SANS

SECURITY 567:

Power Packet Crafting with Scapy – *Tip Sheet*

Get more details at **www.sans.org/training/power-packet-crafting-with-scapy-1382-mid**

Here are some common scapy commands, what they are used for, and examples of their use.

scapy commands	Used For	Example
ls()	List protocols or your variable	ls(myIP)
lsc()	List supported commands	lsc()
send()	Send layer 3, no concern for response	send(packet)
sr1()	Send layer 3, match 1 response	synack=sr1(ip/syn)
sr()	Send layer 3 packets, match all responses	sr(packets)
sendp()	Send layer 2, no concern for response	sendp(frame)
srp1()	Send layer 2, match 1 response	srp1(frame)
srp()	Send layer 2 frames, match all response	srp(frames)
rdpcap()	Read a pcap to a list	recs=rdpcap(“/tmp/pcap”)
wrpcap()	Write a list of packets to pcap	wrpcap(“/tmp/pcap”, list)
var.getlayer(protocol)	Extract a layer(s) from packet	ip=packet.getlayer(IP)
var.payload	Shows all layers after first	ip.payload
var.summary()	Shows a summary of packet	packet.summary()
sniff	Sniffs packets	sniff(filter=”bpf”, count=2)

This is how interactive scapy may be used to send a TCP SYN segment to two destination hosts, 172.22.7.133 and 172.22.10.132, to four different destination ports – 111, 139, 445, and 80.

```
>>> sr(IP(dst=[“172.22.7.133”, “172.22.10.132”])/TCP(dport = [111, 139, 445, 80], flags=”S”))
```

Here is how interactive scapy may be used to craft a UDP datagram to send to destination host 172.22.7.133 and destination port 13 with some payload and listen for the response.

```
>>> ip=IP(dst=”172.22.7.133”)
>>> udp=UDP(sport=1024,dport=13)
>>> payload=”All your base are belong to us”
>>> packet=ip/udp/payload
>>> sr1(packet)
```

The following is a simple scapy program that creates an actual TCP session from source host 172.22.8.135 and source port 1030 to destination host 172.22.7.133 and destination port 80. It crafts the SYN segment, listens for the server’s SYN/ACK and acknowledges it to complete the three-way handshake. Next, it sends a segment that contains data. There is an issue with an undesirable side effect that creates a reset and inadvertently aborts the session. The reason and circumvention are discussed in the course.

```
#!/usr/bin/python
from scapy.all import *
ip=IP(src=”172.22.8.135”, dst=”172.22.7.133”)
SYN=TCP(sport=1030, dport=80, flags=”S”, seq=10)
SYNACK=sr1(ip/SYN)
my_ack = SYNACK.seq + 1
ACK=TCP(sport=1030, dport=80, flags=”A”, seq=11, ack=my_ack)
send(ip/ACK)
payload = “SEND TCP”
PUSH=TCP(sport=1030,dport=80, flags=”PA”, seq=11, ack=my_ack)
send(ip/PUSH/payload)
```

Author Judy Novak says
“Once you’ve tried scapy, you’ll never go back to using command line packet crafting tools!”

Want to find out more about packet crafting with scapy?

Check out the new
SEC567: Power Packet Crafting with Scapy

http://www.sans.org/training/
power-packet-crafting-with-scapy-1382-mid