Harvesting Skype Super-Nodes

Anat Bremler-Barr, Omer Dekel
School of Computer Science, the Interdisciplinary Center
bremler@idc.ac.il
Dekel.omer@idc.ac.uil

Hanoch Levy
Computer Networking and Networks Lab, ETH, Zurich
On leave of absence from Tel-Aviv University

OWASP
Dec-03-07

Agenda

- Skype
- Blocking Skype – why ? and why is so difficult ?
- Our proposal
  - Harvesting Super-Nodes in order to block Skype - Under provisional patent
- Experiment results
- Conclusion

Skype

"… a free program that uses the latest P2P…technology to bring affordable and high quality voice communications to people all over the world…"

(Skype.com)

Skype - what is it good for?

- Instant messaging
- Audio Chatting
- File transfer (AV scanned)
- Video chatting
- Skype Out – connecting to PSTN networks
- Skype In – connecting to Skype clients from PSTN networks
- Voicemail
- SMS
- API
- Very very very easy and simple UI
- And much much more

Skype – how does it work ?

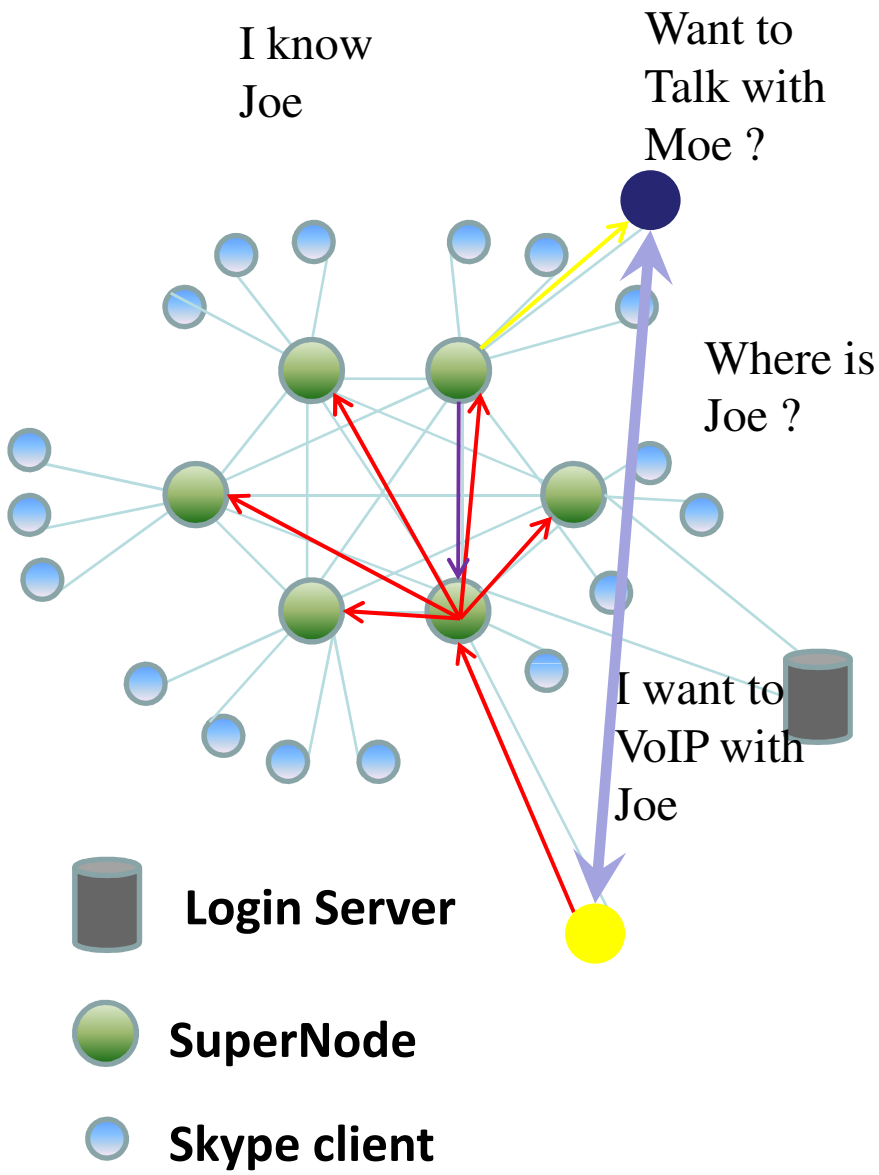
- No one knows
- Closed source
- Built-in Anti-debugging mechanism
- All communication is encrypted

Skype – This is what we do know

- Based on p2p architecture
  - based upon Kaaza p2p architecture
- Proprietary signaling and media protocol
  - Voice/Video calling
  - Instant messaging
  - File transfers
- It can work, almost seamlessly, across NATs and firewalls

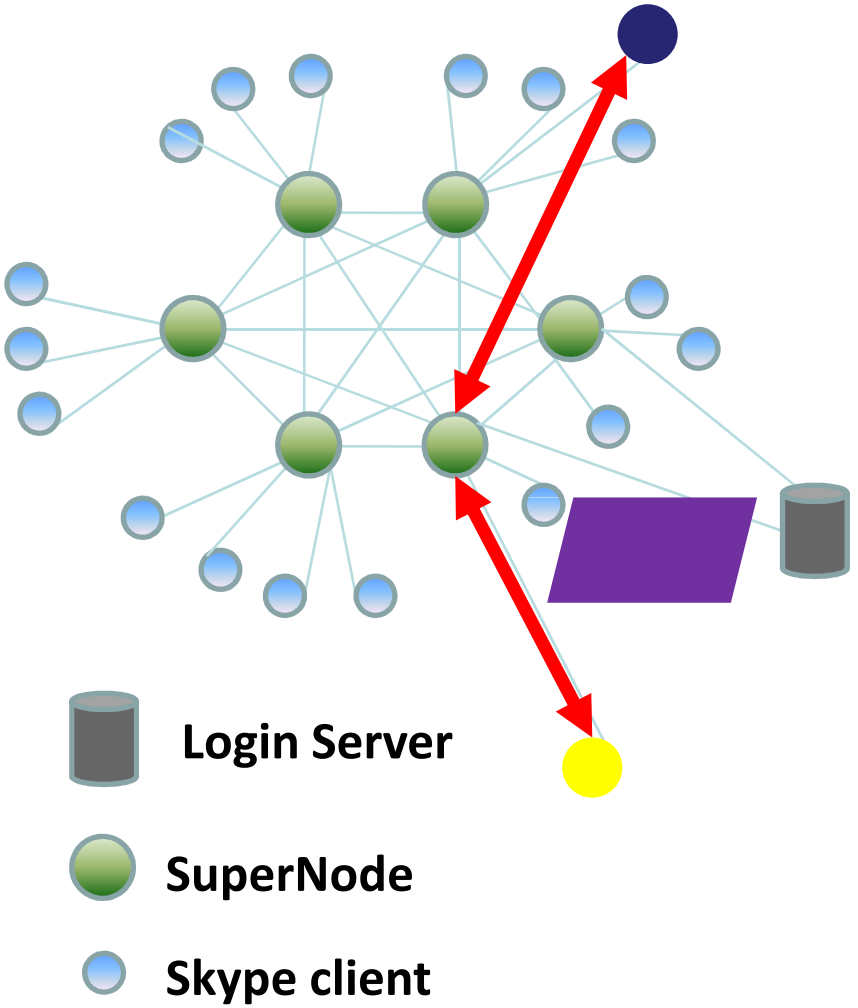
Skype Architecture

- Two Type -
  - Skype Client (SC)
  - Super Nodes (SN)
- SNs manage control between clients
  - Clients will communicate media directly (P2P)
- SNs have full knowledge/access to all the network
- Any client can become a Super Node(SN)
  - No indication to the user



Skype Architecture

- Usually the media is straight client to client
- In case of FW/ any other obstacle –
  - SNs can act as proxy and relay the client communication



SO WHY BLOCK SKYPE ?

End user Perspective

- Skype is great!!!
- Don’t need to configure FW
- Don’t need long lengthy instruction manual
- It works!

Enterprise Security

- Can't see what users send/receive via Skype (encryption)
  - Data leak prevention
- Is there a back door?
- Does it have any spyware / malware ?

For ISP Business

- Want to have the ability to:
  - **Block usage of Skype**

    force usage of their non-free VoIP service
  - **Rate limit the usage of Skype**

Why it is hard to block Skype ?

- Very popular
- Closed source - Obscure
- P2P architecture - no server IP to block
- Random port usage
  - if high ports are blocked, Skype uses port 80 and 443
- It can work almost seamlessly across NATs and firewalls
- All communication is encrypted (including management signaling)
- Skype Inc has implemented many (if not all) of the methods to avoid being blocked

Currently, How is Skype being blocked?

- **Application control** – not 100% effective
  - U3 installation
  - Install/download client while outside the enterprise

Currently, How is Skype being blocked?

- **Block Skype signature**
  - What happens if Skype Inc. decides to change the Encryption scheme and signature every week/day/hour ?
  - What happens if Skype Inc starts adding random bits to packets?
  - Signature – problems: False Positive, heavy processing

Currently, How is Skype being blocked?

- **Block All unknown encrypted data**
  - High false negative

Block Skype by mapping the Super Node network

OUR PROPOSAL

Client interaction with Skype

- When Skype client is installed

  it contacts one of the 7 known Skype Servers (bootstrap Nodes)
- Once connected, a list of Super-Nodes (SN) is saved locally
  - Version 1 – SN list is saved in the registry
  - version 2-2.5 – SN list is saved in and XML (shared.xml)
  - Version 3 – SN list is encrypted/obscured
- The SN list hold up to 200 alternative SNs
- If the client is unable to connect to the any SN (bootstrap SN or the list of 200 SN) it is blocked

The SN list Usage

- Each client is connected to Skype by connecting to an SN
- If the SN to which you are connected to, fails for any reason (reset of the SN machine, severed connection, etc...) → a connection to another SN from the list takes place
- The list is updated regularly (minutes/hours) by the Skype network

Our solution

- Compile a master list of all SNs
  - based upon IP + port usage to avoid false positive
- Continually update it
- Feed the list to the enterprise/ISP FW
- The FW will block access to the SN “black-list”
- Thus, Skype will be blocked/limited within the enterprise/ISP

So how can the SN IP's be harvested ?

- Extract from the shared.xml
- Harvester –
  - Skype Client (SC) (version 2.5)
  - Small application which performs the following steps in each iteration:
    1. Extract the SN addresses and ports from the XML
    2. Flush most of the SN addresses from the list - leaving only specific SNs
    3. Restart the SC and wait until the SN list if filled up again with 200 SN IP addresses and ports
- Each iteration is 2-2.5 minutes

```
<HostCache>
  <_1>62.49.250.140:1280,10</_1>
  <_10>68.147.68.104:53224,4</_10>
  <_100>68.45.77.15:39219,10</_100>
  <_101>71.62.168.61:31743,4</_101>
  <_102>129.74.132.14:46688,10</_102>
  <_103>82.244.65.229:35071,10</_103>
  <_104>69.141.44.82:50911,10</_104>
  <_105>83.89.29.137:15291,10</_105>
  <_106>24.248.199.189:46604,10</_106>
  <_107>24.168.61.141:44091,10</_107>
  <_108>91.139.202.145:27630,10</_108>
  <_109>72.38.56.107:60447,10</_109>
  <_11>69.180.61.247:14166,4</_11>
  <_110>193.10.215.214:18360,10</_110>
  <_111>84.210.75.194:2575,10</_111>
  <_112>81.234.76.222:17943,10</_112>
  <_113>83.30.145.201:40988,10</_113>
  <_114>80.220.86.162:9150,10</_114>
  <_115>204.112.132.44:40130,10</_115>
  <_116>131.247.206.137:39682,10</_116>
  <_117>212.51.199.153:3324,10</_117>
  <_118>216.27.159.47:46698,10</_118>
  <_119>84.209.28.248:16667,10</_119>
  <_12>85.1.35.113:4662,10</_12>
  <_120>24.148.8.54:7963,10</_120>
  <_121>129.79.90.200:44113,10</_121>
  <_122>82.131.1.181:4094,10</_122>
  <_123>87.94.20.170:51238,10</_123>
  <_124>88.103.46.87:31272,10</_124>
  <_125>84.9.176.225:13597,10</_125>
  <_126>70.174.176.212:27223,10</_126>
  <_127>76.189.160.77:5063,10</_127>
```

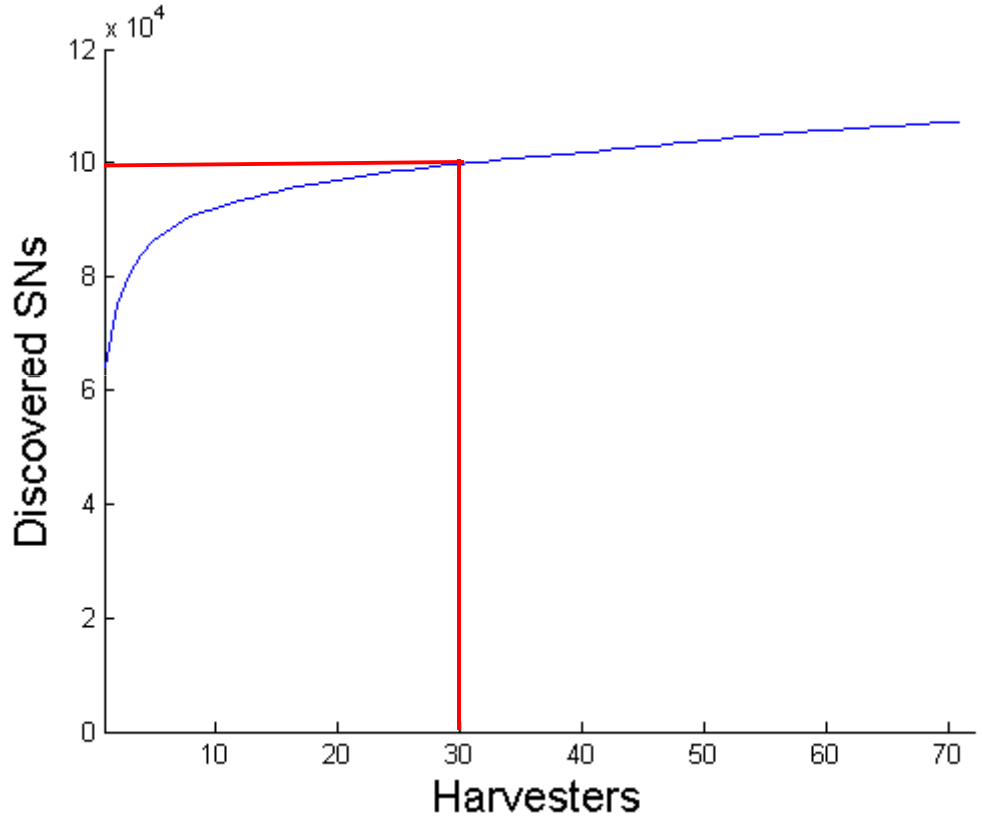
Our Experiment

- Harvesting Cluster – 77 harvesters
  (in several sites: Israel, Switzerland)
- 80 hours (~ 2700 iterations)
- ~ 41.5 million SN IP+port were collected

Experiment results

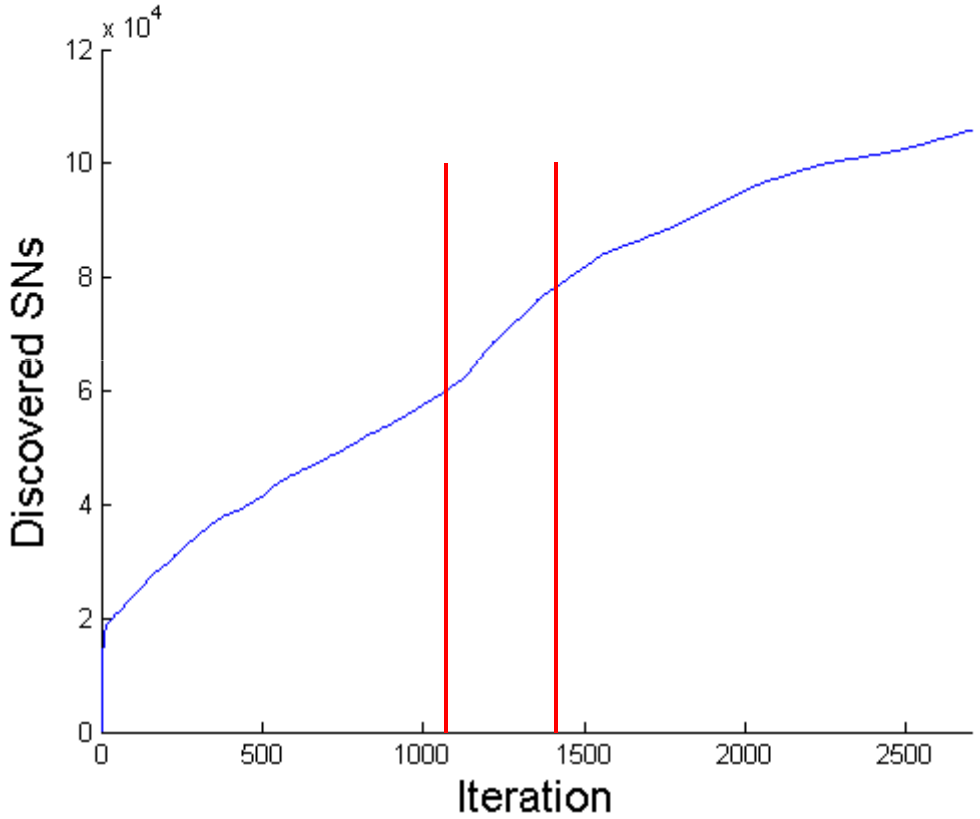
- Over 107,000 unique SN's (IP + port)
- 106,300 (unique IP only)
  - difference is negligible
  - Blocking based upon IP+port is more refined
  - Less false negatives

Number of Harvesters impact on cumulative number of SNs discovered



First 30 harvesters provide the vast majority of SN Discovered

Number of Iterations impact on cumulative number of SNs discovered



More iterations always help discover more SNs
– Skype population constantly changes

Normal office hours in the US

Examining the probability of Blocking Skype

- Connection attempts from within the enterprise can be classified:
  - **Freshly installed Skype Client (SC)** – easily blocked
    - access to Skype Inc hard-coded SNs – there are 7 reported
  - **SC which was installed outside of the enterprise / ISP**
    - Enterprise user who installed Skype at home on his portable computer
    - Returns to the enterprise network after a lengthy period (hours and even days)
  - **User who retrieves a very 'fresh' list of SNs**
    - User runs down to the Café internet hotspot and immediately returns to the enterprise network

Examining the probability of Blocking Skype

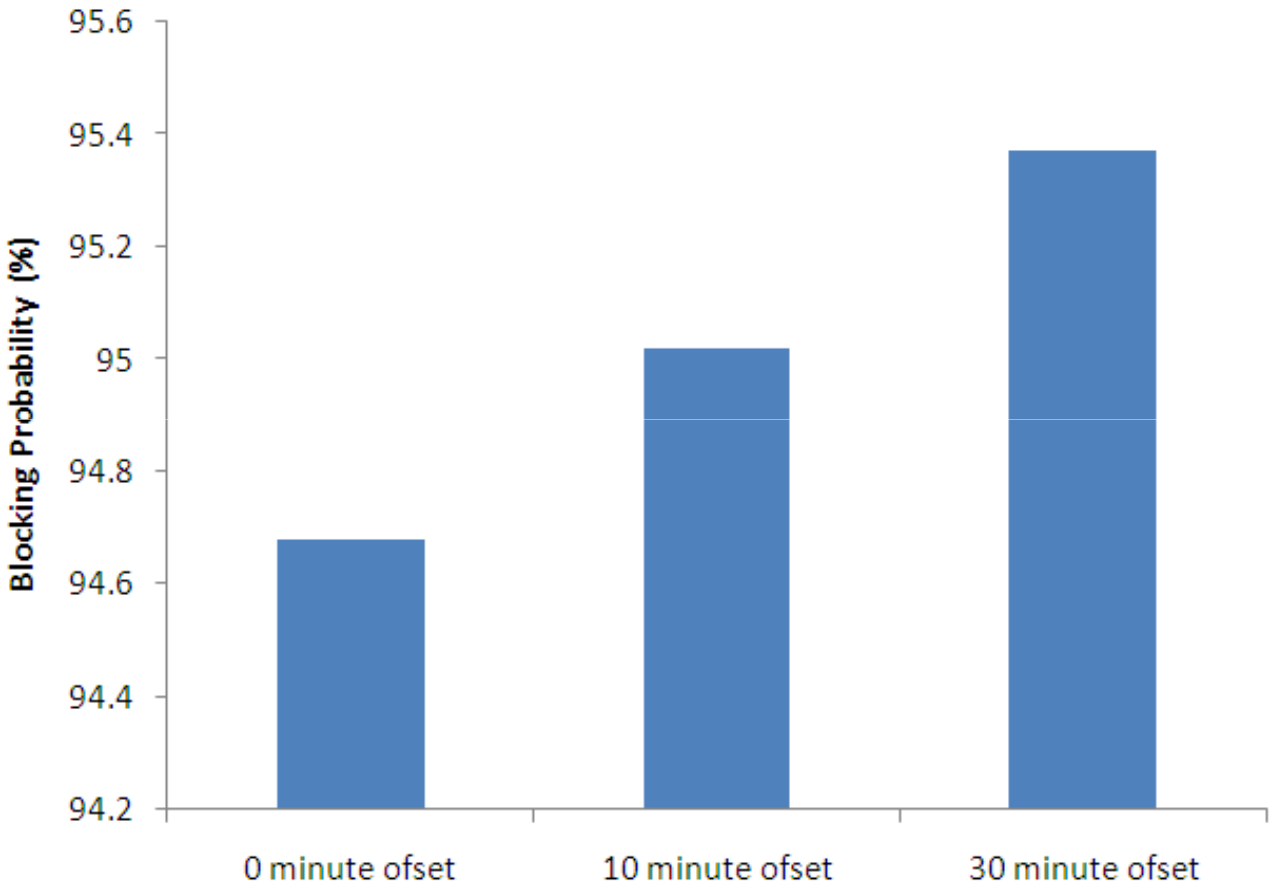
- **Tester Clients (TC)** – SC that simulate the attempts of a regular user to connect to Skype
- TC will simulate the most challenging connection attempt →

  User who retrieves a very 'fresh' list of SNs
- During the experiment, 12 TC performed 240 rounds (each round 20 minutes long):
  - Start the SC and login
  - Stay Idle for 10 minutes
  - Shutdown and wait for 10 minutes

Examining the probability of Blocking Skype

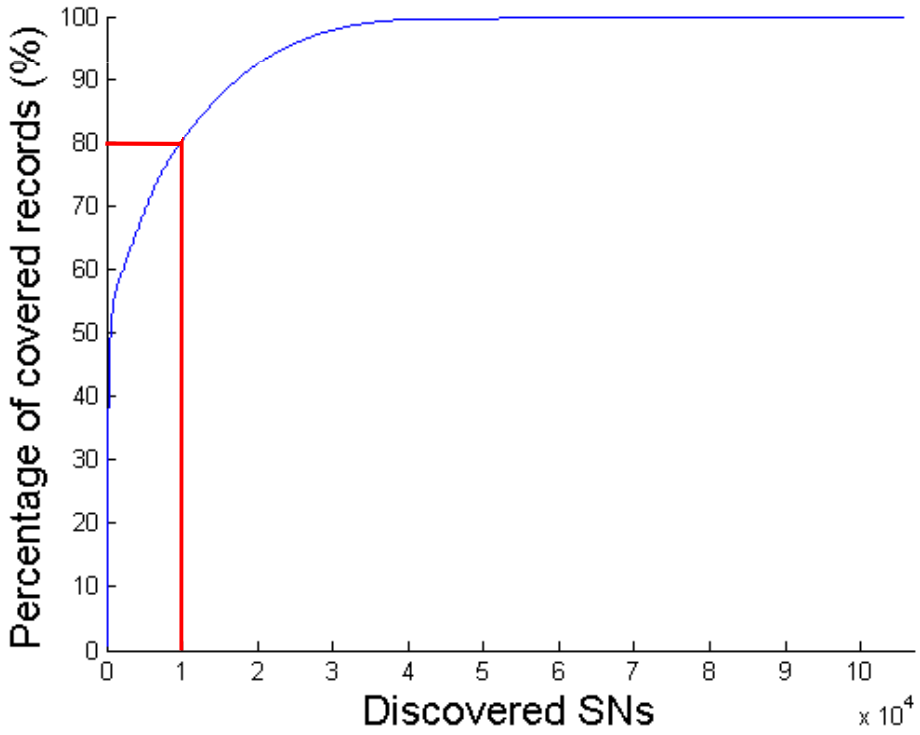
- After each round – check if TC can connect
- Check if the TC SN list contains an SN which was not yet discovered by the harvesters
- We measured this probability of blocking
  - Immediately after the completion of the TC round
  - 10 minutes after each round
  - 30 minutes after each round
- These timeframes represent the time it take the user to return to the enterprise network

Blocking probability as a function of the TC SuperNode list age



The Characteristics of the SuperNodes can it help to block ?

- ***SN Distribution***
  - 10% of the most frequent SNs are responsible for ~80% of the total collected records

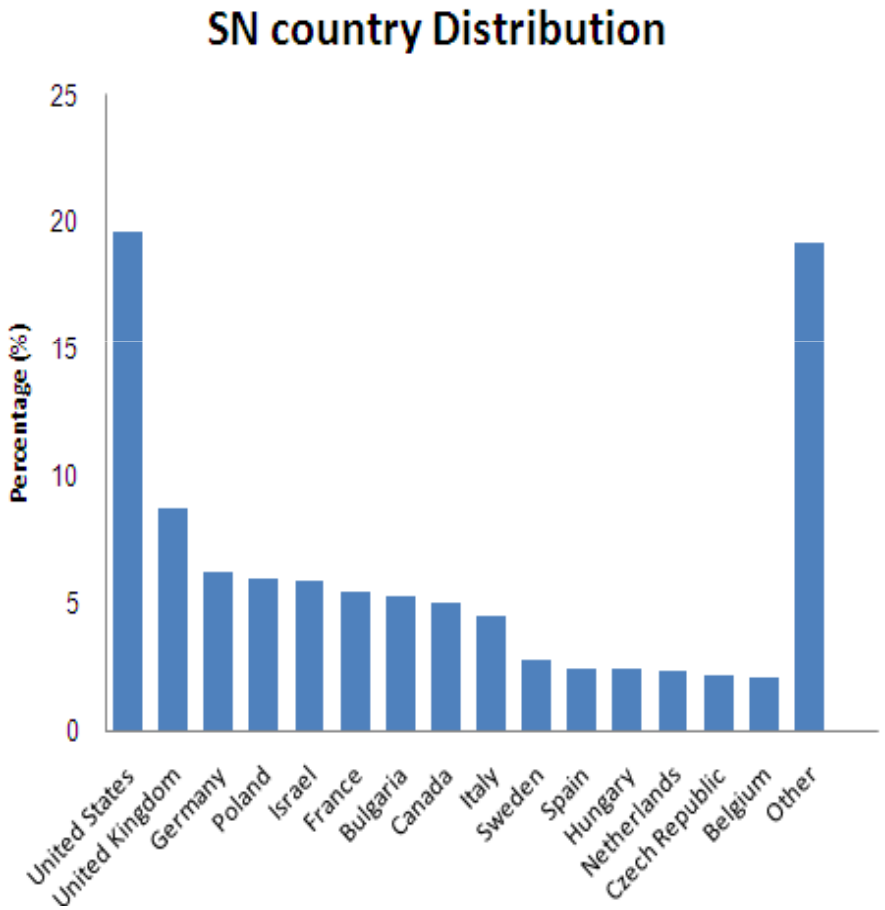


The Characteristics of the SuperNodes can it help to block ?

- ***Port Distribution*** –
  - *random port usage*
  - *The most used port appeared*
    - *0.86% (hardcoded SN)*
    - *0.34% (non hardcoded SN)*

The Characteristics of the SuperNodes can it help to block ?

- ***Geographic Distribution***
  - *No dominant AS*
  - *Several dominant countries*
  - *Similar view from all geographical locations (Israel, US, Turkey, Canada, South Africa, Sweden Switzerland)*

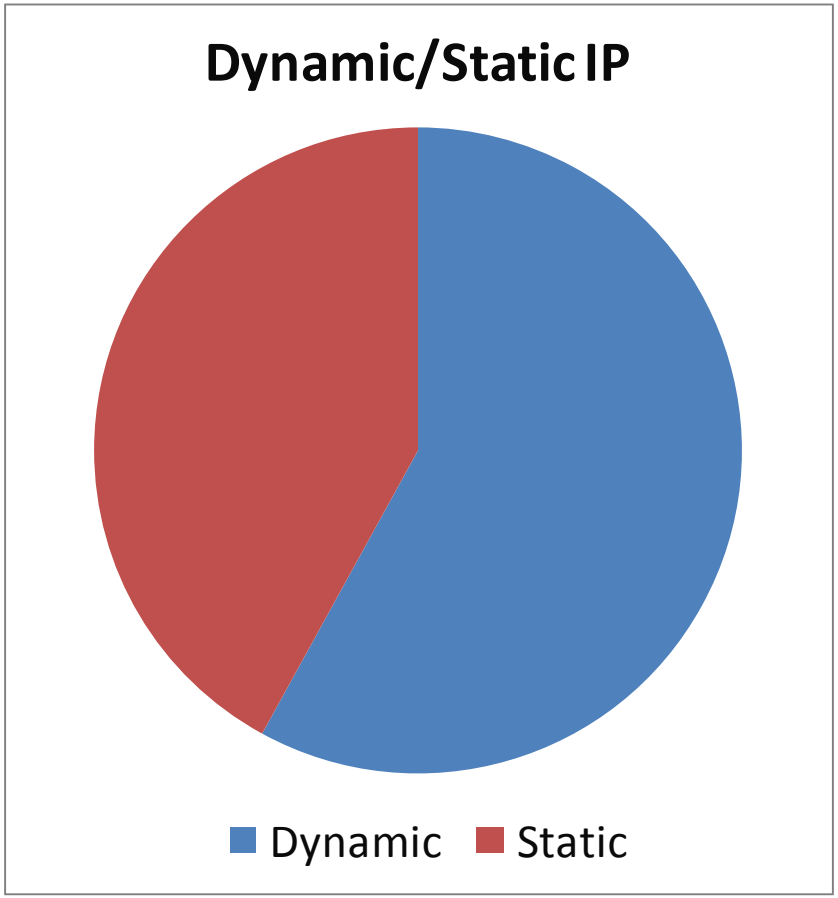


The Characteristics of the SuperNodes can it help to block ?

- ***Dynamic vs. Static IP Distribution –***
  - *58% dynamic*

- ***<u>Impact</u>: Skype needs to consistently update the SNs***

  and also us



Blocking / Rate Limiting

- If the SN network can be mapped it can be:
  - Effectively blocked within an enterprise
  - Effectively limit rate of Skype within an enterprise
    - We use the method to identify Skype SN
      - By identifying the SN we can pinpoint the clients
      - By identifying the clients we can rate limit of their Skype traffic (using signatures, but with less false-positive)

Properties of our Technique

- Scalable (IP + port) packet header and not its content – no need to analyze packet signature
- False positive ~0 (because IP + port)
- Blocking with very high probabilities (above 95%)

Conclusions about Skype/ and general P2P vulnerabilities

- Can Skype bypass our technique?
  - Skype – Version 3.0 Shared.xml is encrypted
  - we can bypass it also
  - Game of give-and-take: speed of SN discovery vs. Skype obscurity
    - In our research we have provided a statistical model to explore this trade-off
- We suspect this vulnerability is not only for Skype but for all p2p topologies
  - Hence, we can use this method to block all p2p protocols

Questions ?

Thank you