) ICMP 10.0.0.3 > 192.168.1.4 Echo reply (type=0/code=0)
ttl=64 id=29711 iplen=28
Windows responce:
RCVD (0.0060s) ICMP 10.0.0.5 > 192.168.1.4 Echo reply (type=0/code=0)
ttl=128 id=21351 iplen=32 ipopts={ NOP ??{\x08\x03\x04}}

Linux removed bugus/unsupported option, windows left it without changes.
Other thing is order of options. Linux is changing sequence, windows is
leaving it as it was.
Of course this kind of detecting OS is as obscure as looking on ttl field.


EXAMPLE OF TEST IF LOOSE ROUTE IS ENABLED:
My ip 192.168.1.4, tested host 10.0.0.3. Packet is created with custom loose
source route field (type=131). I would get back this packet if loose route
is enabled.
./nmap -n -sO -p1 -PE --max-retries 0 --packet-trace --ip-options
"\1\131\7\4\192\168\1\4" 10.0.0.3
SENT (0.0070s) ICMP 192.168.1.4 > 10.0.0.3 Echo request (type=8/code=0)
ttl=42 id=50362 iplen=36 ipopts={ NOP LSRR{#192.168.1.4}}
RCVD (0.0080s) ICMP 192.168.1.4 > 192.168.1.4 Echo request (type=8/code=0)
ttl=40 id=50362 iplen=36 ipopts={ NOP LSRR{ 192.168.1.3#}}
We sent echo request, target forwarded this request back -> loose routing
is working.

Let's try different host:
SENT (0.0060s) ICMP 192.168.1.4 > 10.0.0.5 Echo request (type=8/code=0)
ttl=39 id=28683 iplen=36 ipopts={ NOP LSRR{#192.168.1.4}}
RCVD (0.0070s) ICMP 10.0.0.5 > 192.168.1.4 source route failed
(type=3/code=5) ttl=128 id=21357 iplen=64
```

Source routing is not working.

And my favorite useless example:

```
./nmap -n -sO -p1 -PE --max-retries 0 --packet-trace -e eth0 --ip-options
"L 192.168.1.3" 127.0.0.1
```

Linux is just dropping this packet with no response.

EXAMPLE OF STRICT ROUTE USAGE FOR NETWORK DISCOVERY:
When using strict source route, router must have next ip address directly
on link. This allows us to check what networks are connected to such host.

Scan target 192.168.1.1 is router nearest to me.
Lets try first net:

```
./nmap -n -sO -p1 -PE --max-retries 0 --packet-trace  --ip-options "S
192.168.1.1" 172.0.0.123
SENT (0.0060s) ICMP 192.168.1.4 > 192.168.1.1 Echo request (type=8/code=0)
ttl=40 id=57758 iplen=40 ipopts={ NOP SSRR{#192.168.1.1 172.0.0.123}}
RCVD (0.0060s) ICMP 192.168.1.1 > 192.168.1.4 source route failed
(type=3/code=5) ttl=64 id=29720 iplen=68
```

Nope, network 172.0.0. not present. The same probe with host 10.0.0.123
results in no answer. Packet went into darnkess, but network is probably
present.

Of course the main advantage of loose/strict route is that
you can scan bypass some misconfigured acl's on routers.

Some simillar information can be found in my previous email:
http://seclists.org/lists/nmap-dev/2006/Apr-Jun/0431.html

You should notice that while OS detection sending ip options is disabled.
It's recommended not to use ip options and os detection at the same time.

Cheers,
Marek Majkowski

By Date By Thread

Current thread:

- **[patch] support for ip options** *majek04 (Jul 06)*

[Nmap | Sec Tools | Mailing Lists | Site News | About/Contact | Advertising | Privacy]

Search