SIPSA anonymization protocol

Kirils Solovjovs 28DEC2016

“We Fix the Net” session

About myself

- Kirils Solovjovs
  - IT security expert; researcher at 1st Ltd, Latvia
  - Network flow analysis, reverse engineering, social engineering, penetration testing, security incident investigation, and the legal dimension of cyber security and cyber defence

Anonymity online

- Anonymity on the internet has been a topic of wild debates and opposing opinions since the creation of the internet.
- So far there have been multiple attempts[1] to achieve partial anonymity, most of them done by routing the data through third-party machines, thus making it impossible to achieve true anonymity in an untrusted environment that is the internet.

[1] Tor, I2P, etc.

SIPSA

- Source IP spoofing for anonymization over UDP (SIPSA) is a proposal for a protocol that in many network environments would allow two hosts on the network to hide both their source and destination addresses in IP packets on the network level, without relying on any third party, while still being able to send and receive information.

Problem statement

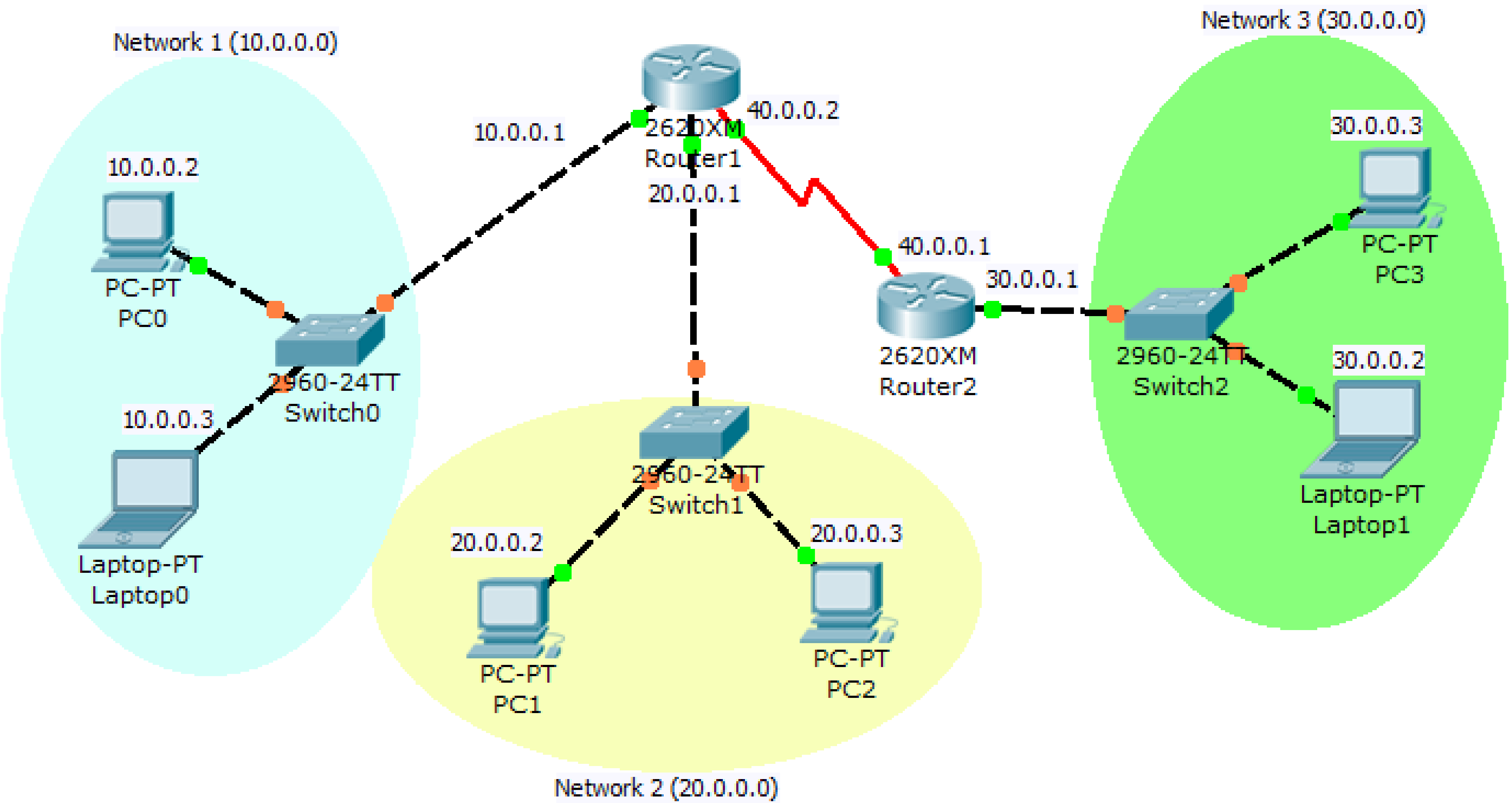
Anonymity on the internet

"On the Internet, nobody knows you're a dog."

Problem statement

Anonymity on the internet is hard

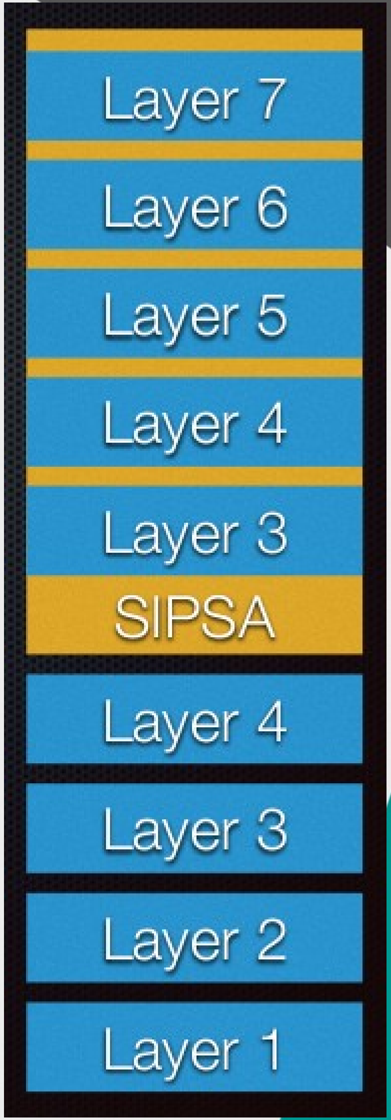
OUR NEXT GUEST HAS CHOSEN TO REMAIN ANONYMOUS, SO HE WILL REMAIN BEHIND THIS SCREEN DURING THE INTERVIEW.
PLEASE WELCOME MR. ISAAC PHILLIPS, AGE 26, FROM AMARILLO, TEXAS!
7
ASHER FREEMAN


Network 1 (10.0.0.0)
Network 3 (30.0.0.0)
10.0.0.1
2620XM
Router1
40.0.0.2
20.0.0.1
30.0.0.3
10.0.0.2
PC-PT
PC3
PC-PT
PC0
40.0.0.1
30.0.0.1
2960-24TT
Switch0
2620XM
Router2
2960-24TT
Switch2
30.0.0.2
10.0.0.3
2960-24TT
Switch1
Laptop-PT
Laptop1
20.0.0.2
20.0.0.3
Laptop-PT
Laptop0
PC-PT
PC1
PC-PT
PC2
Network 2 (20.0.0.0)

USE MORE
BANDWIDTH

SIPSA overview

- Instead of sending a single UDP datagram, many are sent
  - Different pairs of (randomised) source and destination IPs
- Protocol goes on top of Layer 4, but below Layer 3 [!]
- Current version (04) chooses IPs in pairs within a class C network



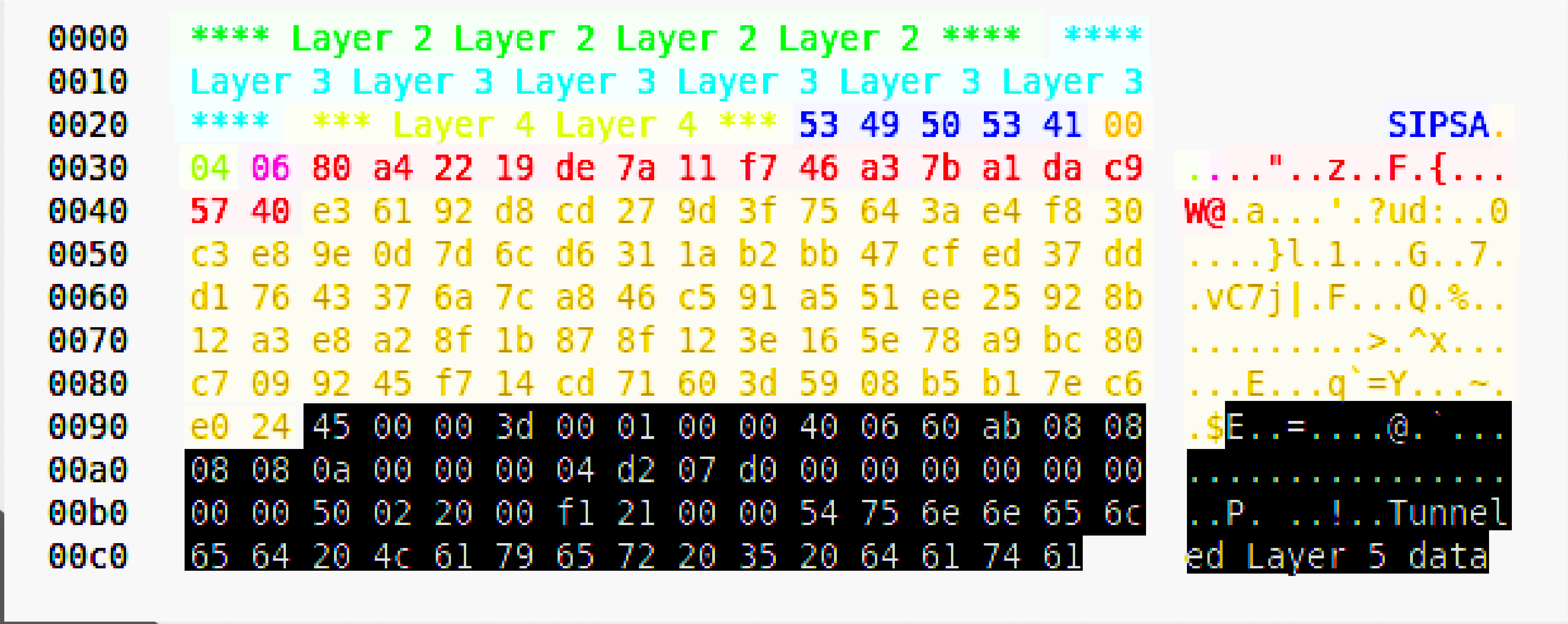
Randomisation algorithm (v04)

```
genPair(addr):
    addr1.addr2.addr3.addr4←addr
    genPair←[]
    genPair[]←addr
    genPair[]←addr1.addr2.addr3.{1-254}

addressList←[]
addressList[]←genPair(real)
for i←1..n:
    addressList[]←genPair({1-239}.{0-255}.{0-255}.{1-254})
```

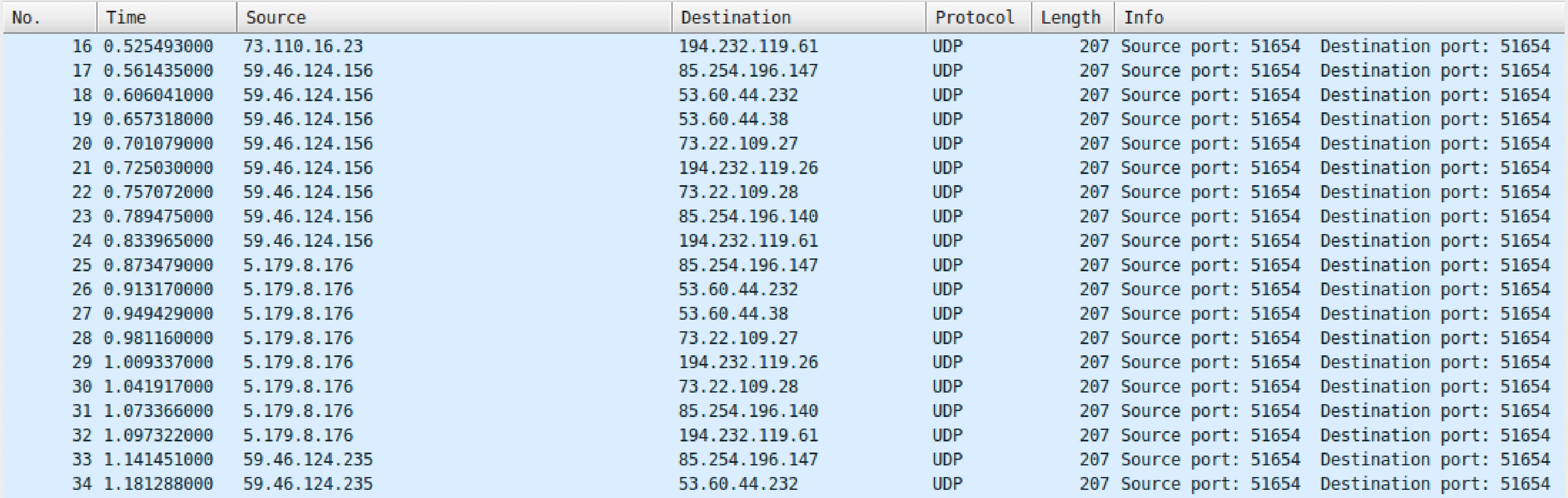
SIPSA datagram format

					ENCRYPTED with AES256, CBC mode, 16B block, iv=IV, total size = (Metalen-1)*16B							
Header 5B “SIPSA”	Reserved 1B “\00”	Proto ver 1B “\04”	Metalen 1B	IV 16B	Real src IP 4B may be zeros	Real dst IP 4B may be zeros	Src IP list 4B x n n≥0	End marker 1B “\xFF”	Dst IP list 4B x n n≥0	End marker 1B “\xFF”	Padding 0B – 15B “\x00”	Payload 0B +



```
0000   **** Layer 2 Layer 2 Layer 2 Layer 2 ****  ****
0010   Layer 3 Layer 3 Layer 3 Layer 3 Layer 3 Layer 3
0020   ****  *** Layer 4 Layer 4 *** 53 49 50 53 41 00              SIPSA.
0030   04 06 80 a4 22 19 de 7a 11 f7 46 a3 7b a1 da c9    ...."..z..F.{...
0040   57 40 e3 61 92 d8 cd 27 9d 3f 75 64 3a e4 f8 30    W@.a...'.?ud:..0
0050   c3 e8 9e 0d 7d 6c d6 31 1a b2 bb 47 cf ed 37 dd    ....}l.1...G..7.
0060   d1 76 43 37 6a 7c a8 46 c5 91 a5 51 ee 25 92 8b    .vC7j|.F...Q.%..
0070   12 a3 e8 a2 8f 1b 87 8f 12 3e 16 5e 78 a9 bc 80    .........>.^x...
0080   c7 09 92 45 f7 14 cd 71 60 3d 59 08 b5 b1 7e c6    ...E...q`=Y...~.
0090   e0 24 45 00 00 3d 00 01 00 00 40 06 60 ab 08 08    .$E..=....@.`...
00a0   08 08 0a 00 00 00 04 d2 07 d0 00 00 00 00 00 00    ................
00b0   00 00 50 02 20 00 f1 21 00 00 54 75 6e 6e 65 6c    ..P. ..!..Tunnel
00c0   65 64 20 4c 61 79 65 72 20 35 20 64 61 74 61       ed Layer 5 data
```

Results



No.	Time	Source	Destination	Protocol	Length	Info
16	0.525493000	73.110.16.23	194.232.119.61	UDP	207	Source port: 51654 Destination port: 51654
17	0.561435000	59.46.124.156	85.254.196.147	UDP	207	Source port: 51654 Destination port: 51654
18	0.606041000	59.46.124.156	53.60.44.232	UDP	207	Source port: 51654 Destination port: 51654
19	0.657318000	59.46.124.156	53.60.44.38	UDP	207	Source port: 51654 Destination port: 51654
20	0.701079000	59.46.124.156	73.22.109.27	UDP	207	Source port: 51654 Destination port: 51654
21	0.725030000	59.46.124.156	194.232.119.26	UDP	207	Source port: 51654 Destination port: 51654
22	0.757072000	59.46.124.156	73.22.109.28	UDP	207	Source port: 51654 Destination port: 51654
23	0.789475000	59.46.124.156	85.254.196.140	UDP	207	Source port: 51654 Destination port: 51654
24	0.833965000	59.46.124.156	194.232.119.61	UDP	207	Source port: 51654 Destination port: 51654
25	0.873479000	5.179.8.176	85.254.196.147	UDP	207	Source port: 51654 Destination port: 51654
26	0.913170000	5.179.8.176	53.60.44.232	UDP	207	Source port: 51654 Destination port: 51654
27	0.949429000	5.179.8.176	53.60.44.38	UDP	207	Source port: 51654 Destination port: 51654
28	0.981160000	5.179.8.176	73.22.109.27	UDP	207	Source port: 51654 Destination port: 51654
29	1.009337000	5.179.8.176	194.232.119.26	UDP	207	Source port: 51654 Destination port: 51654
30	1.041917000	5.179.8.176	73.22.109.28	UDP	207	Source port: 51654 Destination port: 51654
31	1.073366000	5.179.8.176	85.254.196.140	UDP	207	Source port: 51654 Destination port: 51654
32	1.097322000	5.179.8.176	194.232.119.61	UDP	207	Source port: 51654 Destination port: 51654
33	1.141451000	59.46.124.235	85.254.196.147	UDP	207	Source port: 51654 Destination port: 51654
34	1.181288000	59.46.124.235	53.60.44.232	UDP	207	Source port: 51654 Destination port: 51654

SIPSA should be able to provide anonymity and deniability

Weaknesses

- SIPSA gives only statistical improvement not 100% anonymity, so statistical attacks are likely possible
- Success largely depends on the ISPs involved
- Network load increase

BCP38

- Best Current Practice, May 2000 [!]
- Network Ingress Filtering
- Drops packets having unknown source prefix
- Supposed to solve DoS
- Worked well, but did not solve DoS in the long term (today)

Strengths and opportunities

- Ingress filtering has been sparsely implemented
- SIPSA may provide an additional layer of anonymity as part of a larger suite
- SIPSA provides deniability by virtue of UDP (and having fixed port numbering)
- Internet speeds are increasing fast

> “No, your honour. My devices neither requested nor acknowledged receipt of the communication in question.”
>
> –You, on SIPSA

Alternative configurations

- Consider not including real source IP in the metadata
  - Even the server has no way of knowing or logging client IPs
- Consider not sending packet from the real source at all
  - It's of course impossible to do both

Future work

- Key management
- Possible weaknesses due to statistical and other attacks
- Stateful SIPSA
- IPv6 support
- NAT support (impossible?)

FIN

- https://GitHub.com/0ki/SIPSA
- Now:
  - Discuss!
  - Fork and send merge/pull requests!
  - Test and implement in real protocols

- Want to contribute to a research paper?
- Reach me at @KirilsSolovjovs or via kirils.org