How sipgate Uses Debian And Other Open Source Software

MiniDebConf Hamburg 2019

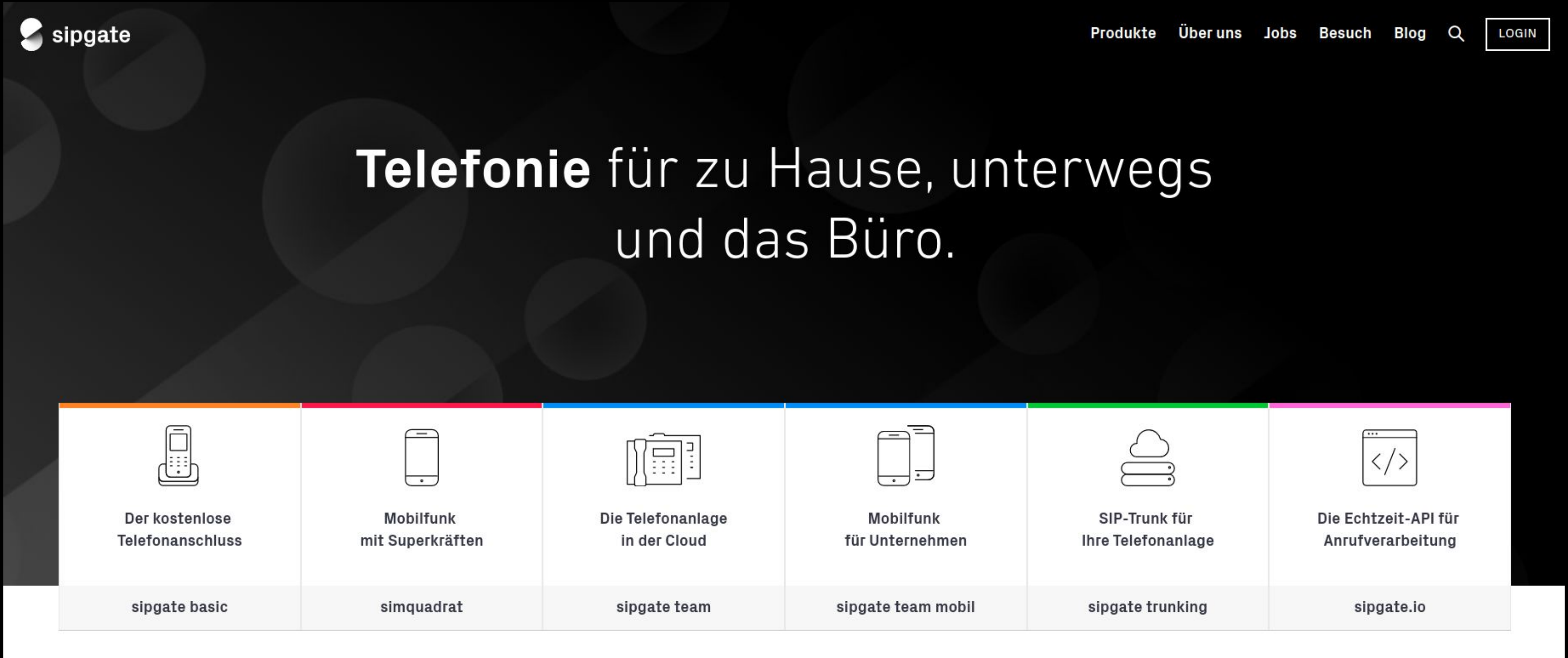
Rudolph Bott
@rbo_ne

Cenk Kücük
@_ifthenelse_

sipgate

sipgate?

sipgate



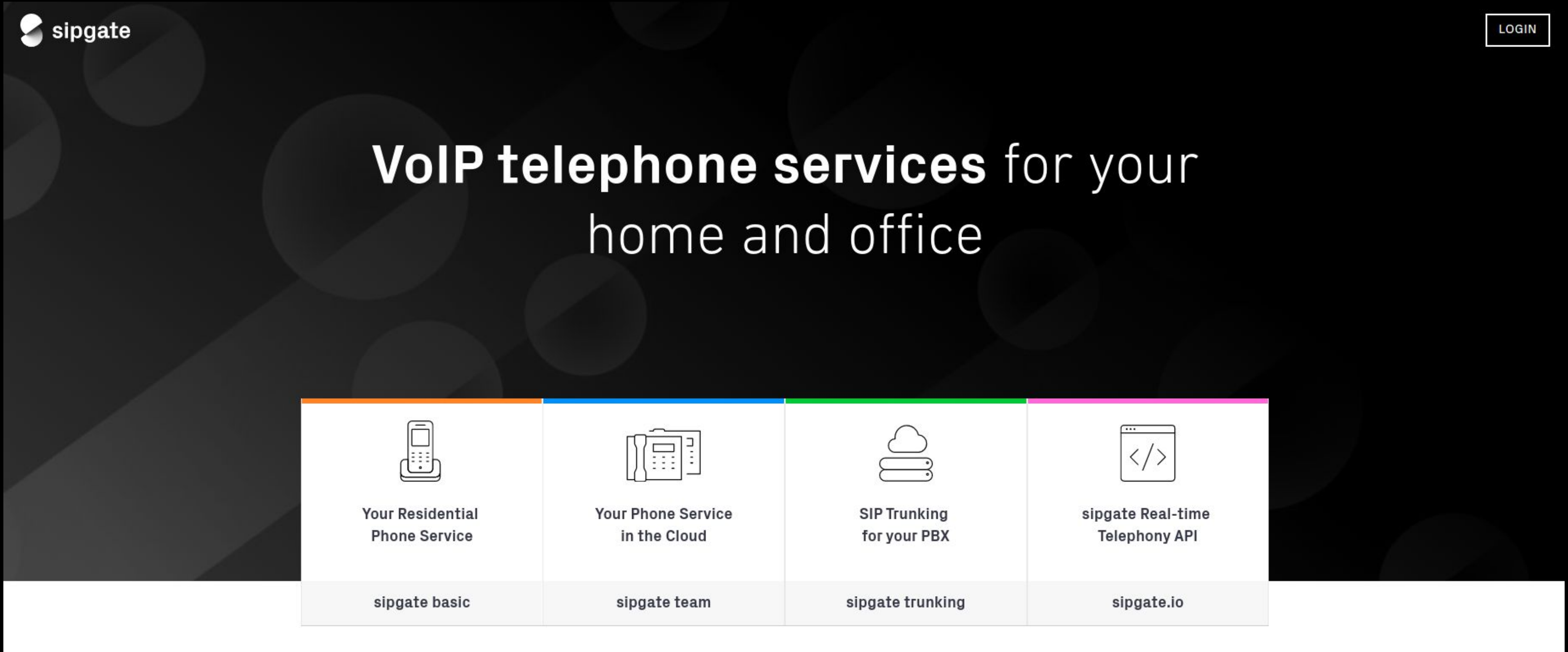
Telefonie für zu Hause, unterwegs und das Büro.

Der kostenlose Telefonanschluss	Mobilfunk mit Superkräften	Die Telefonanlage in der Cloud	Mobilfunk für Unternehmen	SIP-Trunk für Ihre Telefonanlage	Die Echtzeit-API für Anrufverarbeitung
sipgate basic	simquadrat	sipgate team	sipgate team mobil	sipgate trunking	sipgate.io

satellite

CLINQ

sipgate



VoIP telephone services for your home and office

Your Residential Phone Service

sipgate basic

Your Phone Service in the Cloud

sipgate team

SIP Trunking for your PBX

sipgate trunking

sipgate Real-time Telephony API

sipgate.io

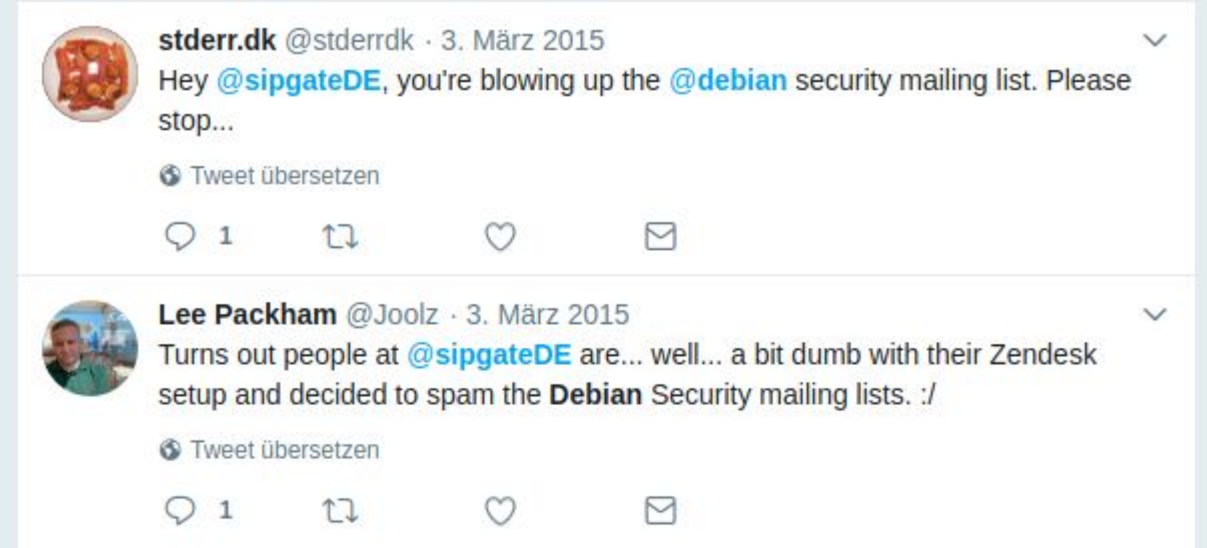
sipgate

Technical Facts

- ~700 servers
  - ~250 are bare metal (mostly telephony-related, databases, virtualisation, infrastructure)
- all servers are Debian based
  - except those few nasty office-related services
- client systems: Apple with macOS or Lenovo Thinkpads with Ubuntu
- Windows (test) environments provided via AWS Workspaces

Technical Facts

- ~700 servers
  - ~250 structure)
- all serve
  - exce
- client sy ntu
- Windows



stderr.dk @stderrdk · 3. März 2015

Hey @sipgateDE, you're blowing up the @debian security mailing list. Please stop...

Lee Packham @Joolz · 3. März 2015

Turns out people at @sipgateDE are... well... a bit dumb with their Zendesk setup and decided to spam the **Debian** Security mailing lists. :/

DebConf15
Heidelberg, Germany

DEBCONF17
MONTRÉAL · CANADA

KAMAILIO WORLD
CONFERENCE & EXHIBITION

MiniDebConf
Hamburg
2018

hiera nginx haproxy postfix pacemaker squid gopass qemu GCC openjdk WildFly composer consul ansible git quagga vert.x strongswan ImageMagick yate cfssl maven jenkins-debian-glue gradle ulogd nodejs bacula kafka iptables ganeti luks homer openvpn logstash foreman kamailio material-ui make beaver drbd ucarp lighttpd syslog-ng terraform jenkins MySQL kibana OpenNTPD npm openLDAP exim Akka fail2ban puppet asterisk traefik OpenSSH Galera rtpengine swagger MongoDB Grafana JUnit freeradius gnupg icinga Metabase react vagrant docker keycloak bird cyrus telegraf InfluxDB maxwell apache Wordpress Mockito elasticsearch kerberos ntopng yarn redis

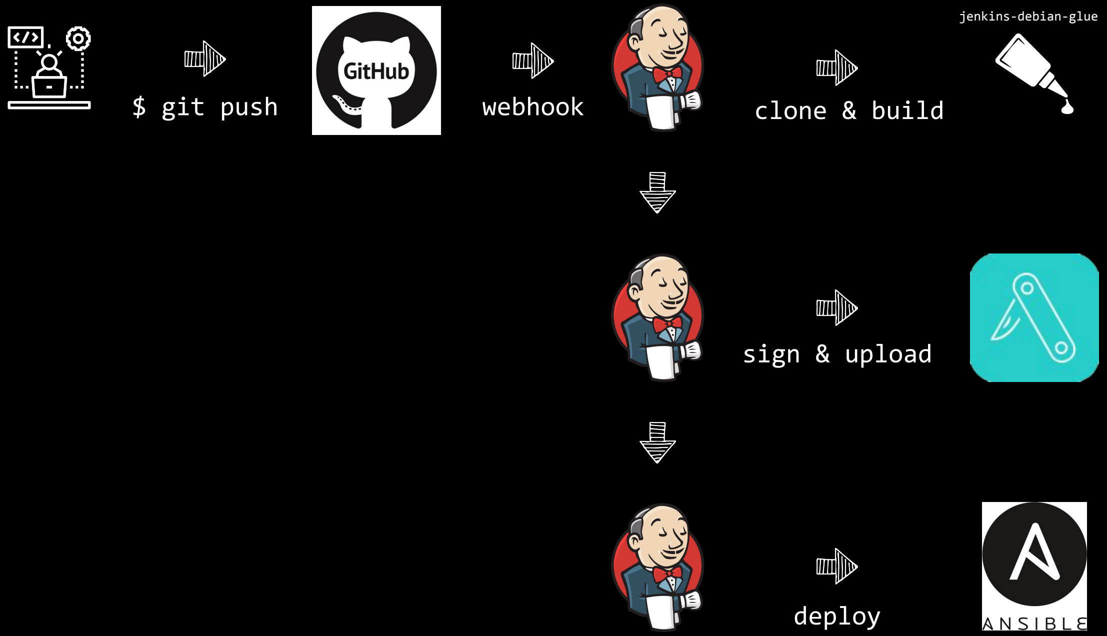
hiera nginx haproxy postfix pacemaker squid gopass qemu
GCC openjdk WildFly composer consul ansible git quagga vert.x
strongswan ImageMagick yate cfssl maven jenkins-debian-glue
gradle ulogd nodejs bacula kafka iptables ganeti luks homer
openvpn logstash foreman kamailio material-ui make beaver
drbd ucarp lighttpd syslog-ng terraform jenkins MySQL kibana
OpenNTPD npm openLDAP exim Akka fail2ban puppet asterisk
traefik OpenSSH Galera rtpengine swagger MongoDB Grafana
JUnit freeradius gnupg icinga Metabase react vagrant docker
keycloak bird cyrus telegraf InfluxDB maxwell apache
Wordpress Mockito elasticsearch kerberos ntopng yarn redis

sipgate

Next up

- Debian Packaging (or: How we deploy our applications)
- Unattended Upgrades (or: How we keep our systems up-to-date)
- cfssl (or: How to run your own CA)
- ganeti (or: How we play the virtualisation game)
- ELK Stack (or: How we help our colleagues access logs)

How we deploy our applications



$ git push

webhook

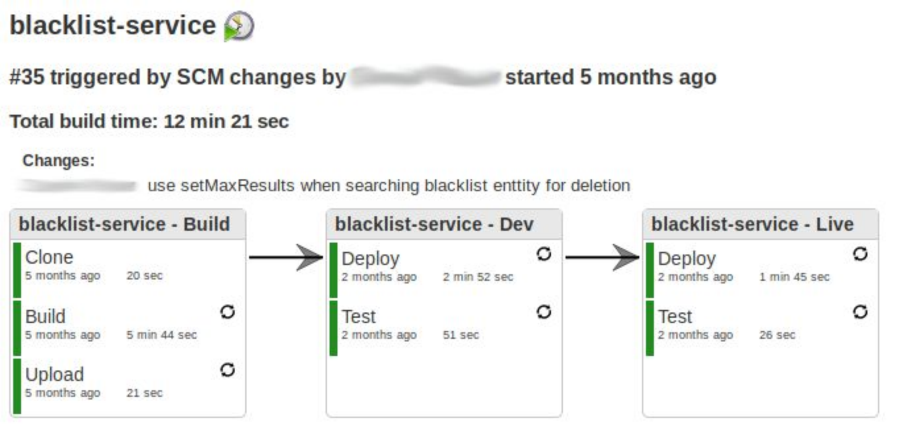
jenkins-debian-glue

clone & build

sign & upload

deploy

How we deploy our applications



blacklist-service

#35 triggered by SCM changes by [redacted] started 5 months ago

Total build time: 12 min 21 sec

Changes:

[redacted] use setMaxResults when searching blacklist enttity for deletion

blacklist-service - Build → **blacklist-service - Dev** → **blacklist-service - Live**

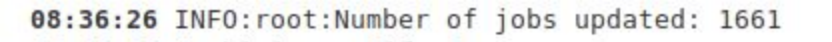
blacklist-service - Build
- Clone 5 months ago 20 sec
- Build 5 months ago 5 min 44 sec
- Upload 5 months ago 21 sec

blacklist-service - Dev
- Deploy 2 months ago 2 min 52 sec
- Test 2 months ago 51 sec

blacklist-service - Live
- Deploy 2 months ago 1 min 45 sec
- Test 2 months ago 26 sec

How we deploy our applications

Use Jenkins Job Builder:
https://opendev.org/jjb/jenkins-job-builder/



```
08:36:26 INFO:root:Number of jobs updated: 1661
08:36
08:36
```

How we deploy our applications

- first reprepro, later switched to aptly
  - rollback after failed deployments required multiple versions of each package
- numbers of unique packages per Debian version:
  - wheezy: 206 (1026)
  - jessie: 322 (1599)
  - stretch: 187 (728)
- sipgate’s own applications, but also third-party software because
  - it is not available in Debian and no repository maintained by the author exists
  - the version available in Debian (-backports) is too old
  - we require extra patches/build flags

sipgate

How we deploy our applications

- minimum Ansible version 2.7
- Ansible/Deployment Repository: ~13k commits from ~70 contributors
- 285 playbooks
- 275 roles
- 25 custom modules
- used for service deployment, server setup, local dev setups, maintenance
- Server backups? No Sir! (well, it depends)

Unattended Upgrades - why

- stay up to date
- security threats
- performance improvements
- smaller changes are better than bigger changes
- eliminate toil
  - doing repetitive task manually → mistakes

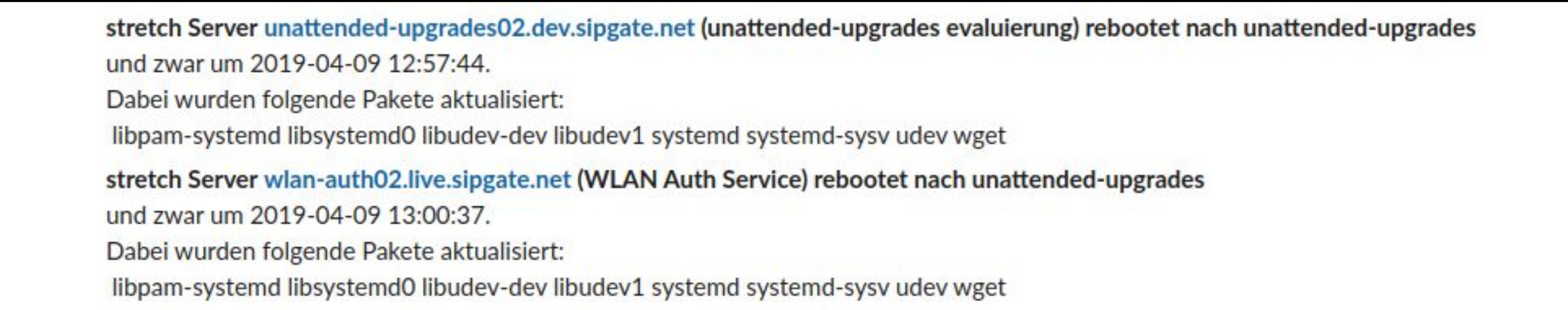
Unattended Upgrades - how

- cronjob
  - day/ night
  - day of week
- only 1 host of a host group per day
- wrapper around unattended-upgrades

Unattended Upgrades - how

- Pre-upgrade tasks
    - check for locks
    - set lock
    - set downtime in the monitoring
    - service specific maintenance tasks
        - deregister from the load balancer
        - do not accept new phone calls and wait them to finish

Unattended Upgrades - how



stretch Server unattended-upgrades02.dev.sipgate.net **(unattended-upgrades evaluierung) rebootet nach unattended-upgrades**
und zwar um 2019-04-09 12:57:44.
Dabei wurden folgende Pakete aktualisiert:
libpam-systemd libsystemd0 libudev-dev libudev1 systemd systemd-sysv udev wget

stretch Server wlan-auth02.live.sipgate.net **(WLAN Auth Service) rebootet nach unattended-upgrades**
und zwar um 2019-04-09 13:00:37.
Dabei wurden folgende Pakete aktualisiert:
libpam-systemd libsystemd0 libudev-dev libudev1 systemd systemd-sysv udev wget

Unattended Upgrades - how

- Post-reboot tasks
  - service specific maintenance tasks
    - register at the load balancer
    - start accepting new phone calls
  - remove lock

Unattended Upgrades

- 4670 reboots via unattended upgrades between October 2018 - May 2019
  - ~ 583 reboots/month
  - ~ 19 reboots/day
- global kill switch
- legacy software
- not every package creates /run/reboot-required
- updated libraries do not restart services

without any impact on our platform!

How to run your own CA

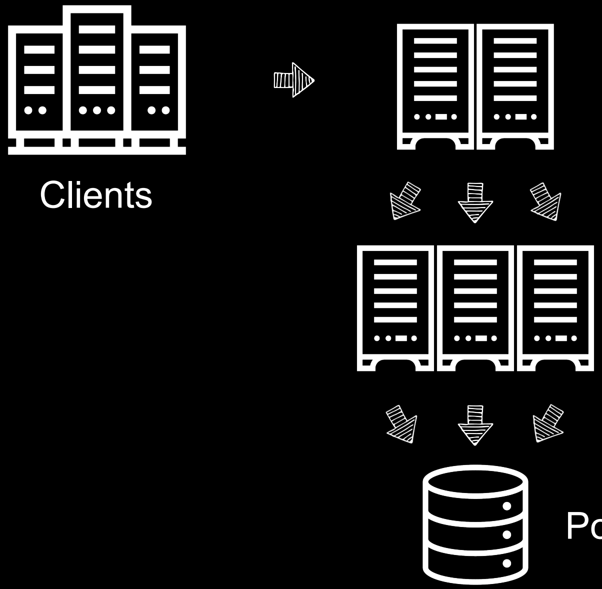
- what do we need certificates for?
  - internal or development websites / services
  - backup (bacula)
  - TLS syslog (syslog-ng)
  - databases (mysql)
  - ...you name it.
- so… you do know Let’s Encrypt is a thing, right?
  - yes, but we cannot use any of the available validation challenges
- but LE is open source, why not use that?
  - open source, but not ready for “home use”
  - essentially a wrapper around cfssl which adds registration/validation. Do we need that?
    - https://github.com/cloudflare/cfssl

How to run your own CA

- basic assumptions
    - our root CA lives offline and runs for a very long time (e.g. 100 years)
    - we use intermediate online CAs for signing, which are valid for one year
    - certificates are only valid for three months (or: until the intermediate CA expires)
    - the CA only issues certificates for a known set of domain names
    - certificates must be renewed automatically (along with service restart/reload)
    - the root CA will be auto-deployed to all relevant systems/keystores (linux servers, workstations, java keystores, macOs clients etc.)
    - all services need to ship the current intermediate CA along with the certificate

sipgate

How to run your own CA



Clients

traefik (generic internal service loadbalancer)
consul node

nginx (handling TLS & static file delivery)
cfssl API backend
consul node

PostgreSQL DB

How to run your own CA

- how does my $service receive a certificate + auto-renewal?

```
- name: Deploy $service certificate config
  include: service_certificate.yml
  vars:
     service_name: $service
     service_certificate_fqdn: "{{ inventory_hostname }}"
     service_certificate_restart_cmd: "service $service restart"
```

How to run your own CA

- lessons learned: there is no ‘standard way’ to provide certs/keys:
  - $service-$hostname-bundle-cert-key.pem
  - $service-$hostname-bundle-key-cert.pem
  - $service-$hostname-cert-bundle-key.pem
  - $service-$hostname-cert-key-bundle.pem
  - $service-$hostname-cert-key-dhparam.pem
  - $service-$hostname-key-bundle-cert.pem
  - $service-$hostname-key-cert-bundle.pem
  - $service-$hostname-key.pem
  - $service-$hostname-bundle-cert.pem
  - $service-$hostname-bundle.pem
  - $service-$hostname-cert-bundle.pem
  - $service-$hostname-cert.pem

How to run your own CA

- how does the renewal work?
  - a cronjob runs each day between 11 and 12am and checks all configured certificates
  - a certificate will be replaced 10 days ahead of expiry
  - after retrieving the new certificate, the restart/reload command will be triggered
- security aspects
  - certificates are only issued for whitelisted domains (built-in cfssl feature)
  - short-lived certificates are better than long-lived ones
  - “authentication” is only available on a network level - KISS and “good enough”
  - credentials or client certificates make things more complicated and would have to be stored on all servers and could leak

How to run your own CA

```
cfssl=> select count(*) from certificates;
 count
-------
 15057
(1 row)
```

How we play the virtualisation game

- Ganeti
  - cluster management tool for VMs
  - Xen, **KVM** or LXC
  - shared nothing cluster with job management
  - management: CLI or ganeti control center (webfrontend written in PHP)
    - https://github.com/sipgate/ganeti-control-center

How we play the virtualisation game

- three ganeti clusters with
    - 31 nodes
    - 427 instances
    - 608 physical CPU cores
    - 3.4 TB of memory
- DRBD replication to ensure instance availability
- node groups to bundle different hardware generations
- separate network for disk replication/live migration
- instance distribution: `hbal/hail` & exclusion tags

How we play the virtualisation game

- current state of ganeti: Google has mostly stopped active development

> “Ganeti 2.16.1 was released today (yes, on April 1). This is a bugfix and compatibility release, aiming mostly at making Ganeti usable out of the box on current Linux environments. This is also the first release done outside Google, who were kind enough to grant commit access to external contributors. “

- live migration - Java applications sometimes end up in a broken state

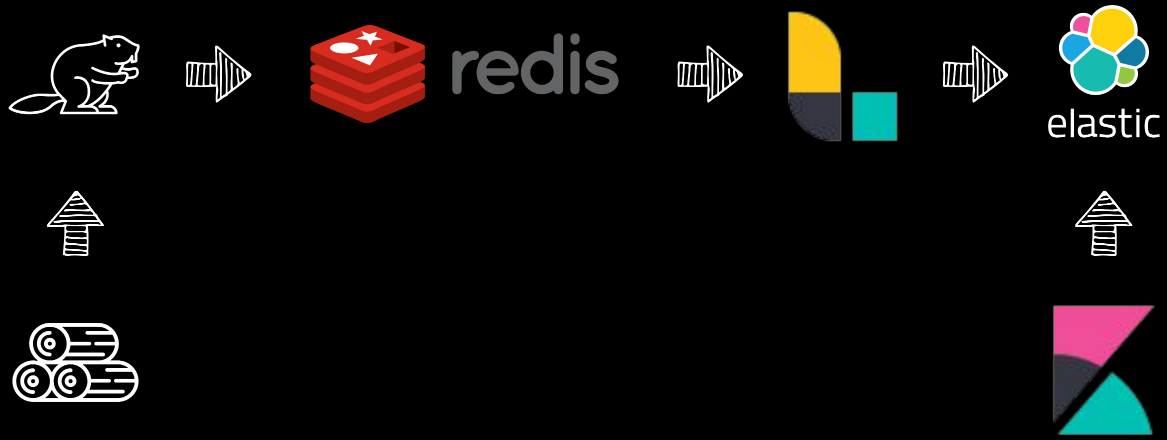
GanetiCon 2019
18th & 19th of June
Umeå/Sweden

https://ganeticon.org/

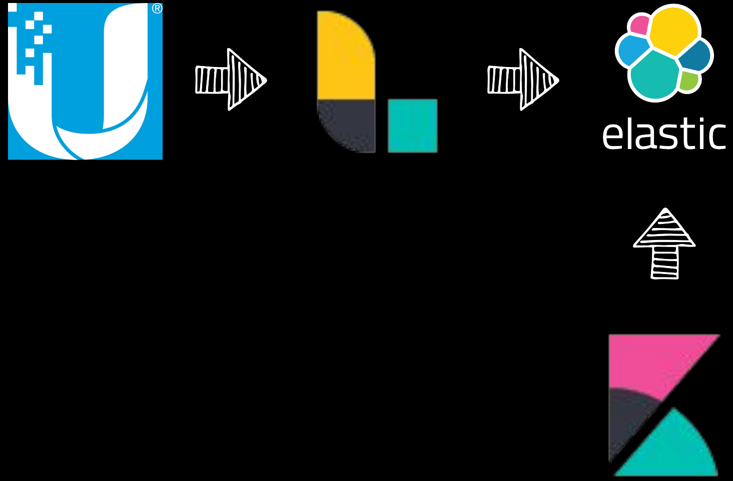
sipgate

ELK-Stack

- Elasticsearch, Logstash and Kibana
- actually, there is also redis and beaver (BLERK?)



ELK-Stack



ELK-Stack: lessons learned

- if you need an elasticsearch cluster, go for many smaller nodes
- if you allocate more than 32GB memory to elasticsearch, things will actually get worse, unless you have > 48 GB of memory
  - https://linux.m2osw.com/compressed-ordinary-object-pointers-java-64bit-jvm
- do not retain the original message after parsing it with logstash (grok et al) unless you really need it
- consider spawning simple automated “ELK-in-a-box” setups (e.g. with Ansible) VMs instead of larger clusters to store your data
  - easier upgrade of the stack (more servers/work but less risk to break a bunch of dashboards at the same time)

ELK: Everyday Dashboard Examples

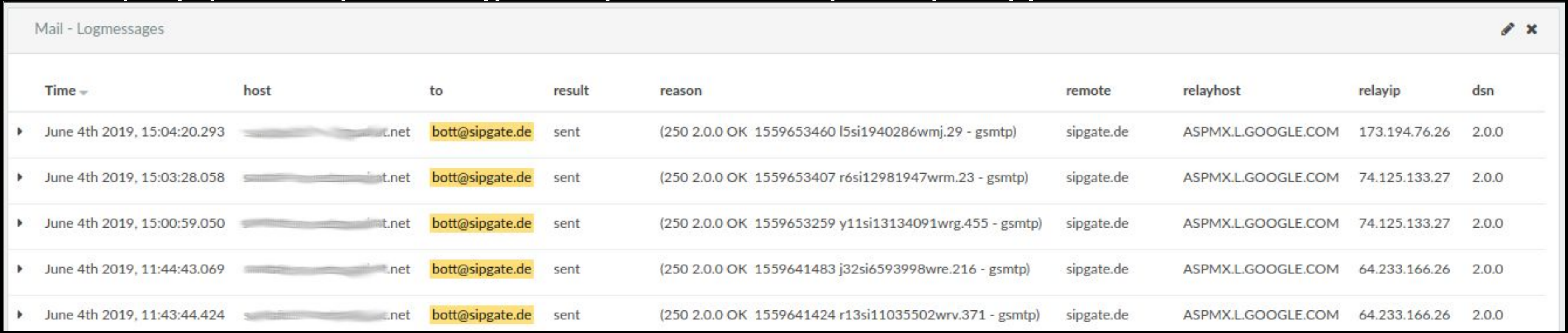
sipgate

ELK-Stack: Postfix/Mail Logging

- simply forward your mail.log using beaver, filebeat etc.
- look for existing postfix grok parser configuration files
- makes it easier for e.g. customer service to debug customer mail problems

ELK-Stack: Postfix/Mail Logging

- simply forward your mail.log using beaver, filebeat etc.



Mail - Logmessages

	Time	host	to	result	reason	remote	relayhost	relayip	dsn
▸	June 4th 2019, 15:04:20.293	[redacted]t.net	bott@sipgate.de	sent	(250 2.0.0 OK 1559653460 l5si1940286wmj.29 - gsmtp)	sipgate.de	ASPMX.L.GOOGLE.COM	173.194.76.26	2.0.0
▸	June 4th 2019, 15:03:28.058	[redacted]t.net	bott@sipgate.de	sent	(250 2.0.0 OK 1559653407 r6si12981947wrm.23 - gsmtp)	sipgate.de	ASPMX.L.GOOGLE.COM	74.125.133.27	2.0.0
▸	June 4th 2019, 15:00:59.050	[redacted]t.net	bott@sipgate.de	sent	(250 2.0.0 OK 1559653259 y11si13134091wrg.455 - gsmtp)	sipgate.de	ASPMX.L.GOOGLE.COM	74.125.133.27	2.0.0
▸	June 4th 2019, 11:44:43.069	[redacted].net	bott@sipgate.de	sent	(250 2.0.0 OK 1559641483 j32si6593998wre.216 - gsmtp)	sipgate.de	ASPMX.L.GOOGLE.COM	64.233.166.26	2.0.0
▸	June 4th 2019, 11:43:44.424	[redacted].net	bott@sipgate.de	sent	(250 2.0.0 OK 1559641424 r13si11035502wrv.371 - gsmtp)	sipgate.de	ASPMX.L.GOOGLE.COM	64.233.166.26	2.0.0

ELK-Stack: DHCP Dashboard

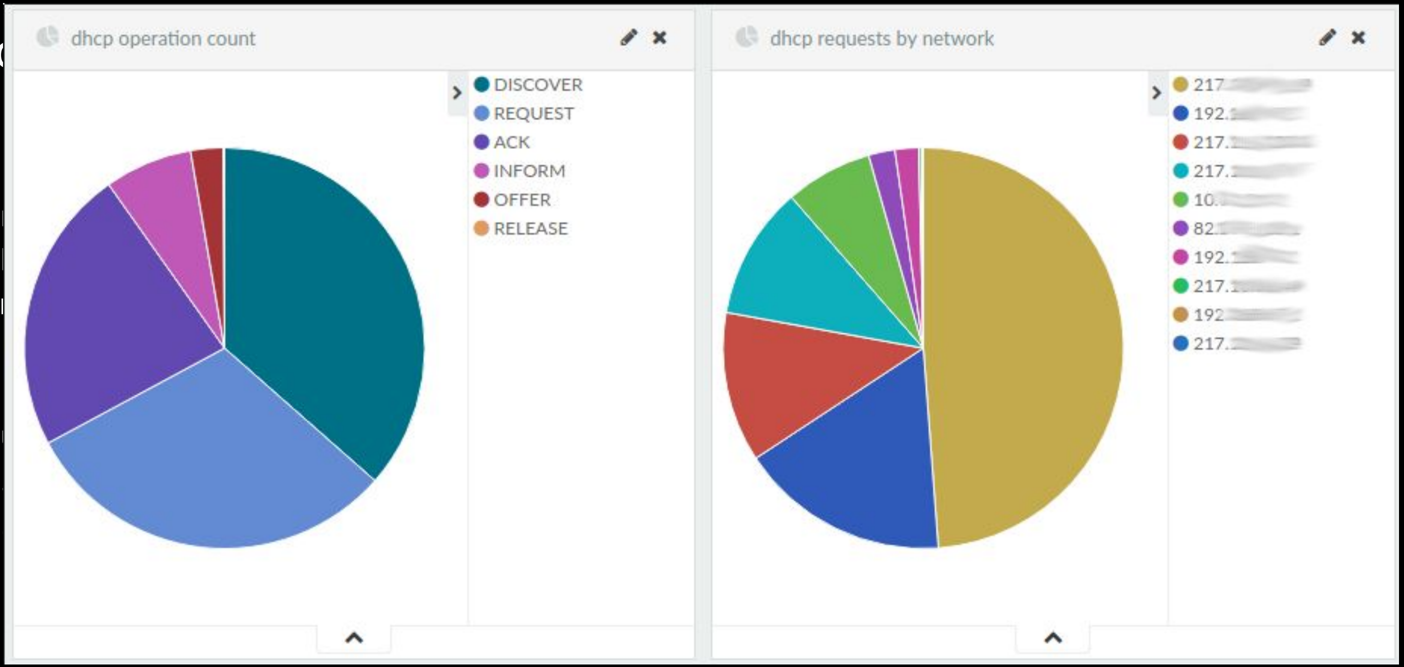
- isc-dhcpd logs to syslog only
- use (e.g.) syslog-ng to redirect dhcpd logging to a separate file:

```
destination df_dhcpd { file("/var/log/dhcpd/dhcpd.log" group(adm) perm(0640) ); };
filter f_dhcpd {
    program("dhcpd");
};
log {
        source(s_all);
        filter(f_dhcpd);
        destination(df_dhcpd);
};
```

ELK-Stack: DHCP Dashboard

- isc-
- use

```
destinatio
filter f_d
    progra
};
log {
        so
        fi
        de
};
```

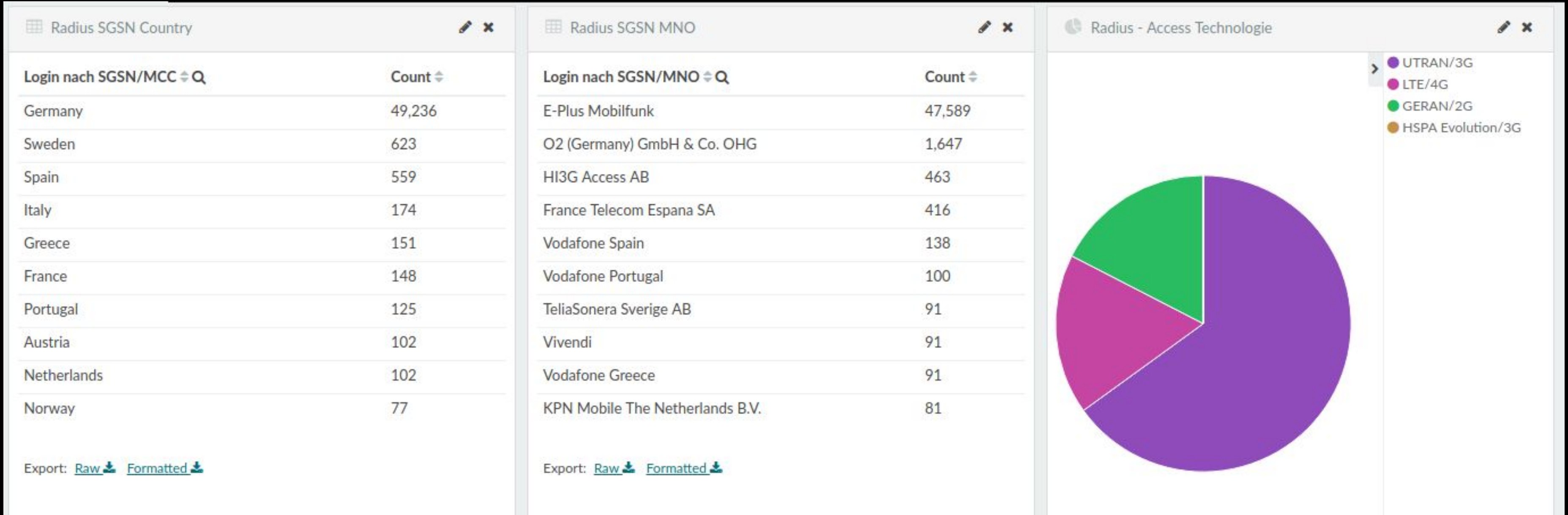


sipgate

ELK-Stack: Radius Dashboard

- freeradius does not log very line-by-line-parser-friendly
- it offers the `linelog` module which is able to log any request field to a separate logfile
- add the information you need and use logstash’s CSV filter

ELK-Stack: Radius Dashboard



Radius SGSN Country

Login nach SGSN/MCC	Count
Germany	49,236
Sweden	623
Spain	559
Italy	174
Greece	151
France	148
Portugal	125
Austria	102
Netherlands	102
Norway	77

Radius SGSN MNO

Login nach SGSN/MNO	Count
E-Plus Mobilfunk	47,589
O2 (Germany) GmbH & Co. OHG	1,647
HI3G Access AB	463
France Telecom Espana SA	416
Vodafone Spain	138
Vodafone Portugal	100
TeliaSonera Sverige AB	91
Vivendi	91
Vodafone Greece	91
KPN Mobile The Netherlands B.V.	81

Radius - Access Technologie

sipgate

ELK-Stack: Firewall Logging

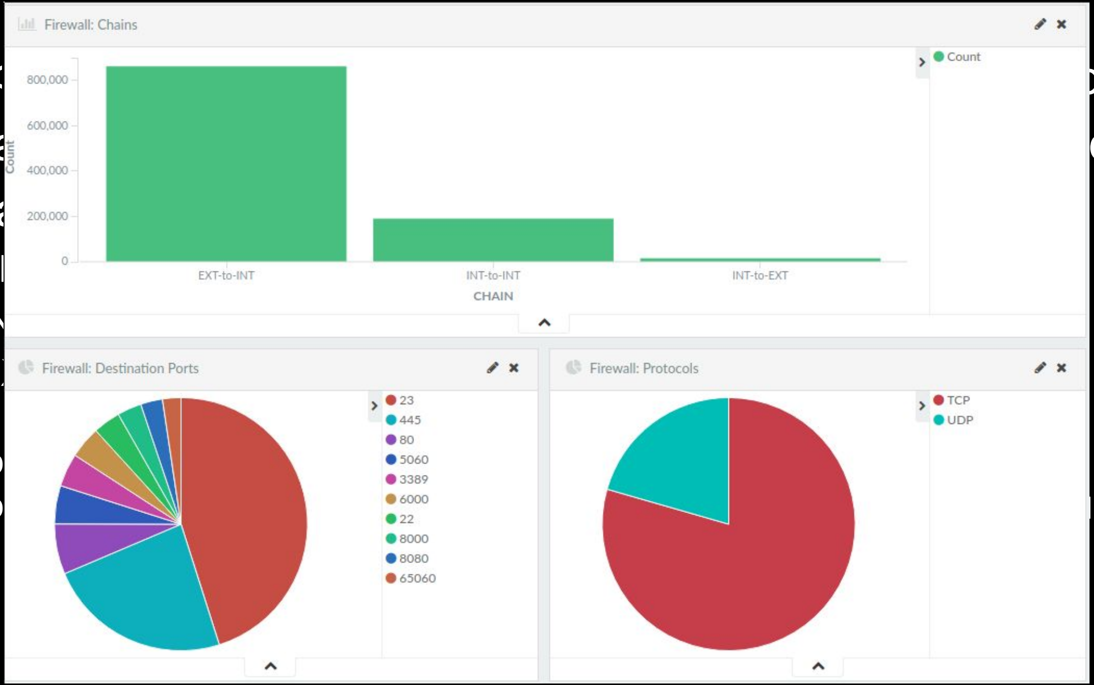
- use ulog (<= jessie) or nflog (>= stretch) for performance reasons
- use fine-grained blocking rules at the end of your chains and add log prefixes to create traffic categories for later use in Kibana, e.g.:
    - EXT-to-INT
    - INT-to-INT
    - INT-to-EXT

```
iptables -A FORWARD --nflog-group 1 --nflog-prefix "INT-to-INT" -j NFLOG
iptables -A FORWARD --ulog-nlgroup 1 --ulog-prefix "INT-to-INT" --ulog-qthreshold 1 -j ULOG
```

ELK-Stack: Firewall Logging

- use ulog (< ons
- use fine-gra dd log prefixes to create tra
  - EXT-to-I
  - INT-to-IN
  - INT-to-E

```
iptables -A FORWARD
iptables -A FORWARD                                              1 -j ULOG
```


Firewall: Chains

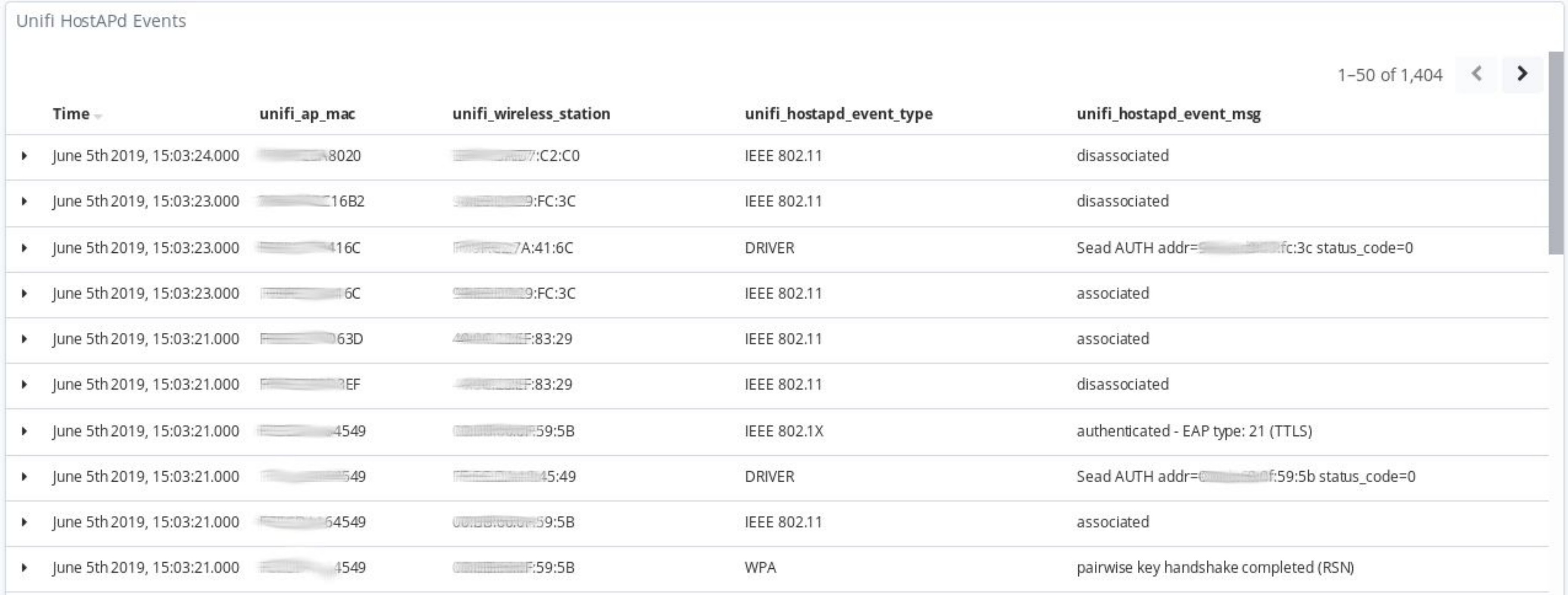
Firewall: Destination Ports

Firewall: Protocols

ELK-Stack: Unifi Wireless Logging

- configure your Unifi gear to send syslog to your logstash instance
- parse hostapd messages with a grok filter
- debug wireless client roaming/authentication methods used etc.

ELK-Stack: Unifi Wireless Logging



Unifi HostAPd Events

1–50 of 1,404

	Time	unifi_ap_mac	unifi_wireless_station	unifi_hostapd_event_type	unifi_hostapd_event_msg
▸	June 5th 2019, 15:03:24.000	[redacted]8020	[redacted]7:C2:C0	IEEE 802.11	disassociated
▸	June 5th 2019, 15:03:23.000	[redacted]16B2	[redacted]9:FC:3C	IEEE 802.11	disassociated
▸	June 5th 2019, 15:03:23.000	[redacted]416C	[redacted]7A:41:6C	DRIVER	Sead AUTH addr=[redacted]fc:3c status_code=0
▸	June 5th 2019, 15:03:23.000	[redacted]6C	[redacted]9:FC:3C	IEEE 802.11	associated
▸	June 5th 2019, 15:03:21.000	[redacted]63D	[redacted]F:83:29	IEEE 802.11	associated
▸	June 5th 2019, 15:03:21.000	[redacted]EF	[redacted]F:83:29	IEEE 802.11	disassociated
▸	June 5th 2019, 15:03:21.000	[redacted]4549	[redacted]F:59:5B	IEEE 802.1X	authenticated - EAP type: 21 (TTLS)
▸	June 5th 2019, 15:03:21.000	[redacted]549	[redacted]45:49	DRIVER	Sead AUTH addr=[redacted]f:59:5b status_code=0
▸	June 5th 2019, 15:03:21.000	[redacted]64549	[redacted]59:5B	IEEE 802.11	associated
▸	June 5th 2019, 15:03:21.000	[redacted]4549	[redacted]F:59:5B	WPA	pairwise key handshake completed (RSN)

sipgate

Thanks for listening.

Questions?

sipgate

Image Links/Sources

- official logos of
  - Ansible - https://www.ansible.com/
  - Aptly - https://www.aptly.info/
  - Elasticsearch/Logstash - https://www.elastic.co/de/brand
  - Github - https://github.com
  - Jenkins - https://jenkins.io/
  - Redis - https://redis.io/
- https://media.giphy.com/media/1oJLpejP9jEvWQlZj4/giphy.gif
- https://giphy.com/gifs/lA3qoZE4TKQhi
- other Icons: the Noun Project - https://thenounproject.com/