Stop the Flood – Perimeter Security– and Overload–Pre-Evaluation in Carrier Grade VoIP Infrastructures

Michael Hirschbichler, Joachim Fabini, Bernhard Seifert, and Christoph Egger

TU Wien, Institute of Telecommunications
Favoritenstr. 9-11/E389
A 1040 Wien, Austria
{michael.hirschbichler, joachim.fabini, bernhard.seifert,
christoph.egger}@tuwien.ac.at

Abstract. With the upcoming introduction of the Session Initiation Protocol to carrier grade telecommunication infrastructures, the threat of attacks is increasing massively. Multiple types of unsolicited communication, like high and low rate Denial-of-Service attacks as well as Spam over Internet Telephony driven by Botnets will be an upcoming risk for all telecommunication operators.
In this document, we introduce an enhanced Session Border Controller which is able to detect high-rate DoS attacks and which will mark all forwarded requests with a value indicating the "quality" of the request. This value, which we denote as "dropability", reflects the effort the system has already invested for this request. This dropability-value depends amongst other presented factors on the spam-probability and the economic- or QoS-effect of this request.
This introduced value supports overloaded core-components to decide with minimum processing effort, which requests to drop first and which requests have severe effects on the customers perception or the economic income of the carrier.

Keywords: SIP; DoS; Spam; Carrier Grade Networks: Overload Control

1 Introduction

Standardization bodies are working on solutions both against Spam over Internet Telephony (SPIT) and for implementing overload control communication mechanisms. In complex carrier grade environments with multiple hops and components, there is a strong requirement to keep overload at the perimeter in order to keep the core clear of unnecessary load.

In this document, we merge the ideas of IETF standardization activities and extend these by developing an extended perimeter security component named "Session Border Controller - Advanced" (SBC-A). This multistage-component

prevents from flooding denial-of-service (DoS) attacks on one hand and, on the other hand, it marks all requests for later overload-control dropping algorithms. By focusing all qualification mechanisms on the entry-point of the core infrastructure, this component unloads the core and supports prioritization of important requests during high-load periods.

In the first part of this document, we introduce the SBC-A and a new Session Initiation Protocol (SIP)-header `X-dropability` which is added by the SBC-A to communicate the "quality" of a request to the protected core. For example, INVITE-requests of authenticated and trusted customers have a high quality (and a low `X-dropability`-value), probable SPIT-requests of unknown callers have a low quality. In overload state, the affected components are able to drop low-quality requests earlier than high-quality requests by considering the `X-dropability`-header.

In the second section, we show, how this `X-dropability` value can be integrated in current overload-control communication standards and standard drafts.

The last section summarizes the concept and presents further steps verifying our approach.

With the presented technique, the processing load on carrier grade core components can be reduced to the benefit of handling more goodput.

2 Related Work

SIP, as underlying protocol of the presented overload control technique is specified by Schulzrinne et. al. in RFC 3261[12]. The Session Border Controller and its provided functionality is discussed in RFC 5853[3].

In RFC 5390[10] J. Rosenberg defined different causes of overload and compared their impact on SIP-operated infrastructure.

For solving the overload threat, ETSI/TISPAN has introduced the highly complex transprotocol overload control protocol *GOCAP* in standard TS 283. 039.2-4 [1]. A more generic approach is the informational IETF standard "Design Considerations for Session Initiation Protocol (SIP) Overload Control" RFC 6357[4]. Based on this standard and improved by this paper, the IETF sip-overload working group develops a hop-to-hop overload control protocol for SIP components[2]. This protocol communicates overload states to neighbor upstream hosts. The used dropping and rate-limiting algorithms are proposed in the early-state draft [7].

Noel and Johnson compare in [6] multiple overload control algorithms with the finding to throttle the overload as close to the source as possible. In the here discussed carrier grade infrastructure, this location is the core perimeter where we will place the SBC-A.

Preventing (low-rate) DoS-attacks on SIP application layer and the need of SIP- and RTP-aware *perimeter security* is presented by Ormazabal et. al. in [8]. Using the Hellinger distance for detecting stealth flooding is discussed in [15]. Methods for detecting unsolicited communications are defined in RFC 5039[11]

and the marking of SIP-messages with spam-score headers are proposed in the (outdated) IETF-draft of Wing et. al. in [16].

3 *Session Border Controller - Advanced*

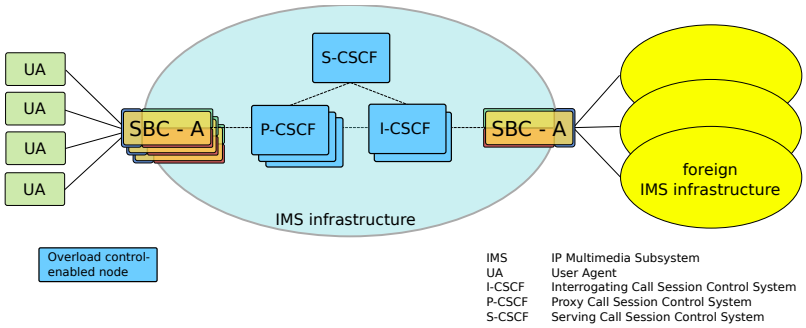

Fig. 1. Multiple SBC-A' protecting an oc enabled IMS infrastructure both from accessnet as also from peering partners

The term SBC is unspecific and not standardized but meanwhile used by many VoIP providers as a (often) proprietary border security function. At this edge node, the network policies are usually enforced using the SBC. The RFC 5853[3] summarizes these common functionalities and highlights three SBC-specific areas: " a) perimeter defense(...) b) functionality not available in the endpoints (...) and c) traffic management (...)". We extend both area "a)" and "c" by adding overload control-preevaluation and -marking and name this extended SBC *Session Border Controller - Advanced (SBC-A)*.

The SBC-A consists of multiple nested stages (see figure 2) with an increasing grade of packet inspection and an increasing load per request (*lpr*). We first define two categories, *Unknown Relationship (UR)* and *Known Relationship (KR)*, whereas the latter is split into *Trusted Relationship* and *Untrusted Relationship*. An incoming request passes the two categories (*UR* and *KR*) and the three nested analysis stages:

1. Denial of Service (DoS)-detection and -protection
2. Unsolicited Communication (UC)-detection
3. Overload Specific Request Qualifying (OSRQ)

In the next subsections, we describe these three stages and show the results in overload-control marking.

3.1 Denial-of-Service–detection and –protection

To keep the overall load in the next steps low, we block incoming high-rate DoS-attacks first, using techniques proposed and discussed in [8] (figure 2 [1]).

Due to the high complexity of SIP-operated infrastructure, successful blocking of high-rate attacks does not stop the problem of overload in general.

RFC 5390[10] defines six different reasons for overload, which can be grouped into a) operator initiated overload ("Poor Capacity Planning", "Dependency Failures" and "Component Failures") and b) external initiated overload ("Avalanche Restart of clients", "Flash Crowd of multiple users simultaneously creating a call" and "DoS"[10]). From this list, only DoS from one single source can be blocked by simple high-rate attack preventing mechanisms. All other reasons for overload stem from multiple sources, which create in sum the overload. With the proposed algorithm, the system is capable to mark suspicious and/or unimportant requests for earlier dropping than other requests.

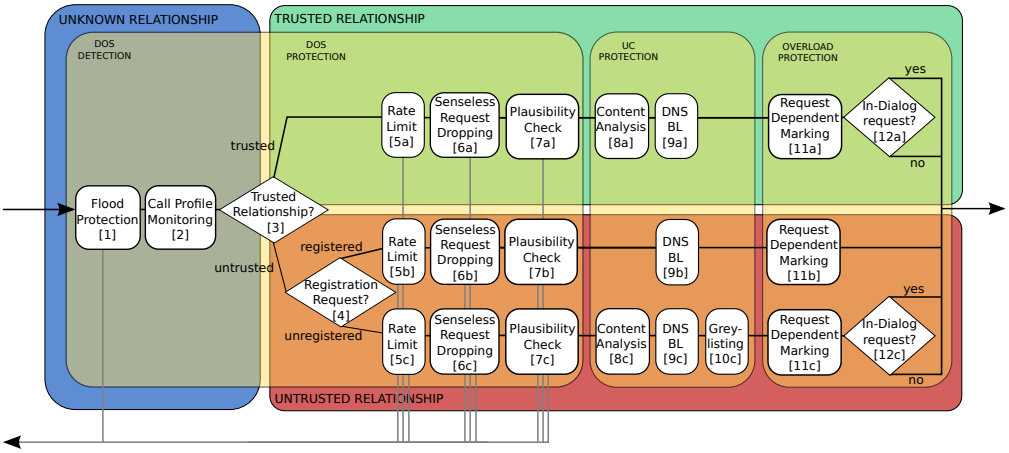

Fig. 2. Detailed step-by-step processing of SBC-A

For this reason, the next step compares the current incoming traffic profile with already observed and recorded daily call-profiles (see figure 3 and figure 2 [2])). If there is a significant difference between the current call amount and this profile, all requests passing this SBC-A are marked with an increased `X-dropability`-header and a factor, by which the current profile exceeds the expected profile. Requests, which arrived during a timespan with common load will also be marked with a `X-dropability`-header and a value of "0".

Using the Hellinger distance as proposed in [14] and [15] will additionally assist to mark stealthy flooding requests.

The presence of a `X-dropability`-header within the signaling flow notifies the other SIP components about an SBC-A within the signaling path.

In listing 1.1, we present the RFC 3261 INVITE-request extended by the dropability factor 2

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc3.atlanta.com;branch=z9hG4bKnashds8
```

```
X-dropability: 2
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
```

Listing 1.1. INVITE request (w/o SDP) with added `X-dropability`-header

In opposite to the high-rate flood protection, the requests are not dropped (or canceled).

As the irregular increased load can not clearly be allocated to evil attacks, the requests are instead marked and forwarded to the next processing stage. With this approach, we prevent false-positives and the protected core can block suspicious requests only in case of overload.

In the next step of DoS-protection, the SBC-A decides, whether the incoming request is part of a *trusted* or an *untrusted relationship*. Here, an incoming request is assumed as *trusted*, when the sending socket of the request is already actively registered (fig. 2 [3]).

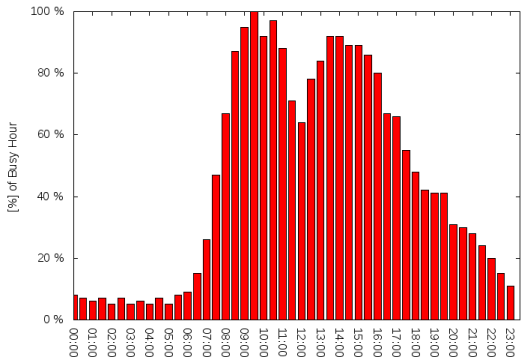

Fig. 3. Example request profile (INVITE requests) in a carrier grade VoIP-infrastructure. The diagram represents the call-setup distribution over a whole working day. Equivalent profiles are used for all other relevant SIP requests to detect abnormal request load.

By differentiating between trusted and untrusted relationships, the system can handle new requests distinctly and offer them various priorities. In our architecture, we propose to limit the allowed amount of system load for processing untrusted requests in high-load-situations to a fixed, configurable value. All requests exceeding this limitation shall be terminated with a 5xx-final response (fig. 2 [5a-c]).

As a result, the requests from trusted relationships (which are more valuable to the operator) have guaranteed processing resources even in high-load-situations.

Most of the "untrusted" requests are initial registration requests of well-known customers. Hence we define the next differentiation that depends on the type of request. Requests that are received from an untrusted socket and that are REGISTER-requests must be handled with higher priority than other requests (fig. 2, [4]).

After this point, we propose that the operator includes simple request-dependent filter rules: For instance, SUBSCRIBE-, NOTIFY- and OPTIONS-requests from unregistered sockets can simply be canceled with a "403 Forbidden" right there, as they will never result in a positive response from the core network (fig. 2, [6a-c]).

The next step in DoS-protection is the *plausibility check*. Following predefined regular expression-patterns, we decide whether requests are compliant to RFC 3261 and operator-specific security policies (fig. 2, [7a-c]). Depending on the security or local regulatory settings, we either drop suspicious requests immediately or increase the dropability parameter (see listing 1.2).

```
INVITE sip:bob@biloxi.com SIP/2.0
..
X-dropability: 3
...
```

Listing 1.2. increased `X-dropability`-header

This plausibility step is located subsequent to the trusted-relationship delimitation, as we want to assure for performance reasons, that operations with a high *lpr* must be executed after operations with low *lpr*.

After the DoS-detection and protection stage, the system has divided all requests into a preferred trusted and a non-preferred untrusted relationship and marked all "suspicious" requests with a `X-dropability`-value larger than 0.

3.2 Unsolicited Communication Detection-Marking

The next stage in the SBC-A pipeline is the *Unsolicited Communication* (or *SPIT*)-analysis. For highest transparency, we focus mainly on non-intrusive SPIT analysis where the procedures differ between trusted relationship, untrusted relationship–REGISTER-request and untrusted relationship–other-requests.

From this stage on, we do not drop suspicious requests at the SBC-A (due to the risk of getting false positive results) but we increase the `X-dropability`-header-value by a value according to the evaluated SPIT-grade.

In the outdated IETF-draft[16], Wing et. al. propose an additional SIP-header for marking SPIT-suspicious messages. We decide to merge this header with our `X-dropability`-header as both are aiming at the same goal. Here – as a side-effect to overload-control – the `X-dropability`-header could also be used by the callee to decide whether he wants to accept a specific request or not. In the following we detail on three variants of UC detection:

- **UC marking for requests from trusted relationships**

In this paper, we propose to use two explicit and one implicit UC-detecting techniques: First, we analyze the content and the payloads of all requests for terms like "Rolex", "Viagra" or "Cyalis", equivalent to mail-spam-filtering. If such terms are found, the `X-dropability`-header-value is increased(fig. 2, [8a,c]).

As second approach, we adopt the DNS blacklist technique presented in [5] and compare all IP-addresses in the requests with these lists. Although the system knows that the socket is trustworthy regarding authentication and authorization in SIP universe, it cannot guarantee, that the customer's infrastructure is not misused as SPIT-proxy (fig. 2, [9a-c]).

The third UC-detecting technique is the whitelist-test proposed in RFC 5539. This test is implicitly solved by using the trusted socket. All requests arriving from a known socket have been tested against the home subscriber server before and are implicitly more trustworthy than other requests.

– **UC marking for REGISTER-requests from untrusted relationships**
The REGISTER-request is the most intrusive request when creating low-rate DoS attacks. Inherently, a REGISTER request creates the highest load in an infrastructure as it traverses a high number of components to the location database (e.g., in IMS the Home Subscriber Server (HSS)).

A REGISTER request is not able to transfer SPIT messages at all, but we can use SPIT-qualification methods to verify the plausibility of an incoming REGISTER-request. When using the DNS-whitelist and -blacklist method, we can check if a REGISTER request arrives from a socket which has sent an successful initial REGISTER request before. Additionally, we can check, if a REGISTER request arrives from a socket, which is on the black-list for sending unsolicited messages before.

According to the result of these DNS-lookups, the `X-dropability`-header-value is increased.

– **UC marking for all other requests from untrusted relationships**
These requests, arriving from an unregistered host are the most untrustworthy requests.

During our real-live measurements on the A1overIP-core[1], we noticed that only about 1% of all valid call-setup requests are from a foreign domain directed to the local domain.

This small number of requests must be inspected carefully, as this is the main entrance for possible SPIT and (next to REGISTER flooding) the main threat for low-rate attacks to the core infrastructure.

Here, the proposed SBC-A uses content analyzing techniques (in addition to DNS-whitelist and -blacklist analyses and regarding the request type). Additionally, we propose to move away from non-intrusive tests to the intrusive greylisting-test[9] for INVITE-requests (fig. 2, [10c]). Using greylisting, a call-setup attempt is canceled by the SBC-A with a "486 Busy Here"-response, pretending that the callee is currently in another call. The caller

[1] A product of A1 Telekom Austria, the largest austrian mobile and fixed net operator, `http://www.a1.net`

Request from *Trusted* Socket		**Request from *Untrusted* Socket**	
Request	**Rank**	**Request**	**Rank**
ACK	1	REGISTER	9
BYE	2	ACK	10
INVITE	3	BYE	11
REGISTER	4	INVITE	12
MESSAGE	5	MESSAGE	13
UPDATE	6	UPDATE	14
NOTIFY	7	NOTIFY	15
SUBSCRIBE	8	SUBSCRIBE	16

Table 1. The `X-dropability`-header is increased by the value of the column "Rank" to enable the upstream neighbor of an overloaded component to drop "invaluable" requests first

ID is stored temporarily and if the caller retries, the call is forwarded to the called party. This technique based on Turing tests is used regularly in Email-spam protection infrastructures.

Splitting the request analysis into three categories supports the system to handle requests of differing quality distinctly and helps – even in case of local SBC-A system-overload – dropping the untrustworthy and invaluable requests first.

3.3 Overload Specific Request Qualifying (OSRQ)

The IETF draft "Session Initiation Protocol (SIP) Overload Control"[2] proposes an overload control mechanism, where the upstream neighbor reduces the load sent to the overloaded downstream component. The drop-rate or alternatively the rate limitation is defined by the overloaded component and communicated to the upstream neighbor.

By default, there is no standardized differentiation between the different request-types. The fact that for example, a dropped ACK-request as part of an INVITE dialog may produce more load through retransmission is only considered informational. Instead, the upstream host is free to decide which requests should be dropped and which should be forwarded.

Our proposal is to support this upstream host deciding, which requests are "cheap" to drop and which requests are "valuable".

Hereby we propose – after "DoS detection and protection" and "UC detecting" – to qualify the type of request as additional input value for our `X-dropability`-header. For this reason, we suggest in table 1 a ranking of requests (as defined in RFC 3261 ff., fig. 2, [10a-c]) and observe, if they are part of an already established dialog (fig. 2, [12a,c]).

The sorting of requests in table 1 considers the request-priority. The most important requests are the in-dialog-requests "ACK" and "BYE". "ACK" concludes a nearly completed call-setup, whereas the BYE-request finishes calls and

reduces the workload on an infrastructure. The INVITE-request is expected by the customer to be handled in soft-realtime for QoS-reasons and must also be handled with high priority.

The REGISTER-requests keep an existing registration alive. If they are dropped, the customer premises equipment (CPE) is unreachable for incoming requests, which decreases customer satisfaction.

The remaining requests are text-based requests and can be ranked equally: they do not have to be handled in realtime and delays of multiple seconds are acceptable. Here, we propose to prefer the MESSAGE-request because of its revenue generation when the message is sent over the providers SMS gateway to the CS environment.

For requests arriving from an untrusted socket, the REGISTER request must be handled prioritized: new REGISTER-requests arrive from potential customers and must be ranked higher than other messages arriving from unregistered sockets.

On all requests, which will be forwarded to the core network, we propose to add the rank-value from table 1 to the `X-dropability`-header. Considering the INVITE-request from listing 1.1, we increase the `X-dropability`-header by 3 if the request arrives from a registered socket, respectively by 12 if the request arrives from an unregistered socket.

3.4 Forwarding to the Next Hop

After the three stages of analyzing and marking in SBC-A, the initial INVITE-request of listing 1.1 appears now like in listing 1.3:

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP sbc-a.biloxi.com;branch=z9hG4bKd84ks2
Via: SIP/2.0/UDP pc3.atlanta.com;branch=z9hG4bKnashds8
X-dropability: 14
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
```

Listing 1.3. INVITE-request after SBC-A analysis

This request is forwarded to the next SIP-downstream hop of the carrier grade infrastructure for further SIP processing.

4 Integrating SBC-A marked Requests into existing Overload Control Infrastructure

The standard RFC 3261 defines only minimum overload- and congestion-control. Overloaded components should cancel incoming requests by responding with "503 Service Unavailable"-responses and optional retry timers. The informational RFC 6357 proposes to avoid local overload control and to locate the actuator responsible for blocking the overload situation at least one hop upstream to the overloaded system.

For satisfying this need, the current IETF draft "Session Initiation Protocol (SIP) Overload Control"[2] introduces the "overload control" parameter. This *"oc"*-parameter, added to the Via-header of a SIP message, communicates the desired traffic-reduction to the neighbor upstream node (see listing 1.4). The current draft does not define the supported algorithms in detail.

```
SIP/2.0 180 Ringing
Via: SIP/2.0/TLS p1.example.net;branch=z9hG4bK2d4790.1;
  oc=20;oc-algo="loss";oc-validity=1000;oc-seq=1282321615.782
  ...
```

Listing 1.4. a *oc*-enabled 180 Ringing-response defining a loss-rate of 20% over a timespan of 1000ms

Instead, this draft proposes in 5.10.1 that the upstream neighbor should prefer requests, which are for example, in-dialogue-requests or marked with *Resource-Priority*-flags as defined in RFC 4412[13]. This evaluation and decision-finding on each upstream host is producing additional load in a per-se overloaded infrastructure. We propose to use the new introduced `X-dropability`-header instead to keep this decision outside of the core infrastructure at the perimeter-security SBC-A.
The new introduced draft "Session Initiation Protocol (SIP) Rate Control"[7] extends [2] by the request-limitation algorithms, where we propose to include the `X-dropability`-header in one of three ways:

1. on loss-based algorithms: we propose to drop requests with higher `X-dropability`-values first until the wanted loss-rate is reached
2. on rate-based algorithms: like in loss-based algorithms, the requests with high `X-dropability`-values shall be dropped until the expected rate is reached
3. `X-dropability`-value–compliant oc-communication: as loss- and rate-based algorithms are extensive to be calculated (especially when dropping requests non-uniformly) we propose to add an algorithm to [7], where the overloaded host explicitly communicates a maximum allowed `X-dropability`-value. All requests with `X-dropability`-values greater than the maximum allowed value shall be dropped at the upstream neighbor. For archiving this task, we further propose to define a `X-dropability`-compliant oc-Via-parameter as in listing 1.5.

```
SIP/2.0 180 Ringing
Via: SIP/2.0/TLS p1.example.net;branch=z9hG4bK2d4790.1;
  oc=12;oc-algo="x-dropability";oc-validity=1000;
  oc-seq=1282321615.782
  ...
```

Listing 1.5. signaling a `X-dropability` compliant overload-control algorithm to the next upstream node. In this example the overloaded hosts signals to drop all requests with a `X-dropability`-value of larger or equal "12" over a timespan of 1000ms

5 Advantages of Pre-Evaluation and Summary

In case of overload in a core element, there is no way around blocking and dropping of requests to reduce the load to an acceptable amount. The preevaluation supports the decision process, which requests are better to be dropped, and which requests are needed to keep existing calls or registrations alive, resp. which are needed to fulfill the customers need of high QoS in the best possible way.

The presented approach is not completely aware of false-positive decisions, but compared to uniform dropping, the preevaluation keeps the number of lost important requests lower as requests marked with a high `X-dropability`-value are dropped earlier. High value requests (with a low dropability-value) are instead "protected" from being dropped at an early stage high-load-situation.

As [2] already presents basic ideas of request dependent request dropping, this draft does not bring aspects such as unsolicited communications or suspicious low-rate DoS-attacks into account.

Our approach stops flooding DoS-attacks and marks requests for possible UC and low-rate DoS as well as it qualifies SIP requests by their (economic- and QoS-) quality. It takes the *oc*-analyzing- and -deciding–load away from the (high-loaded) core components to a single, dedicated perimeter-security component, the SBC-A.

6 Outlook and Future Work

This proposed technique is the first conceptual step. In the next steps, we will use our testlab and simulator to compare uniform- and request dependent-dropping with our multistage approach. Here, we expect interesting chain-reactions and side-effects. For these simulations, we will use carrier grade user profiles provided by our project partners.

Our solution currently focuses primarily on request-qualification, but in further work, we will consider also response-flooding attacks. Based on Kamailio[2], the rate-limit-, a DNS-blacklist- and SpitAssassin[3]-module, we will implement the SBC-A as a reference implementation.

References

1. ETSI. NGN Congestion and Overload Control; Part 4: Overload and Congestion Control for H.248 MG/MGC. ES 283 039-4, Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), April 2007.
2. V. Gurbani, V. Hilt, and H. Schulzrinne. Session Initiation Protocol (SIP) Overload Control. Internet-Draft draft-ietf-soc-overload-control-07, Internet Engineering Task Force, January 2012. Work in progress.

[2] http://www.kamailio.org
[3] http://www.spitassassin.org

3. J. Hautakorpi, G. Camarillo, R. Penfield, A. Hawrylyshen, and M. Bhatia. Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments. RFC 5853, Internet Engineering Task Force, April 2010.
4. V. Hilt, E. Noel, C. Shen, and A. Abdelal. Design Considerations for Session Initiation Protocol (SIP) Overload Control. RFC 6357, Internet Engineering Task Force, August 2011.
5. M. Hirschbichler, C. Egger, O. Pasteka, and A. Berger. Using E-Mail SPAM DNS Blacklists for Qualifying the SPAM-over-Internet-Telephony Probability of a SIP Call. In *Digital Society, 2009. ICDS '09. Third International Conference on*, pages 254 –259, feb. 2009.
6. E. Noel and C.R. Johnson. Novel overload controls for sip networks. In *Teletraffic Congress, 2009. ITC 21 2009. 21st International*, pages 1 –8, sept. 2009.
7. E. Noel and P. PhilipWilliams. Session Initiation Protocol (SIP) Rate Control. Internet-Draft draft-noel-soc-overload-rate-control-02, Internet Engineering Task Force, December 2011. Work in progress.
8. Gaston Ormazabal, Sarvesh Nagpal, Eilon Yardeni, and Henning Schulzrinne. Secure sip: A scalable prevention mechanism for dos attacks on sip based voip systems. In Henning Schulzrinne, Radu State, and Saverio Niccolini, editors, *Principles, Systems and Applications of IP Telecommunications. Services and Security for Next Generation Networks*, volume 5310 of *Lecture Notes in Computer Science*, pages 107–132. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-89054-6_6.
9. Vincent Quinten, Remco van de Meent, and Aiko Pras. Analysis of techniques for protection against spam over internet telephony. In Aiko Pras and Marten van Sinderen, editors, *Dependable and Adaptable Networks and Services*, volume 4606 of *Lecture Notes in Computer Science*, pages 70–77. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-73530-4_9.
10. J. Rosenberg. Requirements for Management of Overload in the Session Initiation Protocol. RFC 5390, Internet Engineering Task Force, December 2008.
11. J. Rosenberg and C. Jennings. The Session Initiation Protocol (SIP) and Spam. RFC 5039, Internet Engineering Task Force, January 2008.
12. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, Internet Engineering Task Force, June 2002.
13. H. Schulzrinne and J. Polk. Communications Resource Priority for the Session Initiation Protocol (SIP). RFC 4412, Internet Engineering Task Force, February 2006.
14. H. Sengar, Haining Wang, D. Wijesekera, and S. Jajodia. Detecting voip floods using the hellinger distance. *Parallel and Distributed Systems, IEEE Transactions on*, 19(6):794 –805, june 2008.
15. Jin Tang and Yu Cheng. Quick detection of stealthy sip flooding attacks in voip networks. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1 –5, june 2011.
16. D. Wing, S. Niccolini, M. Stiemerling, and H. Tschofenig. Spam Score for SIP. Internet-Draft draft-wing-sipping-spam-score-02, Internet Engineering Task Force, February 2008. Work in progress.