Case 1, Denial of Service

CYS 201 | Grupp - 42

A: Vad betyder och gör de olika växlarna (ex,-a) ni använder?

- -1 : Anger att vi använder **ICMP-läge**, vilket motsvarar "ping"-trafik istället för exempelvis TCP.
- – flood : Gör så att paketen skickas så snabbt som möjligt utan att vänta på svar.
- -a : Förfalskar avsändarens IP-adress, vilket gör att det ser ut som att paketen kommer därifrån.
- -p: Anger port 80 (HTTP) som målport, vanligtvis används denna port av webbservrar.

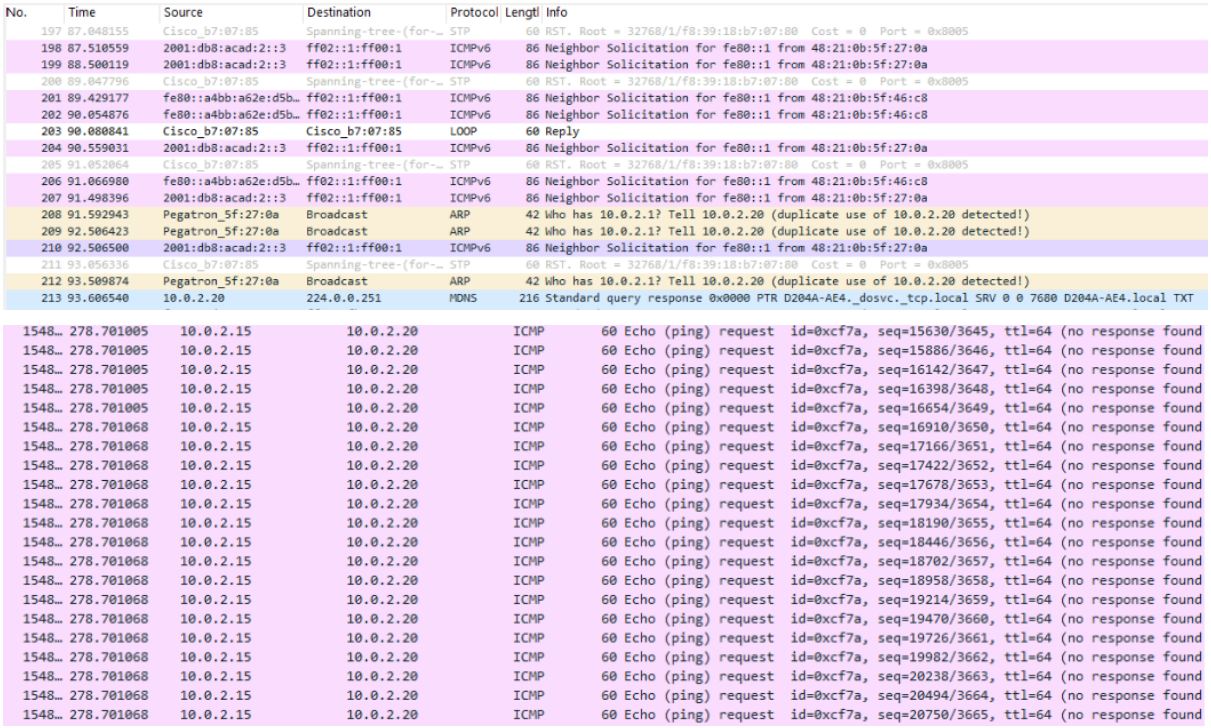
B: **Vad är det för typ av attack och vad är karakteristiskt för just den?**

- Detta är en smurf attack, en typ av Denial of Service. den används för att överbelasta en dator, server eller ett nätverk så att det slutar fungera som det ska.
- Hur fungerar det?
    - Angriparen skickar många ICMP-paket till målet
    - Med kommandot –flood skickas paketet väldigt snabbt
    - Genom Ip-spoofing förflaskas avsändarensaddressen så att det ser ut som att paketet kommer från någon annan, vilket gör det svårt att spåra attacken.
- Nej, det är väldigt svårt att använda datorn när den är under attack. Under attacken blev datorn **mycket långsam**, eftersom både **CPU:n och minnet** belastades hårt.

C: **Hur påverkas Linux datorns resurser?Vilken resurs belastas mest under attacken?**

- När en Linux-dator används för att skicka stora mängder ICMP-paket utan paus, som vid en Smurf-attack, påverkas datorns resurser på flera sätt. Den resurs som belastas mest är att CPU Processorn får arbeta hårt med att hela tiden skapa och skicka ut paket, vilket kan leda till att den blir överhettad eller långsam.

D) Öppna Wireshark på PC A, ta en skärmdump. Klistra in i dokumentet och beskriv vad det är som händer?

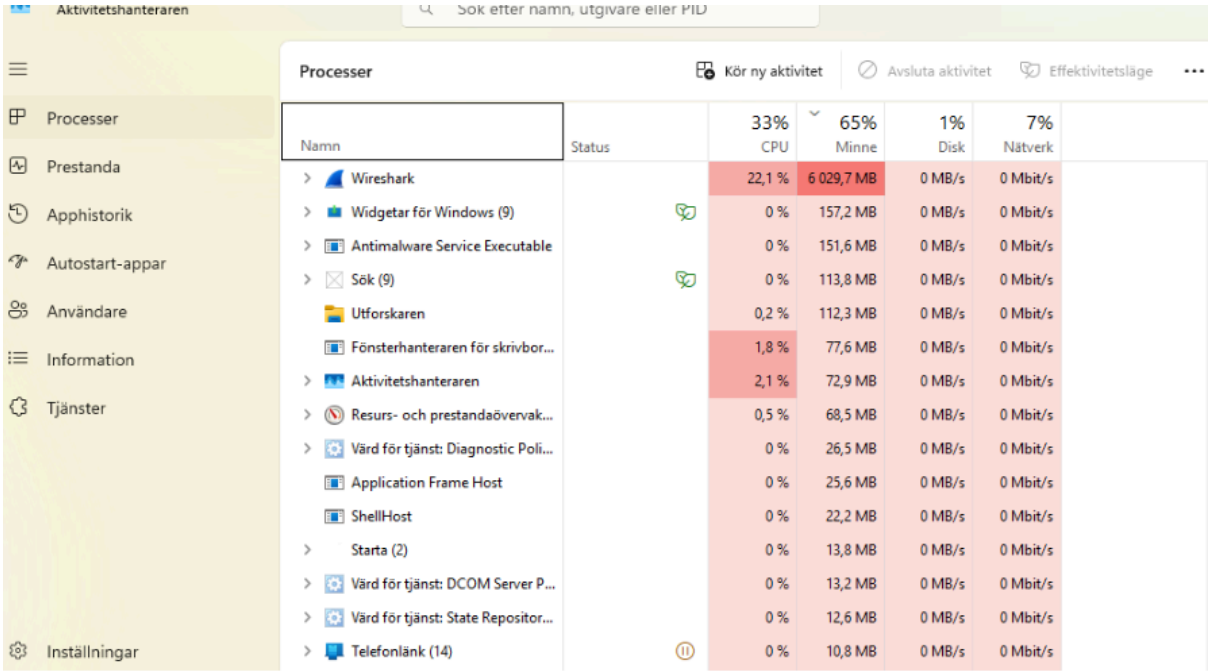


No.	Time	Source	Destination	Protocol	Length	Info
197	87.048155	Cisco_b7:07:85	Spanning-tree-(for-…	STP	60	RST. Root = 32768/1/f8:39:18:b7:07:80 Cost = 0 Port = 0x8005
198	87.510559	2001:db8:acad:2::3	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 48:21:0b:5f:27:0a
199	88.500119	2001:db8:acad:2::3	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 48:21:0b:5f:27:0a
200	89.047796	Cisco_b7:07:85	Spanning-tree-(for-…	STP	60	RST. Root = 32768/1/f8:39:18:b7:07:80 Cost = 0 Port = 0x8005
201	89.429177	fe80::a4bb:a62e:d5b…	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 48:21:0b:5f:46:c8
202	90.054876	fe80::a4bb:a62e:d5b…	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 48:21:0b:5f:46:c8
203	90.080841	Cisco_b7:07:85	Cisco_b7:07:85	LOOP	60	Reply
204	90.559031	2001:db8:acad:2::3	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 48:21:0b:5f:27:0a
205	91.052064	Cisco_b7:07:85	Spanning-tree-(for-…	STP	60	RST. Root = 32768/1/f8:39:18:b7:07:80 Cost = 0 Port = 0x8005
206	91.066980	fe80::a4bb:a62e:d5b…	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 48:21:0b:5f:46:c8
207	91.498396	2001:db8:acad:2::3	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 48:21:0b:5f:27:0a
208	91.592943	Pegatron_5f:27:0a	Broadcast	ARP	42	Who has 10.0.2.1? Tell 10.0.2.20 (duplicate use of 10.0.2.20 detected!)
209	92.506423	Pegatron_5f:27:0a	Broadcast	ARP	42	Who has 10.0.2.1? Tell 10.0.2.20 (duplicate use of 10.0.2.20 detected!)
210	92.506500	2001:db8:acad:2::3	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 48:21:0b:5f:27:0a
211	93.056336	Cisco_b7:07:85	Spanning-tree-(for-…	STP	60	RST. Root = 32768/1/f8:39:18:b7:07:80 Cost = 0 Port = 0x8005
212	93.509874	Pegatron_5f:27:0a	Broadcast	ARP	42	Who has 10.0.2.1? Tell 10.0.2.20 (duplicate use of 10.0.2.20 detected!)
213	93.606540	10.0.2.20	224.0.0.251	MDNS	216	Standard query response 0x0000 PTR D204A-AE4._dosvc._tcp.local SRV 0 0 7680 D204A-AE4.local TXT

1548…	278.701005	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=15630/3645, ttl=64 (no response found
1548…	278.701005	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=15886/3646, ttl=64 (no response found
1548…	278.701005	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=16142/3647, ttl=64 (no response found
1548…	278.701005	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=16398/3648, ttl=64 (no response found
1548…	278.701005	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=16654/3649, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=16910/3650, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=17166/3651, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=17422/3652, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=17678/3653, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=17934/3654, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=18190/3655, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=18446/3656, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=18702/3657, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=18958/3658, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=19214/3659, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=19470/3660, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=19726/3661, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=19982/3662, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=20238/3663, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=20494/3664, ttl=64 (no response found
1548…	278.701068	10.0.2.15	10.0.2.20	ICMP	60	Echo (ping) request id=0xcf7a, seq=20750/3665, ttl=64 (no response found

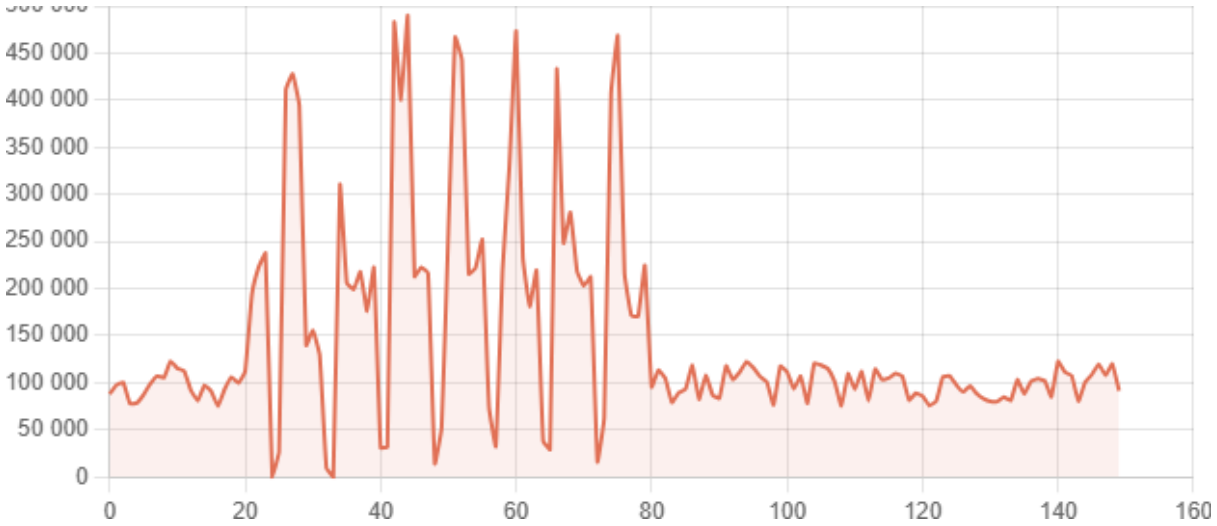
- Wireshark är ett program som används för att övervaka nätverkstrafik. Det visar vad som skickas och tas emot i nätverket i realtid. När vi startar en Smurf-attack och öppnar Wireshark på datorn, kan vi tydligt se vad som händer. I detta fall visar Wireshark en stor mängd ICMP-paket (ping), som kommer hela tiden utan paus. Dessa paket visas ofta i lila färg, vilket gör det lätt att se dem i listan.

E) Öppna också upp Task manager i Windows gå till Processes ta en skärmdump. Klistra in i dokumentet och beskriv vad det är som händer



- När vi öppnar Aktivitetshanteraren i Windows och går till fliken Processer under en pågående attack, kan vi se att datorns resursanvändning ökar kraftigt. Framför allt märks att CPU-användningen går upp, eftersom datorn måste hantera och bearbeta ett stort antal nätverkspaket.

F) Gå in i Wireshark igen klicka på statistics och sedan på IO Graphs, ta en skärmdump och klistra in i dokumentet.Vad kan ni utläsa genom att analysera grafen?



G)Går det bra att använda Windows datorn under attacken?

- Nej, det är väldigt svårt att använda datorn när den är under attack. Under attacken blev datorn mycket långsam, eftersom både CPU:n och minnet belastades hårt. Vi märkte till exempel att det tog lång tid att öppna program och att Wireshark blev så seg att det nästan var omöjligt att använda.

UDP FLOOD ATTACK

A: I det här kommandot används sudo hping3 för att skicka specialpaket över nätverket. Här är vad varje del betyder:

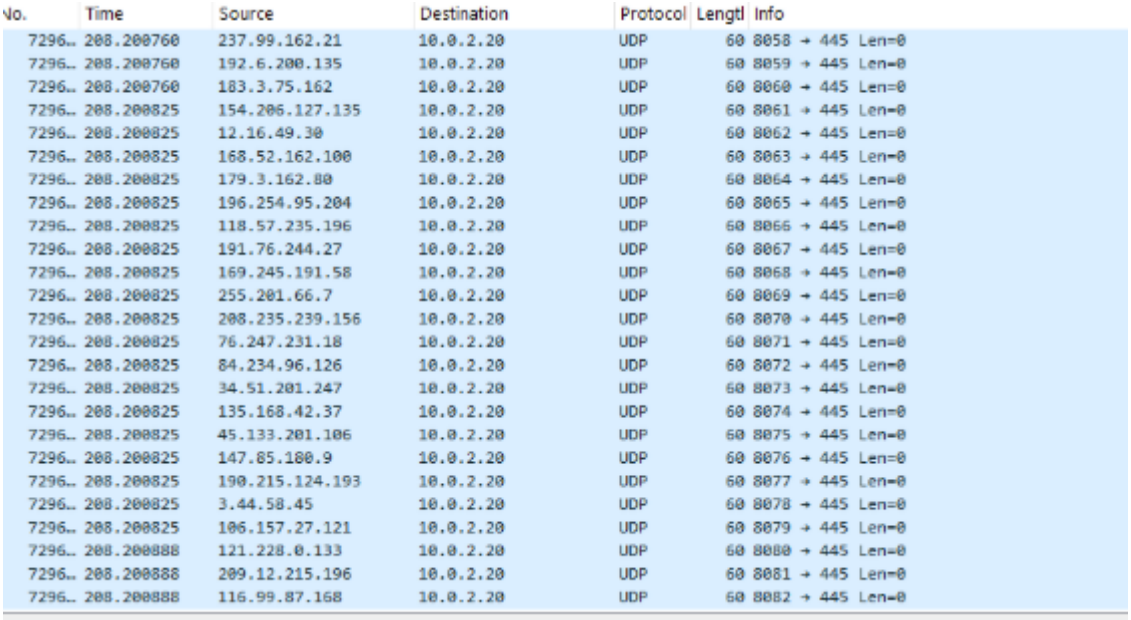
- Sudo: Ger administratörsrättigheter så att kommandot kan köras med full åtkomst, vilket krävs för att skicka nätverkspaket på låg nivå.
- hping3: Ett avancerat verktyg som låter oss skicka olika typer av nätverkspaket som TCP, UDP, och ICMP.
- –flood: Skickar paketen i högsta möjliga hastighet, utan att vänta på svar från mottagaren.
- –rand-source: Gör att varje paket får en slumpad avsändar-IP-adress, vilket försvårar för målet att identifiera eller blockera angriparen.
- –udp: Anger att paketen som skickas är av typen UDP.
- -p 445: Bestämmer att paketen ska skickas till port 445

B: Det finns flera typer av DoS-attacker (Denial of Service), och två vanliga varianter är UDP flood och TCP SYN flood.

- UDP Flood attack: Den här attacken skickar en stor mängd UDP-paket till målets IP-adress – i detta fall till port 445, som ofta används av Windows-system. Attacken använder falska (spoofade) IP-adresser som slumpas för varje paket. Det gör det mycket svårt att spåra vem som ligger bakom attacken.
- TCP SYN Flood Attack: Detta är en annan form av DoS-attack som utnyttjar TCP:s three-way handshake

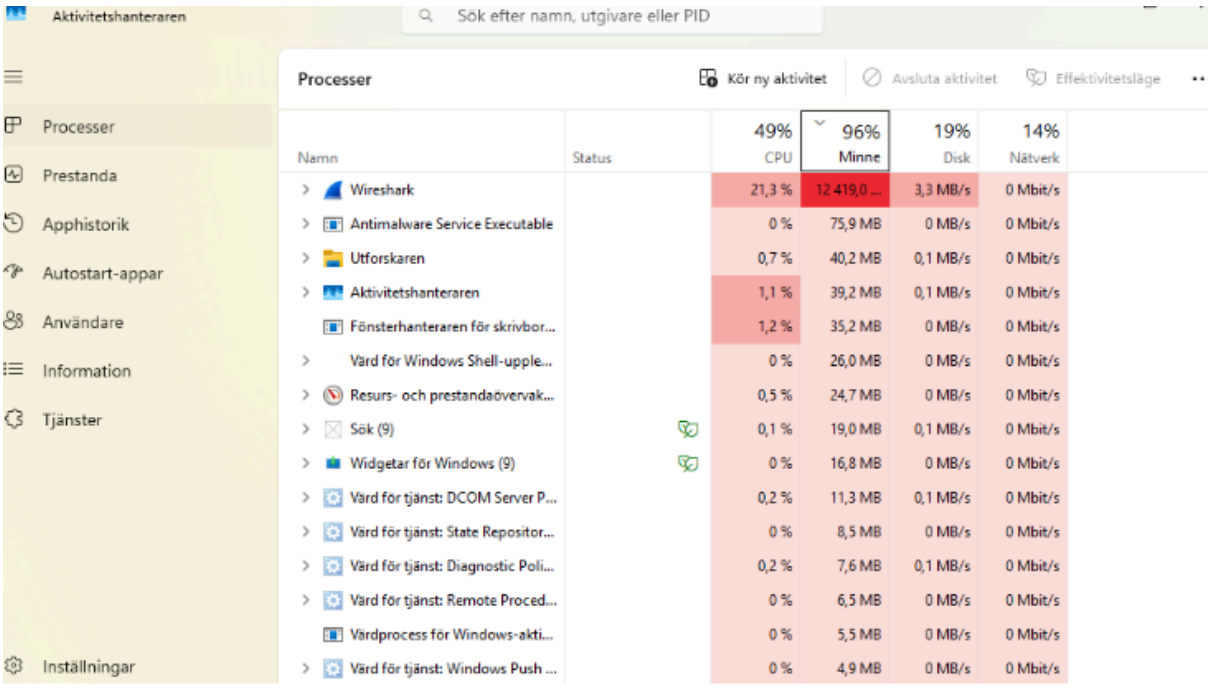
C: Vid en UDP flood-attack påverkas Linux-datorns resurser kraftigt. Den resurs som belastas mest är nätverkskortet eftersom datorn tvingas hantera en mycket stor mängd inkommande datapaket. Även CPU:n får arbeta hårt för att bearbeta all trafik.

D: Här visas wireshark många UDP-paket till IP-adressen 10.0.2.20.

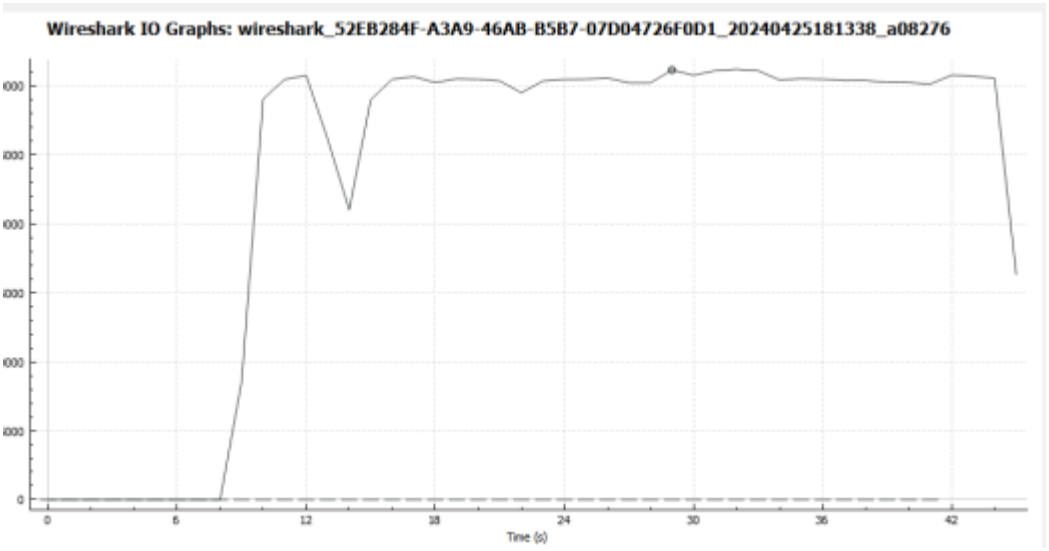


No.	Time	Source	Destination	Protocol	Lengtl	Info
7296…	208.200760	237.99.162.21	10.0.2.20	UDP	60	8058 → 445 Len=0
7296…	208.200760	192.6.200.135	10.0.2.20	UDP	60	8059 → 445 Len=0
7296…	208.200760	183.3.75.162	10.0.2.20	UDP	60	8060 → 445 Len=0
7296…	208.200825	154.206.127.135	10.0.2.20	UDP	60	8061 → 445 Len=0
7296…	208.200825	12.16.49.30	10.0.2.20	UDP	60	8062 → 445 Len=0
7296…	208.200825	168.52.162.100	10.0.2.20	UDP	60	8063 → 445 Len=0
7296…	208.200825	179.3.162.80	10.0.2.20	UDP	60	8064 → 445 Len=0
7296…	208.200825	196.254.95.204	10.0.2.20	UDP	60	8065 → 445 Len=0
7296…	208.200825	118.57.235.196	10.0.2.20	UDP	60	8066 → 445 Len=0
7296…	208.200825	191.76.244.27	10.0.2.20	UDP	60	8067 → 445 Len=0
7296…	208.200825	169.245.191.58	10.0.2.20	UDP	60	8068 → 445 Len=0
7296…	208.200825	255.201.66.7	10.0.2.20	UDP	60	8069 → 445 Len=0
7296…	208.200825	208.235.239.156	10.0.2.20	UDP	60	8070 → 445 Len=0
7296…	208.200825	76.247.231.18	10.0.2.20	UDP	60	8071 → 445 Len=0
7296…	208.200825	84.234.96.126	10.0.2.20	UDP	60	8072 → 445 Len=0
7296…	208.200825	34.51.201.247	10.0.2.20	UDP	60	8073 → 445 Len=0
7296…	208.200825	135.168.42.37	10.0.2.20	UDP	60	8074 → 445 Len=0
7296…	208.200825	45.133.201.106	10.0.2.20	UDP	60	8075 → 445 Len=0
7296…	208.200825	147.85.180.9	10.0.2.20	UDP	60	8076 → 445 Len=0
7296…	208.200825	190.215.124.193	10.0.2.20	UDP	60	8077 → 445 Len=0
7296…	208.200825	3.44.58.45	10.0.2.20	UDP	60	8078 → 445 Len=0
7296…	208.200825	106.157.27.121	10.0.2.20	UDP	60	8079 → 445 Len=0
7296…	208.200888	121.228.0.133	10.0.2.20	UDP	60	8080 → 445 Len=0
7296…	208.200888	209.12.215.196	10.0.2.20	UDP	60	8081 → 445 Len=0
7296…	208.200888	116.99.87.168	10.0.2.20	UDP	60	8082 → 445 Len=0

E: Under attacken ökar CPU:ns belastning kraftig eftersom den har svårt att hinna med att hantera alla inkommande förfrågningar.



F: I/O grafen visar dataflödet som följande



G: Nej, det gick inte bra att använda Windows-datorn under attacken. Både **CPU:n och RAM-minnet** blev kraftigt belastade, vilket gjorde datorn mycket seg.

PING FLOOD ATTACK

A: Vad betyder växlarna?

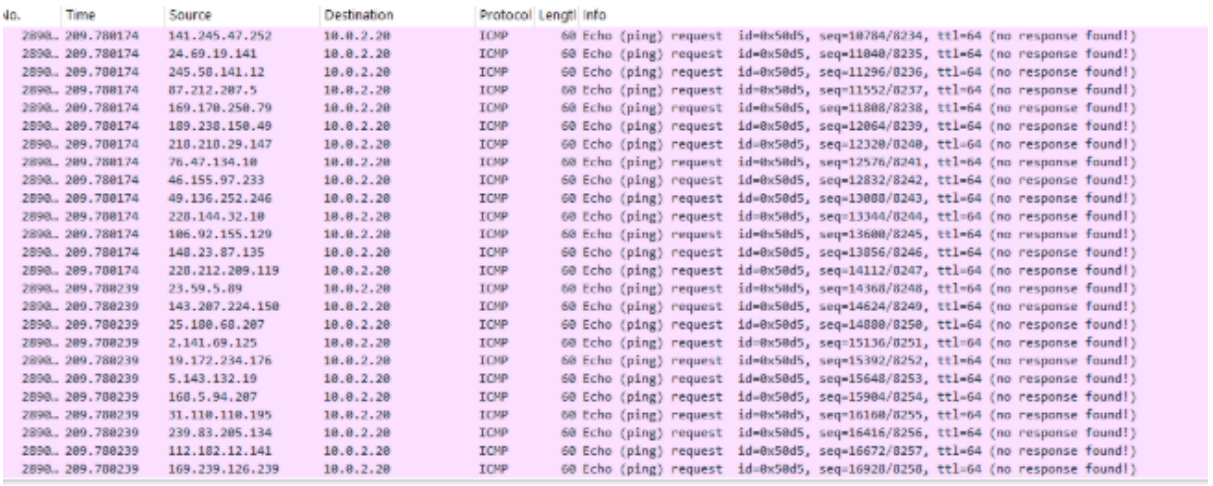
- -c: Antal paket som ska skickas (t.ex. 10 000).
- --flood: Skickar paket så snabbt som möjligt utan att vänta på svar.

- --rand-source: Skickar paketen med slumpade IP-adresser för att dölja avsändaren.
- --icmp: Anger att ICMP-paket används (som i en smurfattack).
- -p: Bestämmer porten som attacken riktas mot (t.ex. port 455).

B) Typ av attack och kännetecken:
Det är en ICMP flood-attack med förfalskade IP-adresser. Paket skickas snabbt utan att vänta på svar, vilket överbelastar mottagaren med ping-förfrågningar.

C) Påverkan på Linux-datorn:
CPU:n belastas mest eftersom den måste skapa och skicka stora mängder paket. Även nätverkskortet påverkas av trafiken.

D) Vad visar Wireshark?
Wireshark visar att många ICMP-paket tas emot från slumpmässiga IP-adresser. Det visar att attacken försöker dölja vem som skickar paketen.



No.	Time	Source	Destination	Protocol	Length	Info
2890…	209.780174	141.245.47.252	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=10784/8234, ttl=64 (no response found!)
2890…	209.780174	24.69.19.141	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=11040/8235, ttl=64 (no response found!)
2890…	209.780174	245.58.141.12	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=11296/8236, ttl=64 (no response found!)
2890…	209.780174	87.212.207.5	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=11552/8237, ttl=64 (no response found!)
2890…	209.780174	169.170.250.79	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=11808/8238, ttl=64 (no response found!)
2890…	209.780174	189.238.150.49	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=12064/8239, ttl=64 (no response found!)
2890…	209.780174	210.210.29.147	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=12320/8240, ttl=64 (no response found!)
2890…	209.780174	76.47.134.10	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=12576/8241, ttl=64 (no response found!)
2890…	209.780174	46.155.97.233	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=12832/8242, ttl=64 (no response found!)
2890…	209.780174	49.136.252.246	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=13088/8243, ttl=64 (no response found!)
2890…	209.780174	220.144.32.10	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=13344/8244, ttl=64 (no response found!)
2890…	209.780174	106.92.155.129	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=13600/8245, ttl=64 (no response found!)
2890…	209.780174	148.23.87.135	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=13856/8246, ttl=64 (no response found!)
2890…	209.780174	220.212.209.119	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=14112/8247, ttl=64 (no response found!)
2890…	209.780239	23.59.5.89	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=14368/8248, ttl=64 (no response found!)
2890…	209.780239	143.207.224.150	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=14624/8249, ttl=64 (no response found!)
2890…	209.780239	25.180.68.207	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=14880/8250, ttl=64 (no response found!)
2890…	209.780239	2.141.69.125	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=15136/8251, ttl=64 (no response found!)
2890…	209.780239	19.172.234.176	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=15392/8252, ttl=64 (no response found!)
2890…	209.780239	5.143.132.19	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=15648/8253, ttl=64 (no response found!)
2890…	209.780239	160.5.94.207	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=15904/8254, ttl=64 (no response found!)
2890…	209.780239	31.110.110.195	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=16160/8255, ttl=64 (no response found!)
2890…	209.780239	239.83.205.134	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=16416/8256, ttl=64 (no response found!)
2890…	209.780239	112.182.12.141	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=16672/8257, ttl=64 (no response found!)
2890…	209.780239	169.239.126.239	10.0.2.20	ICMP	60	Echo (ping) request id=0x50d5, seq=16928/8258, ttl=64 (no response found!)

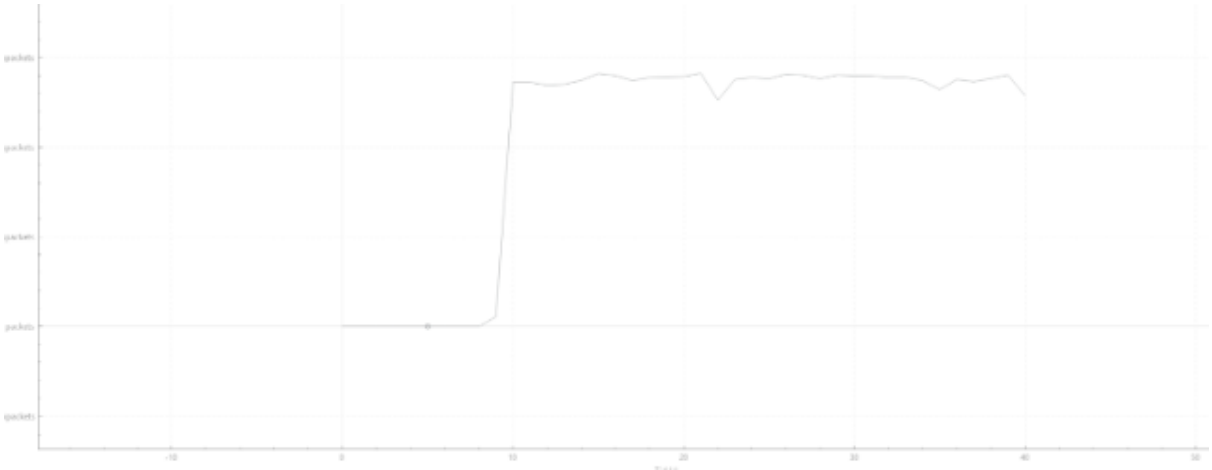
E) Vad visar Task Manager?
Under attacken ökar CPU- och RAM-användningen kraftigt. Datorn blir långsam och Wireshark använde så mycket minne att det nästan kraschade.

Namn	Status	CPU	Minne	Disk	Nätverk
Wireshark		25,4 %	12 262,8 ...	5,3 MB/s	0 Mbit/s
Utforskaren		0 %	112,5 MB	0 MB/s	0 Mbit/s
Antimalware Service Executable		0 %	99,7 MB	0 MB/s	0 Mbit/s
Sök (10)		0,1 %	64,5 MB	0,1 MB/s	0 Mbit/s
Aktivitetshanteraren		0,6 %	62,9 MB	0,1 MB/s	0 Mbit/s
Fönsterhanteraren för skrivbor...		0 %	35,4 MB	0 MB/s	0 Mbit/s
Widgetar för Windows (9)		0 %	35,2 MB	0 MB/s	0 Mbit/s
Foton		0 %	31,9 MB	0 MB/s	0 Mbit/s
Resurs- och prestandaövervak...		0,7 %	28,9 MB	0 MB/s	0 Mbit/s
Värd för tjänst: DCOM Server P...		0 %	12,1 MB	0 MB/s	0 Mbit/s
Värd för tjänst: State Repositor...		0 %	9,0 MB	0 MB/s	0 Mbit/s
Värd för tjänst: Diagnostic Poli...		0 %	8,7 MB	0,1 MB/s	0 Mbit/s
Indexerare för Microsoft Wind...		0 %	6,8 MB	0 MB/s	0 Mbit/s
Värd för tjänst: Remote Proced...		0 %	6,7 MB	0 MB/s	0 Mbit/s
Starta (2)		0 %	6,0 MB	0 MB/s	0 Mbit/s

askmanager under attacken

F) Vad visar IO Graphs i Wireshark?

Grafen visar att cirka 300 000 paket per sekund skickas. Wireshark blev så belastat att det slutade svara efter 40 sekunder.



G) Går det att använda Windows-datorn under attacken?

Nej, datorn blev mycket långsam. Det tog flera sekunder att klicka eller navigera, och vissa program kraschade på grund av minnesbrist.