

SPIT Mitigation by a Network-Level Anti-Spit Entity

01 June 2006

Bertrand Mathieu, Quentin Loudier, Yvon Gourhant, François Bougant, Mathieu Osty

France Telecom,R&D / CORE / M2I

E-mail: françois.bougant@francetelecom.com

Outline



- Why mitigating SPIT ?
- Some Anti-SPAM solutions ...
- Proposal for an algorithm to detect SPIT in the network
- Anti-spit entity : where to place it ?
- Proof of concept : description and basic performance Tests
- Conclusion



Why mitigating SPIT ?

→ SPIT : Spam over Internet Telephony

→ An heritage of spam

- ▶ Unsolicited e-mail diffusion, mostly commercial (>60 % of e-mail traffic)
- ▶ Efficient server based filtering solutions (> 95 % reliability)

→ SPIT opportunity

- ▶ need for using new transport medias
- ▶ VoIP is getting popular

→ Risks

- ▶ VoIP services almost free
 - *opportunity for Illegal and aggressive telemarketing*
- ▶ Ability to perform numerous, automated, simultaneous calls from a single machine
 - *Potential network & VoIP platforms overload*
- ▶ Negative feedback from VoIP users

Some anti-SPAM solutions ...



→ Blacks and white lists

- ▶ Identify SIP users, allowed or blocked to make calls
- ▶ - *updating based on user experience : Limited efficiency (forgable SIP alias) and reactivity (up to several days to identify a SPITer,*
- ▶ - *very easy to create new VoIP user accounts : reputation based system limitation*
- ▶ - *User authentication required for reliability*

→ Turing tests

- ▶ Should be based on pre-authentication of the caller as a human being
- ▶ - *imply human action before calling → bad impact on user experience*

→ Content Filtering tests (e.g. bayes)

- ▶ For SPIT, should be based on voice recognition, keyword detection
- ▶ + *needed for high reliability*
- ▶ - *high memory / CPU required, unrealistic to perform in real-time today*

→ **Need for a simple, transparent solution for anti-SPIT**



Proposed Algorithm : key elements

- Uses network, transport and application level elements
- A SPITer is identified with the IP address on network level
 - ▶ SIP alias : unreliable except when authenticated (but not widely deployed)
 - ▶ IP address in VIA field : reliable but applicable only into local networks
- Each positive analysis of a given criteria does not imply spit by itself
 - ▶ Each detection analysis has a priority : p_i
 - ▶ The sum of them (spitLevel) can classify the call as a spit if above a given threshold

$$\text{spitLevel} = \sum p_i a_i \quad \text{for every } i \text{ where } a_i = 1 \text{ when positive}$$

and $a_i = 0$ when negative

- The threshold is a parameter according to the sender's profile and category (known as frequent spitter, rare spitter, never classified as spitter before...)
- Then a context is handled for each user and memorized during a given sliding window (1 minute, 2 hours, 24 hours, etc.)
- The decision to classify the call as a SPIT depends on the call's analysis, the user's category and the user's history

Proposed Algorithm : the 5 criteria for detection

- ➔ Number of received error message
 - ▶ Analysis of the number of error messages, replied to the Invite messages (404, 603, 486).
 - ▶ If a lot, it means that called user aliases are forged
- ➔ Automated Logic
 - ▶ Can detect if the sender uses a directory : e.g. dupon@x.net, dupond@x.net, dupont@x.net, etc.
- ➔ Simultaneous calls
 - ▶ Above a threshold, it means that a "machine" generates the calls since a human can not handle a lot of parallel calls.
- ➔ Call duration
 - ▶ If a lot of calls have the same duration => can surely be spit (automated messages)
 - ▶ Valid if messages are recorded on vocal boxes, *not if the end-user terminates the call before the end*
 - ▶ Detected at the end of the call, then the spit is already transmitted but may help to "blacklist" the sender for future calls.
- ➔ Call Bombing
 - ▶ Many calls from different users to a single user at the same time : new denial-of-service attack



Proposed Reactions

- Limitation of the number of calls
 - ▶ For smaller spitters
 - ▶ Possibility to allow the user to only send 5 calls per hour (it is a parameter)

- Temporary Blacklisting
 - ▶ For larger spitters
 - ▶ Possibility to block calls from the user during a given period (e.g. 1 hour)

- Call redirection
 - ▶ Redirects the call to an automat
 - ▶ Can reply negative to the calls
 - ▶ Can propose an anti-virus checking (since the spit can be generated by a malicious program running on the user's device, which the user is not aware of).

- Notification
 - ▶ Redirects the call to a server for classifying users or for statistics purposes



Anti-spit entity : where to place it ?

1. Anti-Spit Entity (ASE) into SIP proxy
 - ▶ + *no issue related to traffic encryption*
 - ▶ - *impact on proxy performance during call bursts*
 - ▶ - *VoIP services based on distributed P2P networks (IETF P2P SIP BOF) unsupported*
 - ▶ - *proxy overload is not avoided*
2. Anti-Spit Entity into generic DPI
 - ▶ Used to analyse all IP traffic
 - ▶ + *can be used for other kinds of detection and mitigation (spam, DDoS attacks, etc.)*
 - ▶ - *increase DPI complexity*
 - ▶ - *traffic encryption*
 - ▶ ! *Potential regulatory issues in case of mitigation : what business model ?*
3. Router / DPI + External Anti-Spit Entity
 - ▶ ASE analyses VoIP traffic only
 - ▶ IP traffic detected as VoIP by router or generic DPI and forwarded to the ASE. ASE reinjects legitimate VoIP traffic into network.
 - ▶ + *scalability (N router/DPI → 1 anti-SPIT entity)*
 - ▶ + *no regulatory issues : ASE and router / DPI managed by different actors*
 - ▶ - *if router : impact on network topology*
 - ▶ - *traffic encryption*

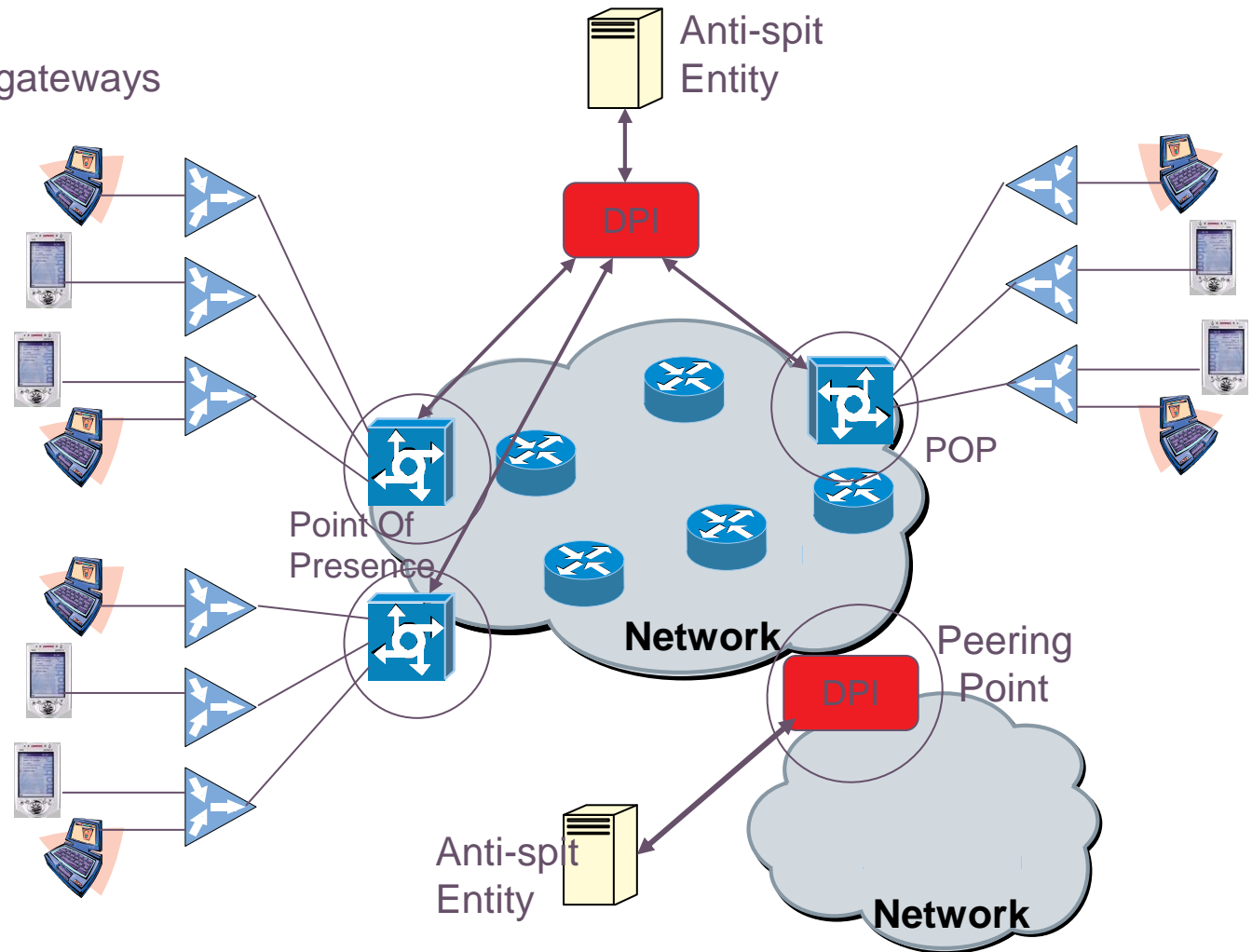
Focus on scenario n°3 with DPIs



➔ DPIs located at one or several strategic points

- ▶ Residential / enterprise gateways
- ▶ Aggregation links
- ▶ Peering links

➔ Anti-spit entity receiving VoIP traffic from DPIs





Deep Packet Inspector : some key points

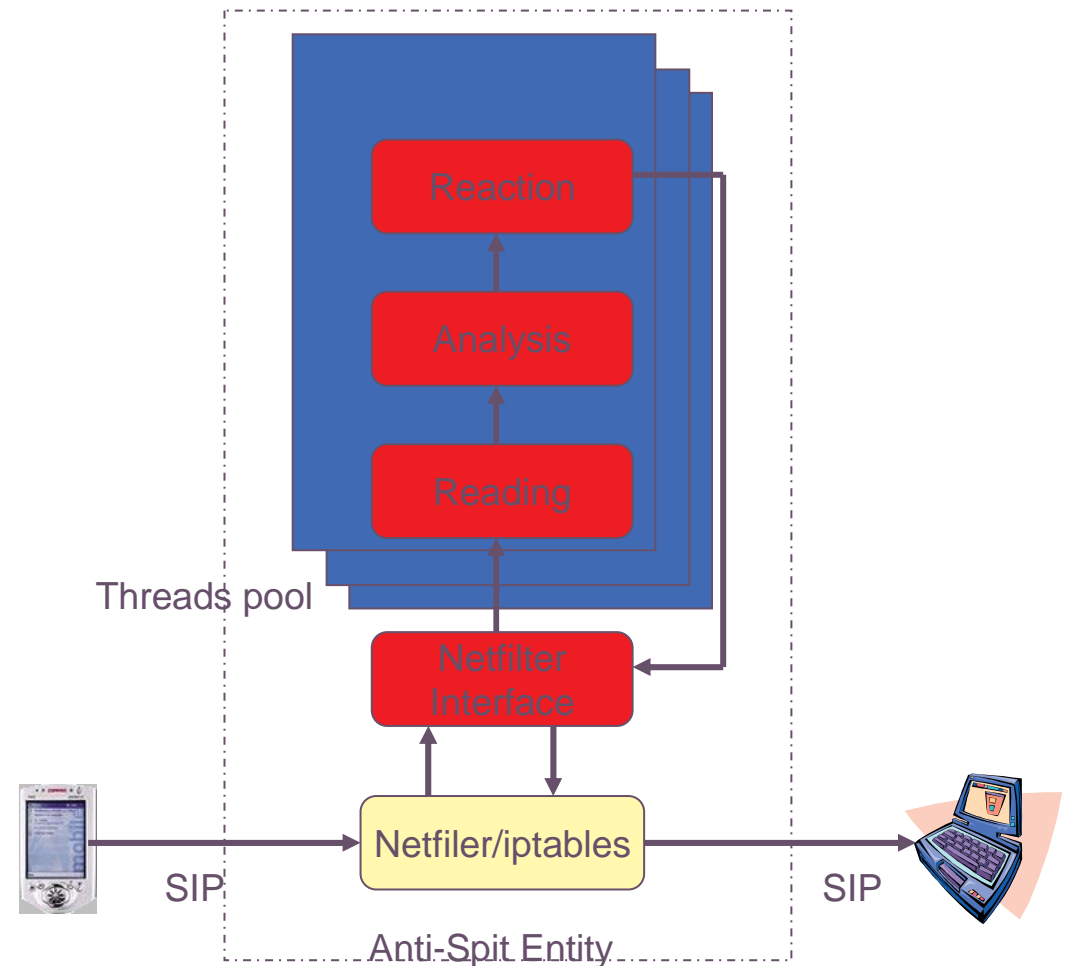
- Real time, layer 7, stateful analysis of IP traffic
 - ▶ 500+ predefined protocols (OSI layers 2-7), with destination port and/or signature
 - ▶ For some protocols, extract network to application level parameters (e.g., VLAN ID, MPLS tag, source/dest. IP address, SIP URI, RTP dest.port, codec)
 - ▶ No switching or routing functions
 - ▶ Frequent protocol signature upgrades
 - ▶ Redirection
- Hardware & integration
 - ▶ multiple FPGA / processors
 - ▶ Mostly external boxes "on the wire" or connected to mirrored links (monitoring), a few internal to switch/routers
 - ▶ External management platform
 - ▶ Integration to OSS (dynamic IP addresses)
- Average performances
 - ▶ Number of subscribers : 100 000
 - ▶ Number of simultaneous sessions : 1 000 000
 - ▶ Number of incoming sessions / sec : 10 000
 - ▶ Max bandwidth (throughput) : 3.6 Gb/s



Proof-of-concept : description

- Written in XML and Java on Linux
- Intel pentium 4 platform, 1 GB RAM
- Reaction Module
 - ▶ Three types : forwarding, dropping, redirecting to a server (not tested)
- Analysis module
 - ▶ Performs analysis y applying the described algorithm
- Reading Module
 - ▶ parses user information (type of message, addresses SIP source, destination etc) from the received packets
 - ▶ Update context information (as a tracker)
- Netfilter Interface
 - ▶ filters VoIP from the received traffic and transmits it to Reading module
- Netfilter/iptables

France Télécom / R&D Division





Basic performance tests

→ Goal

- ▶ Basic performance estimation ... on a basic machine
 - max. number of calls supported by the ASE in analysis mode only, with normal, constant traffic
 - latency introduced by the process
 - RAM used

→ Test bed

- ▶ 1 SIPp instance / 1 machine to make calls, static IP address
- ▶ Anti-spit entity + router on same machine
- ▶ 1 SIPp instance / 1 machine to receive calls, static IP address

→ Results

- ▶ 80 calls / sec max
- ▶ Average processing time (analysis and decision) 5 ms.
- ▶ ~ 300 MB RAM used

→ Analysis

- ▶ estimation of traffic for a VoIP provider on FT network : 40 to 60 calls/s → **OK**
- ▶ Call establishment time : 6.3 s → 5 ms more : **OK**



Conclusion

- **SPIT is a risk for VoIP services**, as SPAM for messaging
- Proposed solution : **control SPIT before it is served by VoIP proxies**
 - ▶ + *limit impacts on proxy performance*
 - ▶ + *Reduce VoIP useless traffic*
 - ▶ + *can be fit into a global network security system based on DPIs*
- Demonstrator : as a **proof-of-concept**, partially demonstrates the feasibility to implement a network-based anti-spit solution
- Future Work : **Improve reliability to detect 80 % of SPITs**
 - ▶ **Solve issues related to SPITer identification on network level, with dynamic IP address + NAT ... or Authenticate identity in VoIP protocols**
 - ▶ **optimize policy** (threshold, sliding period, etc.) for each category of VoIP customers (VoIP provider / residential / enterprise, light / heavy users, etc.) with real VoIP traffic and limit false positives
 - ▶ **Improve criteria** for detection (ex : call duration)
 - ▶ **Solve issues related to traffic encryption and detection of voice over P2P networks from DPIs**



Questions / Comments

