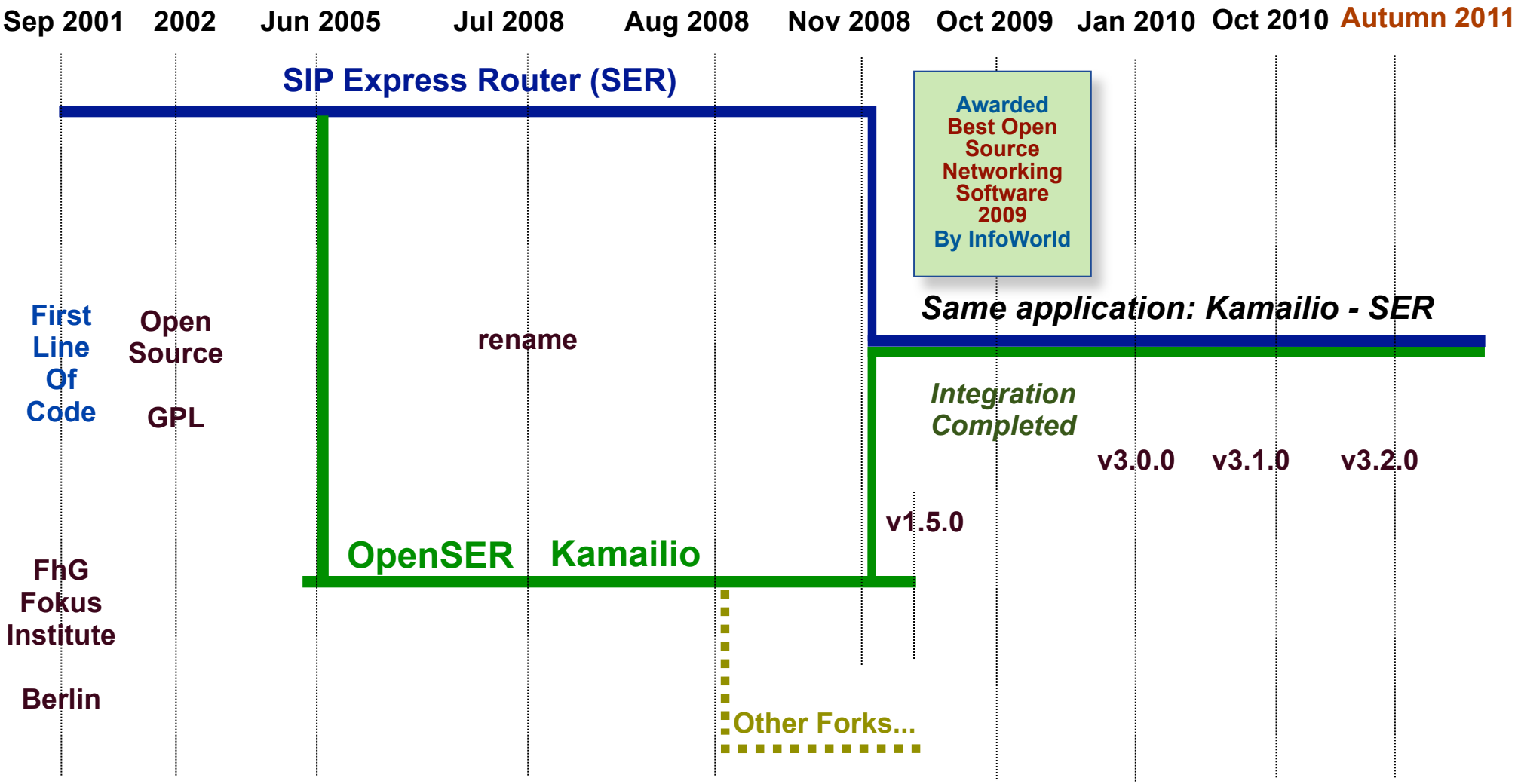
Strong Security
for Large VoIP Networks

ClueCon 2011, Chicago

Daniel-Constantin Mierla
Co-Founder Kamailio SIP Server
http://www.asipto.com

A bit of history - 10 years SER

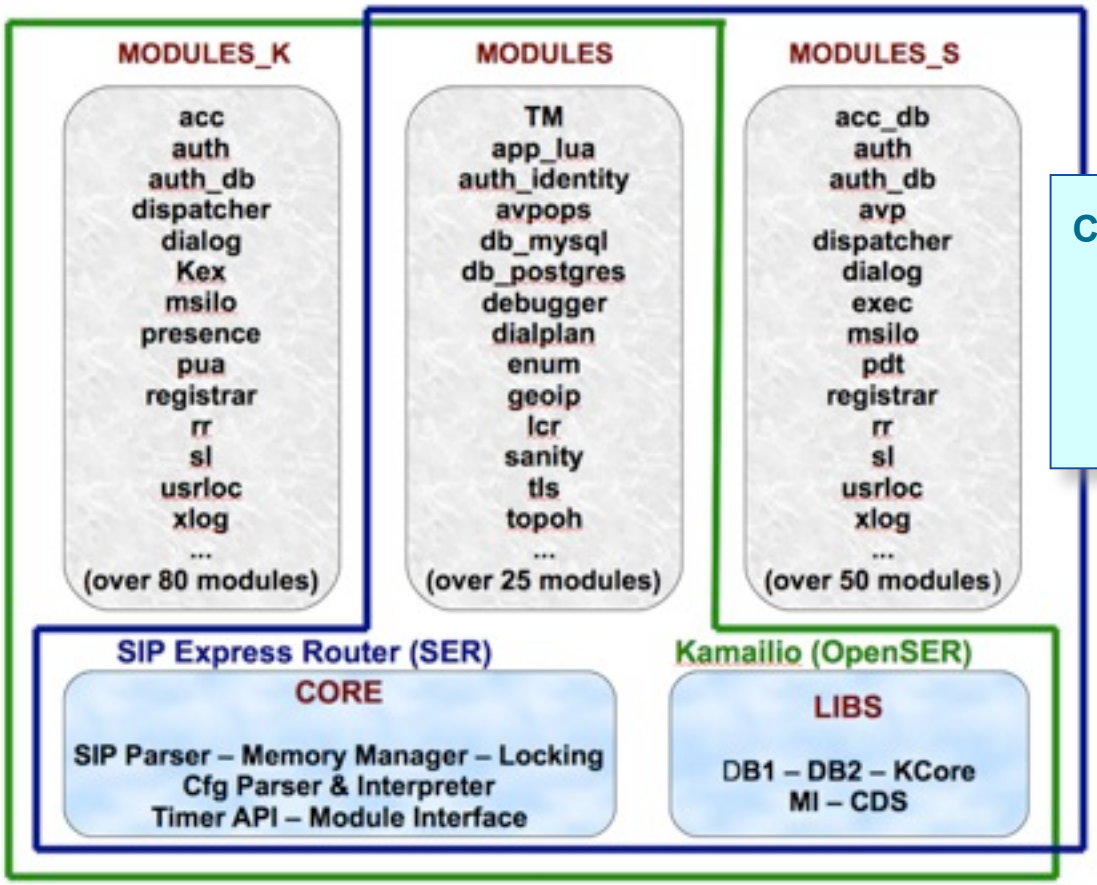


http://www.kamailio.org
http://sip-router.org

I am doing a usage survey (voluntary - confidential - anonymous)

- *minutes or calls per month, active subscribers...*
- *so far about 15 reports, resulting in over 3 000 000 000 minutes/month*

3.x Releases: One application, Two names



Constrained mainly by database schema

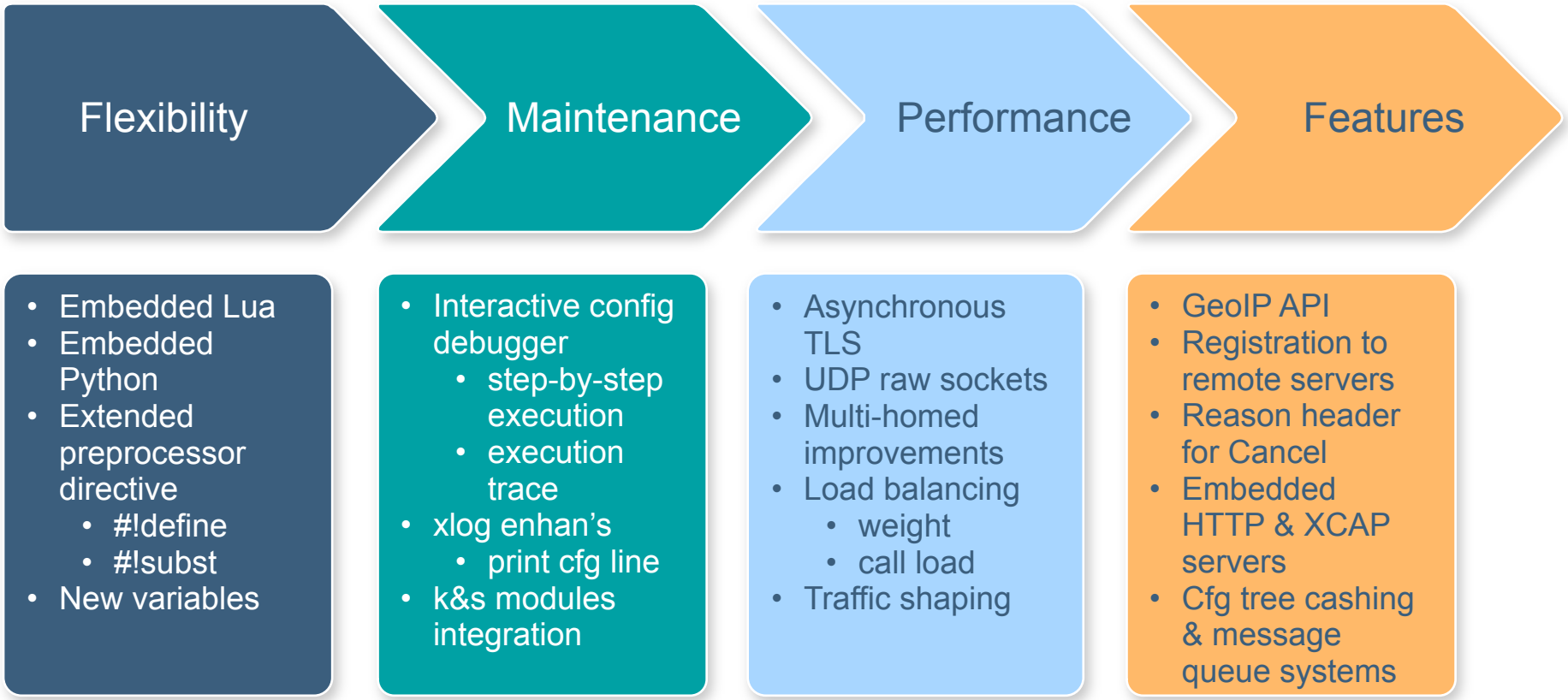
- during 2005 - 2008, SER and Kamailio developed different database structure to store user profiles and routing data
- strong dependency on administration and auto-provisioning systems

Example

- subscriber: username, password, DID, ACL, a.s.o.
- *Kamailio - table with many columns (one attribute in a dedicated column)*
- *SER: table with many rows of (attribute name, attribute value)*

Many duplicated modules were merged meanwhile

New in 3.1.0



http://www.kamailio.org/w/kamailio-openser-v3.1.0-release-notes/

State of the project

- Internal architecture refactored for v3.0.0
  - support asynchronous processing
    - TCP and TLS
    - SIP request handling
  - transaction management
  - internal libraries

Right now

- very stable core and main components
- ➡ toped with our well known scalability and flexibility
- safe framework for future development
- ➡ your work (extensions and deployments) is safe from now on for many years - there is no need to change the architecture again
- focus is on new features
- ➡ 3.2.0 (and the next slides) shows that

Scalability (info from public domain)

- services with millions of active subscribers
- ➡ 1&1 Germany (> 3M)
- services routing billions of call minutes per month
- ➡ might be the guy next to you (or pay attention tomorrow)

New in 3.2.0

Many native extensions to Lua

cfg routing logic all in Lua

Distributed Message Queue

Using SIP and Peer-to-Peer

SQLite connector

use file based database for embedded systems

Partitioned user location service

many nodes sharing location data

Redis No-SQL

connector from config

New in 3.2.0 – presence server

Reg-Info Implementation

RFC3860
pub-sub service for
location data

RLS
OMA specs
split NOTIFY bodies
XPath support within doc

Embedded XCAP server

OMA - specs
If-Match cond

Presence Server

data distribution across
many instances through
database

Presence User Agent

updates for latest
RL services

New in 3.2.0

async module

run asynchronously parts
of config file
(route blocks)

ipops module

a set of operations for
handling IPv4/IPv6 addresses

sdpops module

SDP body
management

New features in old parts

acc - write full CDR at once
dialog - attach extra attributes
core - more pre-processor directives
pv - new variables and transformations
tmx - export of async TM functions
sqlops - support for xavps
uac - enhancements to remote registration
siptrace - traffic replication enhancements
.....

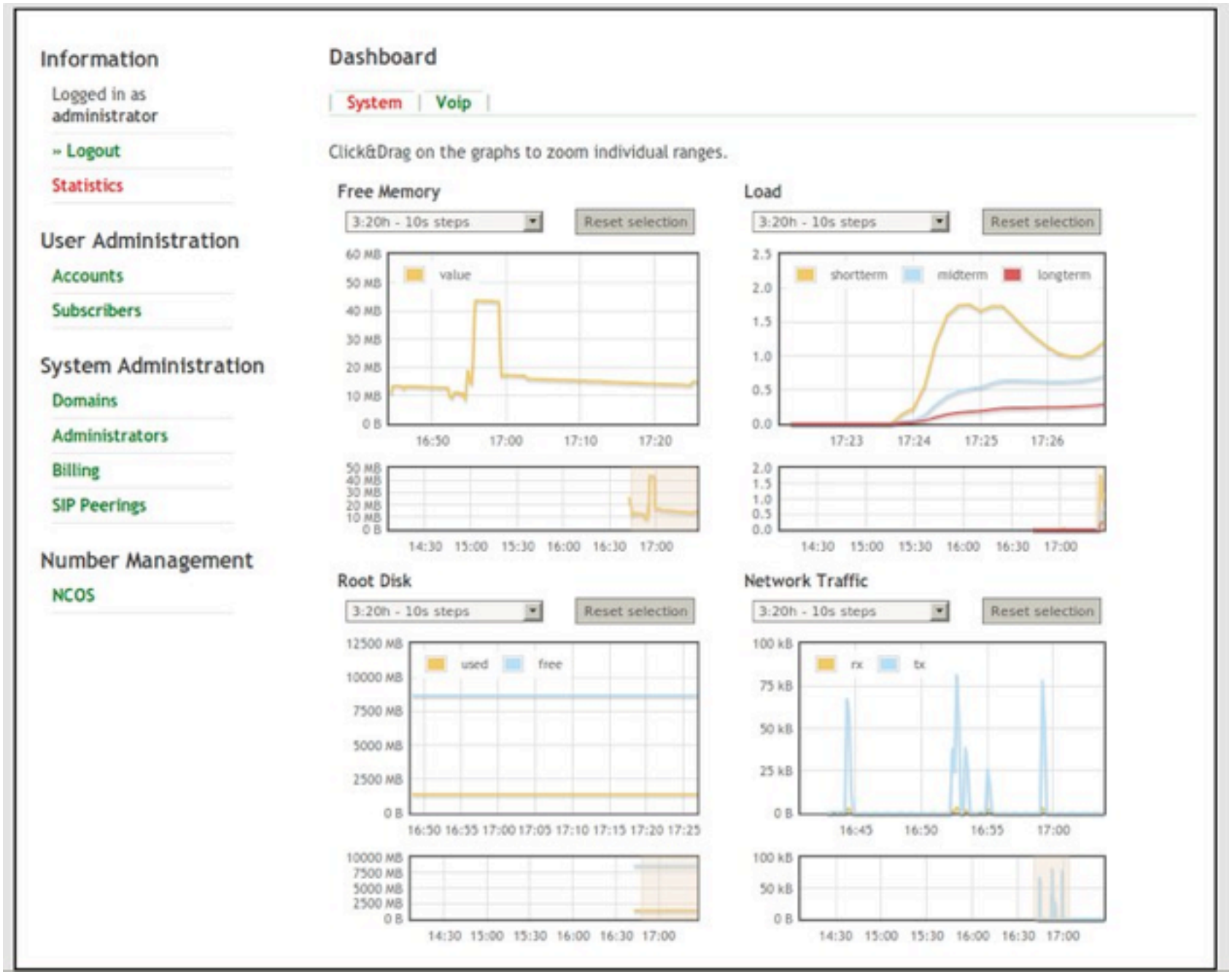
IMS Extensions
about 10 new modules
(P-CSCF, I-CSCF, S-CSCF...)

SIP:Provider CE

SIP:Provider - http://www.sipwise.com/products/spce/

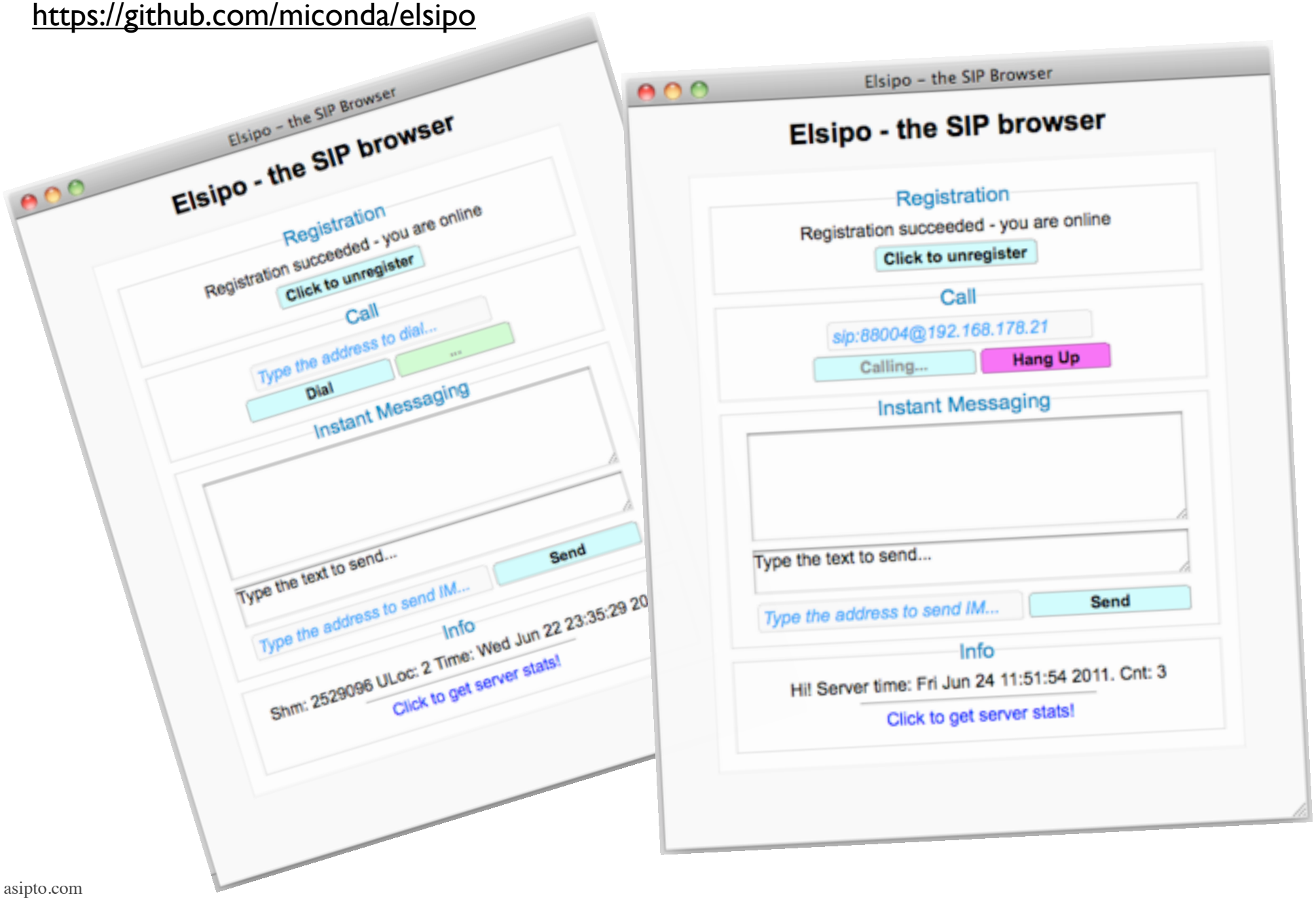
* complete VoIP servicing platform using Kamailio for SIP routing
* administration interface and user portal
* ready to roll-out open source Community Edition
* easy to install with DEB packages - images for VMWare and VirtualBox

SIP:Provider CE



Elsipo – this SIP browser

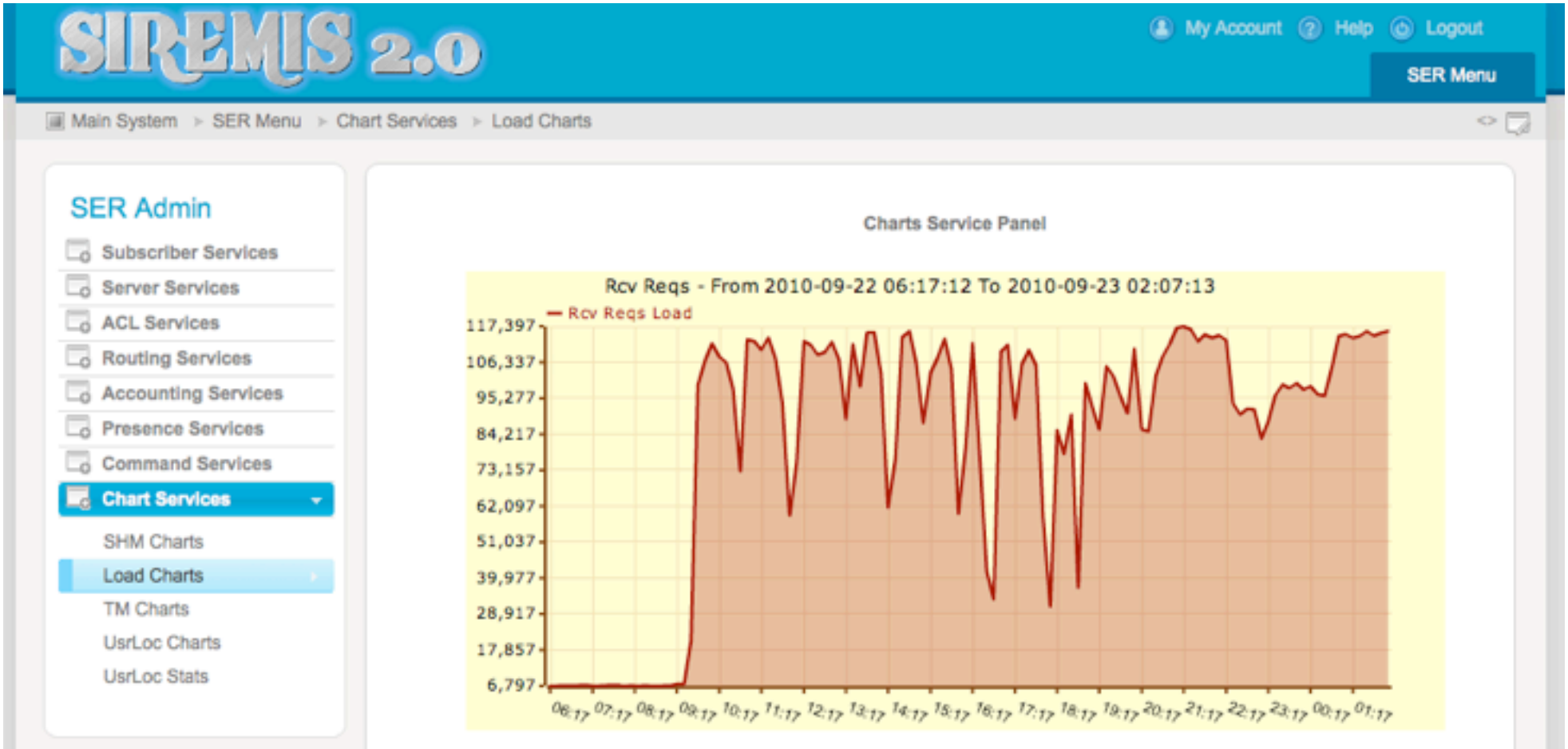
https://github.com/miconda/elsipo



Homer Project

Siremis 2.0

http://siremis.asipto.com/



Security For Large VoIP Networks

TLS – Encryption of communication

- now as simple as loading a module - **tls**
  - no more headaches like in 1.x - no need to recompile everything
  - very scalable
  - asynchronous TLS sending
  - can be configured via module parameter or dedicated config file

```
...
loadmodule "tls.so"

modparam("tls", "private_key", "/etc/kamailio/kamailio-selfsigned.key")
modparam("tls", "certificate", "/etc/kamailio/kamailio-selfsigned.pem")
modparam("tls", "ca_list", "/etc/kamailio/calist.pem")
```

TLS - via Config File

- Config by .ini-like file
  - dedicated file which can contain tls attributes
  - can include config for more than one server
  - can include config specific for clients

```
...
modparam("tls", "config", "/etc/kamailio/tls.cfg")
...
```

```
[server:default]
method = TLSv1
verify_certificate = no
require_certificate = no
private_key = default_key.pem
certificate = default_cert.pem
ca_list = default_ca.pem
```

TLS – Scalability

- a research project about Green VoIP
  - by Columbia University, NY
  - using complete config file, with user authentication and NAT traversal
  - injected traffic captured from an European ITSP

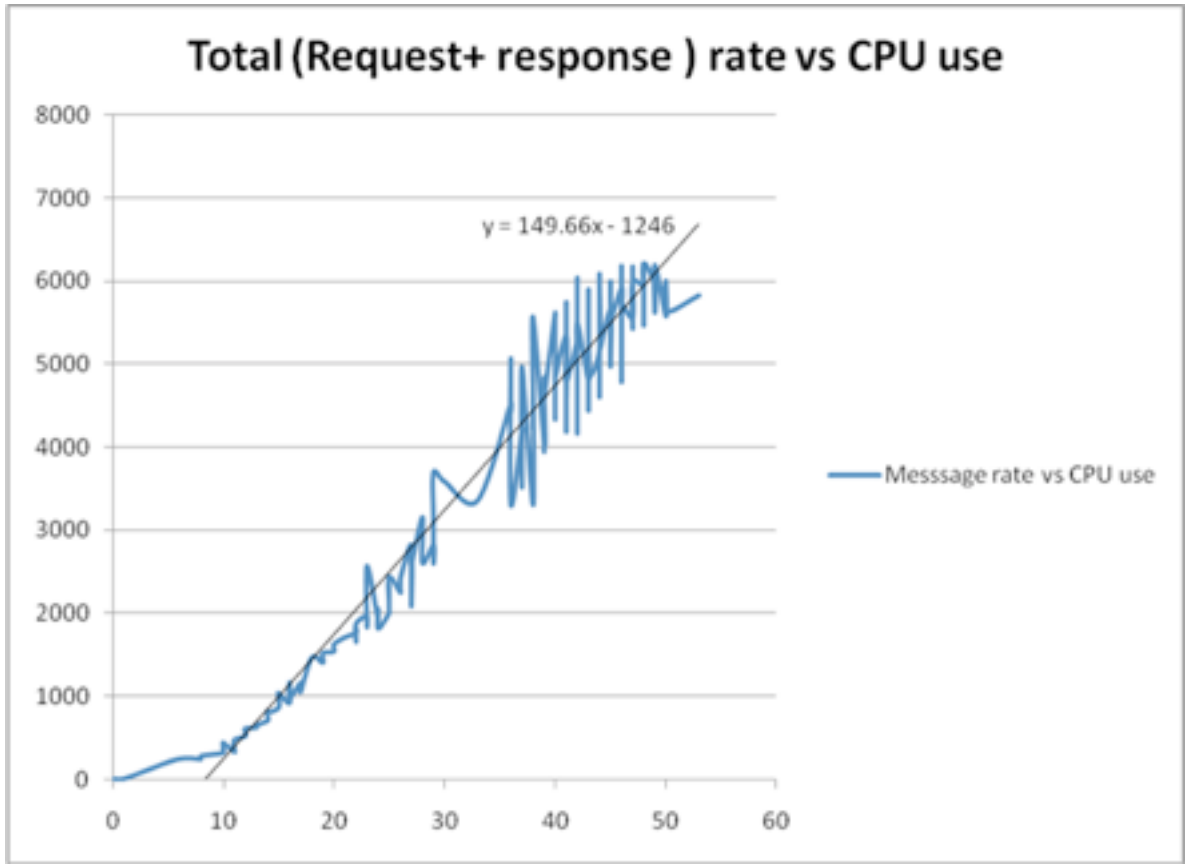
http://www.kamailio.org/w/2011/05/green-voip-energy-efficiency-and-performaces-of-v3-0/

Some interesting results:

- one instance of SIP server with *500 000 online users* (mixed users – behind and not NAT routers) – consumed energy *210W*
- one instance of SIP server with *1 000 000 online users* (no NAT involved) – consumed energy *190W*
- on a 32-bit machine with 4GB of memory and with 2.5GB reserved for SIP server, the server could support *43 000 simultaneous TLS connections* – consumed energy *209W*
- one SIP server instance with *80 000 permanent TCP connections*, the SIP server could still handle at least *1000 requests per second* and a connection arrival rate of *1000 new connections per second*, done for 20 000 new connections. CPU load generated by the SIP server was from *6% to 8%*.

TLS – Stress tests

- private company lab environment
  - Kamailio 3.1.x with 8 children and 2 GB memory
- traffic stress
  - 6000 SIP messages/second for 2 weeks
- socket stress
  - created over 4000 connections
  - released the connections immediately
  - at the same time created more connections



Topology hiding

- **TOPOH** module

- secret key to encode/decode
- encoded fields are SIP grammar valid
- encoding IP and prefixes can be set via parameters
- survive restarts
- no functions to be called in config file
  - everything is done automatically
  - hooks in core after receiving and before sending
  - just load the module and adjust parameters

- *penalty on 2000 call setups / second is not noticeable*

- *use it with a media relay to hide the source of media traffic*

Topology hiding

```
...
loadmodule "topoh.so"
...
# ----- topoh params -----
modparam("topoh", "mask_key", "my secret here")
modparam("topoh", "mask_ip", "10.1.1.10")
...
```

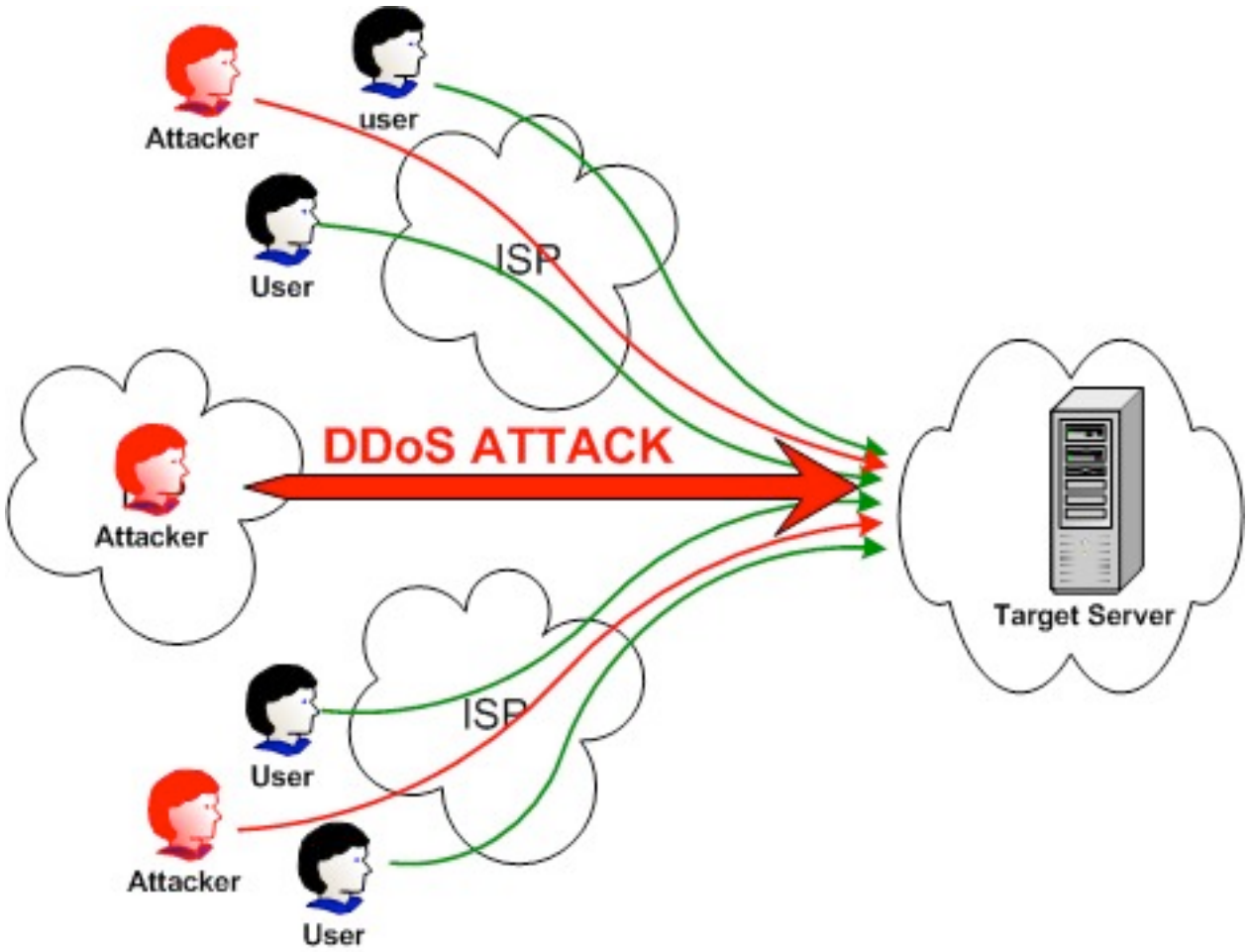
Topology hiding – INVITE in

U 2011/02/18 20:09:05.622472 192.168.178.27:40416 -> 192.168.178.26:5060
INVITE sip:101@192.168.178.26 SIP/2.0.
Via: SIP/2.0/UDP 192.168.178.27:40416;branch=z9hG4bK321149767.
From: "105" <sip:105@192.168.178.26>;tag=166646806.
To: <sip:101@192.168.178.26>.
Call-ID: 989804978-40416-6@BJC.BGI.BHI.CH.
CSeq: 50 INVITE.
Contact: "105" <sip:105@192.168.178.27:40416>.
Max-Forwards: 70.
User-Agent: Grandstream GXV3140 1.0.7.3.
Privacy: none.
P-Preferred-Identity: "105" <sip:105@192.168.178.26>.
Supported: replaces, path, timer.
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO, REFER, UPDATE, MESSAGE.
Content-Type: application/sdp.
Accept: application/sdp, application/dtmf-relay.
Content-Length: 483.
.

Topology hiding – INVITE out

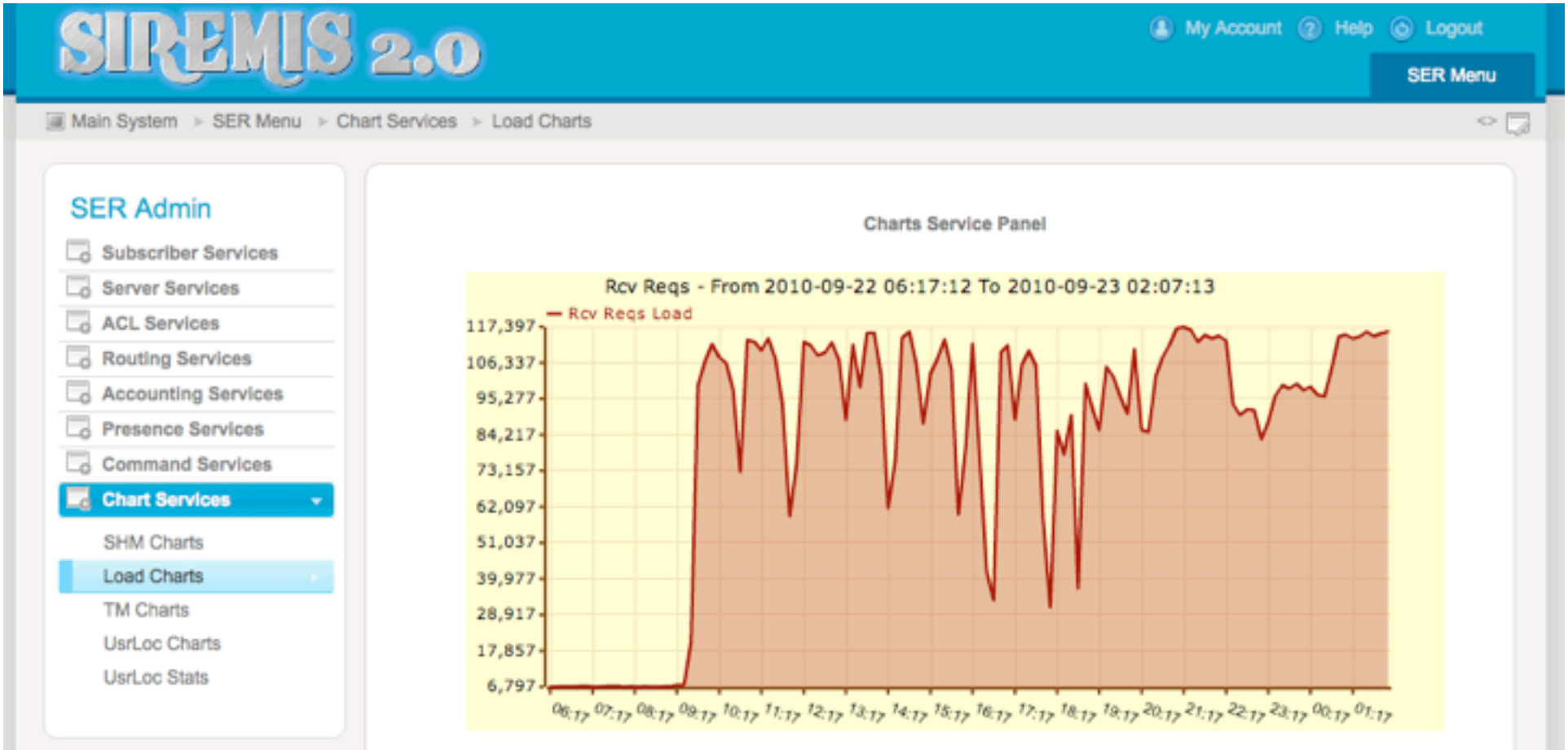

```
U 2011/02/18 20:09:05.628883 192.168.178.26:5060 -> 192.168.178.22:1056
INVITE sip:101@192.168.178.22:1056;line=mu3z2i1j SIP/2.0.
Record-Route: <sip:192.168.178.26;lr=on>.
Via: SIP/2.0/UDP 192.168.178.26;branch=z9hG4bK8d21.062561f6.0.
Via: SIP/2.0/UDP 10.1.1.10;branch=z9hG4bKsr-
JfymiMenCtp4urS5CX1ZiHvRItc.TM5nCHOBT6SfCXN94v5pswyRIRDZN80HU6gBI8LqTwDiCMe.CXm0TMNP
.
From: "105" <sip:105@192.168.178.26>;tag=166646806.
To: <sip:101@192.168.178.26>.
Call-ID: 989804978-40416-6@BJC.BGI.BHI.CH.
CSeq: 50 INVITE.
Contact: "105" <sip:10.1.1.10;line=sr-ORylIHvlTJS.IXenCXNciHvPItcZTMWfC6m.T5**>.
Max-Forwards: 69.
User-Agent: Grandstream GXV3140 1.0.7.3.
Privacy: none.
P-Preferred-Identity: "105" <sip:105@192.168.178.26>.
Supported: replaces, path, timer.
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO, REFER, UPDATE, MESSAGE.
Content-Type: application/sdp.
Accept: application/sdp, application/dtmf-relay.
Content-Length:   483.
.
```

DoS and DDoS attacks



DoS and DDoS attacks

- in a day by day service monitoring ...



Useful modules

- **HTABLE** module
  - generic cache system

- track failed authentication
- forbid new attempts if a threshold is reached in a certain period of time
- send alerts to admin, etc.

- example with registrations
  - prevent discovery of user passwords
  - detect mistyped passwords

Useful modules

- **RATELIMIT** module
  - definition of generic pipes and queues
    - types of SIP requests associated with queues
    - queues associated with pipes
  - similar to BSD ipfw
  - various algorithms to drop traffic
- does not take in consideration source IP address
  - can be used for DDoS alerts as well
- no internal actions for blocking
  - reports when there is an higher traffic than the limit on pipe
    - is the administrator decision in the config file
      - drop silently
      - send stateless reply

Useful modules

- new **PIPELIMIT** module - since 3.1.0
  - like RATELIMIT, but ...
    - pipe definitions in database
    - dynamic names for pipes
    - no-limit for number of pipes
    - re-load at runtime
    - no embedded queues definition, config language gives better tools to define them with conditions

- more modules to look at:
  - pike
  - dialog
  - sqlops
  - memcached, ndb_redis ...

Scanning - Brute Force Attacks

http://kb.asipto.com/kamailio:usage:k31-sip-scanning-attack

- block user for 15 minutes if it fails to authenticate 3 times in a row

```
...
loadmodule "htable.so"
...
modparam("htable", "htable", "a=>size=8;autoexpire=920;")
...
route {
...
  if(is_present_hf("Authorization"))
  {
    if($sht(a=>$au::auth_count)==3)
    {
        $var(exp) = $Ts - 900;
        if($sht(a=>$au::last_auth) > $var(exp))
        {
            sl_send_reply("403", "Try later");
            exit;
        } else {
            $sht(a=>$au::auth_count) = 0;
        }
    }
```

Scanning - Brute Force Attacks

```
    if(!www_authenticate("$td", "subscriber"))
    {
        switch ($retcode) {
            case -1:
                sl_send_reply("403", "Forbidden");
            exit;
            case -2:
                if($sht(a=>$au::auth_count) == null)
                    $sht(a=>$au::auth_count) = 0;
                $sht(a=>$au::auth_count) = $sht(a=>$au::auth_count) + 1;
                if($sht(a=>$au::auth_count) == 3)
                    xlog("auth failed 3rd time - src ip: $si\n");
                $sht(a=>$au::last_auth) = $Ts;
            break;
        }
        www_challenge("$td"/*realm*/,"0"/*qop*/);
        exit;
    }
    $sht(a=>$au::auth_count) = 0;
  } else {
    www_challenge("$td","0");
    exit;
  }
...
}
...
```

Flooding – block SIP attacks in config

http://kb.asipto.com/kamailio:usage:k31-sip-scanning-attack

- block traffic from specific IP address for 5 minutes if it exceeded a threshold

```
loadmodule "htable.so"
...
modparam("htable", "htable", "ipban=>size=8;autoexpire=300;")

...

route {
  if($sht(ipban=>$si)!=$null)
  {
     # ip is already blocked - keep the node warm
     pike_check_req();
     xdbg("request from blocked IP - $rm from $fu (IP:$si:$sp)\n");
     exit;
  }
  if (!pike_check_req()) {
     $sht(ipban=>$si) = 1;
     xlog("L_ALERT","ALERT: pike block $rm from $fu (IP:$si:$sp)\n");
     exit;
  }
...
}
```

```
kamctl fifo sht_dump ipban
```

Fail2ban – blocking in the firewall

http://kb.asipto.com/kamailio:usage:k31-sip-scanning-attack

- firewall traffic from specific IP address if it send traffic that fails to authenticate 3 times in a row

Create **/etc/fail2ban/filter.d/kamailio.conf** with following content:

```
[Definition]
# filter for kamailio messages
failregex = Blocking traffic from <HOST>
```

Edit **/etc/fail2ban/jail.conf** and add:

```
findtime  = 600

[kamailio-iptables]
enabled  = true
filter   = kamailio
action   = iptables-allports[name=KAMAILIO, protocol=all]
logpath  = /var/log/kamailio.log # update it with your kamailio log path
maxretry = 10
bantime  = 1800
```

In Kamailio configuration, use next line whenever you want to ban an IP for half an hour:

```
xlog("Blocking traffic from $si\n");
```

Fail2ban - blocking in the firewall

Note: $si is a config file variable that expands at runtime to source IP address.

```
... Blocking traffic from 1.2.3.4
```

<<< message in syslog

For example, plugging it in the above Kamailio snippets:

```
...
        $var(exp) = $Ts - 900;
        if($sht(a=>$au::last_auth) > $var(exp))
        {
            sl_send_reply("403", "Try later");
            xlog("Blocking traffic from $si\n");
            exit;
        } else {
            $sht(a=>$au::auth_count) = 0;
        }
...
```

Thank you!
Questions?

Daniel-Constantin Mierla
Co-Founder Kamailio
http://www.asipto.com
Twitter: @miconda

BERLIN, GERMANY
SEP 2, 2011
10 years SER Conference
SIP Express Router & Kamailio