

RSPAMD

Neue Konzepte im Antispam

→ **Heinlein Support**

- IT-Consulting und 24/7 Linux-Support mit ~35 Mitarbeitern
- Eigener Betrieb eines ISPs seit 1992
- Täglich tiefe Einblicke in die Herzen der IT aller Unternehmensgrößen

→ **24/7-Notfall-Hotline: 030 / 40 50 5 - 110**

- Spezialisten mit LPIC-2 und LPIC-3
- Für alles rund um Linux & Server & DMZ
- Akutes: Downtimes, Performanceprobleme, Hackereinbrüche, Datenverlust
- Strategisches: Revision, Planung, Beratung, Konfigurationshilfe

Rspamd - Übersicht

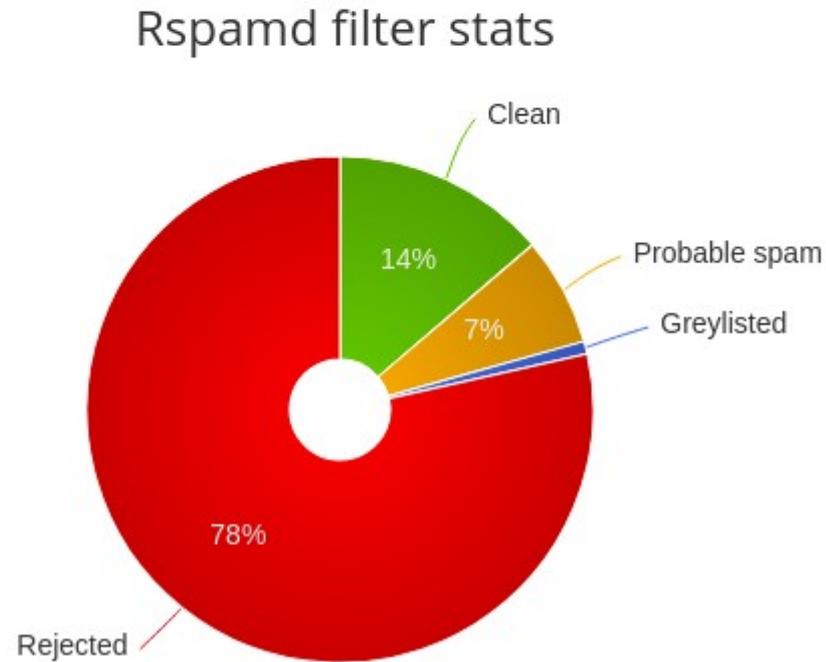
- Viele moderne Ansätze
- Augenmerk auf Geschwindigkeit und Effizienz
- Präferiert dynamische Erkennungstechniken über statischen Regeln
- Funktionen zur Selbstoptimierung
- Event-Driven Processing model
- Kern in C - Erweiterungen und Regeln in Lua
- Kommunikation über HTTP REST
- Redis Datenbank
- Webinterface

 All SERVERS ▾ Status Throughput Configuration Symbols Learning Scan History

Refresh Disconnect

7k Scanned	1k Clean	63 Greylist	543 Probable	6k Reject	144 Learned	1.7.0 Version	3hr 0min Uptime
----------------------	--------------------	-----------------------	------------------------	---------------------	-----------------------	-------------------------	---------------------------

Statistics



Rspamd - Funktionen

→ Features:

- ARC, ASN, AntiVirus, Bayes, Clustering, DCC, DKIM, DMARC, Elastic, Fuzzy, Graphite, Greylisting, Hfilter, ICAP, IPScore, Lua Functions, Lua Rules, MailingLists, Metadata Exporter, Metric Exporter, Mime Types, Multimaps, MX Checks, NeuralNetwork, Oletools, Phishing, RBL, Razor/Pyzor (external), Received Policy, Redis, Replies, Reputation, Rescore, Selectors, SPF, SURBL, Spamassassin Rules, Spamtrap, Upstreams, TrieMatcher, URL Redirector, Whitelist

→ Optimierung:

- AST (Abstract Syntax Tree), Hyperscan, Greedy Reorder Algorithm, PCREJit, LuaJit
- Stoppt die Ausführung wenn der Reject Score erreicht wurde
- Kurzlaufende Checks werden zuerst ausgewertet
- Pre-Filter (Greylisting) können (teure) Standard-Checks verhindern
- Caching vieler Ergebnisse im Redis
- Nutzung von HTTPCrypt anstatt HTTPS

Rspamd - Funktionen #2

- Grundlegende an Spamassassin angelehnte Rules
- Eigene RBL, E-Mail-Hash DB, Fuzzy, ASN Lookups @rspamd.com
- Lokale dynamische Reputationen: IPScore, Bayes, Fuzzy, Clustering, MX-Check, adaptive Ratelimit, generic Reputation, Replies
- Geo IP Support
- DKIM + ARC Signing Support
- Externe Services: Razor, Pyzor, DCC, ICAP, Oletools, Spamassassin
- Actions, Force Actions
 - Voller Milter Feature Support

Rspamd Detail - Actions

→ Actions - Ausführung bei Überschreitung best. Grenzwerte

```
actions {  
  
    greylist =           8;  
    rewrite_subject =   12;  
    add_header =        13;  
    reject =            15;  
  
};
```

Rspamd Detail - Actions / Force Actions

→ Force Actions - Ausführung bei bestimmten Symbolen

```
WHITELIST_EXCEPTION {  
  
    action = "reject";  
    expression = "IS_IN_WHITELIST & CLAM_VIRUS";  
    message = "Rejected due to suspicion of virus";  
  
}
```

Rspamd Detail - Multimaps

- Match Listen
- Abruf als HTTP, Lokale Datei oder aus dem Redis
- Rspamd prüft selbstständig auf Änderungen
- Spezielle, optimierte Map Typen für bestimmte Abfragen
 - z.B. Header, E-Mail-Adresse, IP, Received, URL
 - `asn, content, country, dnsbl, filename, from, header, hostname, ip, mempool, received, rcpt, symbol_options, url`
- Allgemeingültige (aber damit teurere) Prüfung
- Können Symbole und Scores setzen

Rspamd Detail - Multimaps

- Beispiel Prüfung eines speziellen Headers:

```
amavis_result {  
    type = "header";  
    header = "X-Spam-Score";  
    map = "file://$LOCAL_CONFDIR/maps.d/amavis_result.map";  
    symbol = "AMAVIS_RESULT";  
    filter = "regexp:/^[0-9]+/";  
    score = 1.0;  
}
```

- Map Datei amavis_result.map

```
6      AMAVIS_RESULT:4  
7      AMAVIS_RESULT:7  
8      AMAVIS_RESULT:8
```

- X-Spam-Score: 7.709

Rspamd Detail - Multimaps

- Beispiel Heinlein Header Checks als Map

```
hs_headers {  
  
    type = "content";  
    map = "file://$LOCAL_CONFDIR/maps.d/hs_headers.map";  
    symbol = "HS_SA_HEADERS";  
    score = 1.0;  
    filter = "headers"  
    regexp = true;  
  
}
```

- Map Datei hs_headers.map

/Content-Disposition.*Multipart message/	HS_SA_HEADERS:3
/Date.* \[AP\]M/	HS_SA_HEADERS:2
/Date.*[+-](1[4-9] 2d)dd\$/	HS_SA_HEADERS:5
/From.*icyhot\.bakas24\.de/	HS_SA_HEADERS:4
/Subject .*You wanna check how good in bed I am\?.*/	HS_SA_HEADERS:5

Rspamd Detail - User Settings

- Settings werden in Maps definiert (HTTP Abruf möglich)
- Alternativ: Settings im Redis
- Existieren mehrere zutreffende Einstellungen (z.B. IP, User, Domain, Von, An, Header Match ...) werden diese nach Priorität zusammen gefasst (gemerged)
- Was kann eingestellt werden?
 - Grenzwerte (actions)
 - Score für einzelne Symbole
 - Symbole oder Gruppen aktivieren oder deaktivieren
- Deaktivierung bestimmter Symbole verhindert die Ausführung der Callback Funktion
 - z.B. keine RBL, kein Antivirus

Rspamd Detail - User Settings #2

Postfix: -o milter_macro_daemon_name=*testsetting*

```
TEST {
    id = "test";
    priority = high;
    request_header = {
        "MTA-Tag" = "testsetting";
    }

    apply {
        #actions {
            # reject = 17.0;
            # "add header" = 12.0; # Please note the space, NOT an underscore
        #}

        symbols_disabled = [
            "SOPHOS_VIRUS",
            "CLAM_VIRUS",
            "DCC_BULK",
            "RAZOR",
            "PYZOR",
        ];
    }
    # Always add these symbols when settings rule has matched
    symbols ["TEST"]
}
```

Rspamd Detail - IPScore (neu Reputation)

- **IP_SCORE(4.28) [ip: (9.91), ipnet: 89.163.128.0/17(7.24), asn: 24961(4.28), country: DE(-0.04)]**
- Mitlernende persönliche IP-Reputationsdatenbank
 - RBL + GeoIP
- Der Score einer Nachricht wird mit entsprechend absteigender Gewichtung für die IP, das IP-Netz, ASN (Besitzer), Land gespeichert
- Je negativer eine IP, ein IP-Netz... bewertet ist und desto höher die restliche Bewertung der Mail ist, desto höher ist der IPScore
 - $ip_score = action_multiplier * \tanh(e * (metric_score/score_divisor))$
 - Scores { ip = 1.0; ipnet = 0.8; asn = 0.5; country = 0.1; }

Rspamd Detail - adaptive Ratelimit

- Composeable: asn, bounce, from, ip, user, rip, to
 - ip = "1000 / 1min";
 - from_to = "200 / 1min";
 - asn = "10000 / 1min";
 - rip = "500 / 1min";
 - User = "500 / 1min"

- Adaptive Anpassung je nach Spam-Erkennung
 - ham_factor_rate, spam_factor_rate

- Inbound: verrückte Server, Bots
- Outbound: gehackte Accounts

Rspamd Detail - Greylisting

- Soft Reject anhand des Scores oder Force Action oder ...
- z.B. Greylisting bei 10+ Punkten
 - Newsletter & Co kommen wieder
 - Bots kommen nicht wieder
- E-Mails mit guter Reputation kommen sofort durch
- Läuft als Pre- und Post-Filter
 - Hosts (Triples) im Greylisting werden früh zurück gewiesen
 - Durch Redis haben alle Hosts das gleiche Wissen
 - Greylisting Entscheidung nach Mail-Auswertung

Rspamd Detail - Neural Network

- Neuronales Netzwerk, KI
- Entdeckt Symbole und Zusammenhänge, die typischerweise in Spam oder Ham vor kommen
- Post-Filter - läuft nach der normalen Erkennung
- Fügt auf Basis seiner Erkenntnisse einen positiven oder negativen Score hinzu
- Lernt immer wieder neu (alte Daten werden verworfen)
- Mehrere Profile möglich (short_time, normal, long_time)
 - Erkenntnisse der letzten 100 Tage
 - Erkenntnisse der letzten 10 Tage
 - Erkenntnisse der letzten 2 Tage

Rspamd Detail - Spamtrap

- Lernt alle Mails an definierte E-Mail Adressen gleich als SPAM (Bayes, Fuzzy, Torch)
- Kann diese danach immer rejecten oder ...
- Super, wenn man viele Mails an Phantasie-Adressen bekommt

Rspamd Detail - MX Check + HFILTER

- Prüft Domain des SMTP From (mail from:) oder alternativ SMTP Helo auf korrekte MX Server
- HFILTER: macht typische DNS Test wie typischerweise Postfix Restrictions
 - # reject_non_fqdn_sender
 - # reject_non_fqdn_recipient
 - # reject_unknown_sender_domain
 - # reject_unknown_recipient_domain
 - # reject_invalid_hostname
 - # reject_invalid_helo_hostname
 - # reject_non_fqdn_helo_hostname
- MX Check prüft per SMTP connect ob zumindest ein MX der Domain erreichbar ist

Rspamd Detail - Modules - Lua Functions

- Lua Erweiterungen speziell für Rspamd
- Wird vor allem bei Rspamd Plugins verwendet
- Task - Laufzeit-Daten
- TCP Client
 - Einfache zu implementierende TCP Kommunikation. Genutzt z.B. bei Antivirus, HTTP, SMTP-Tests, Razor, Pyzor, DCC ...
- Redis - Daten in Redis speichern und lesen
- Upstream - Verwaltung multipler Endpunkte (HA)
- Außerdem: Mime, UCL (Config), Regex, IP, HTTP, DNS, Logging, Crypto (Hashes), URL, Maps

Rspamd Detail - Config merging

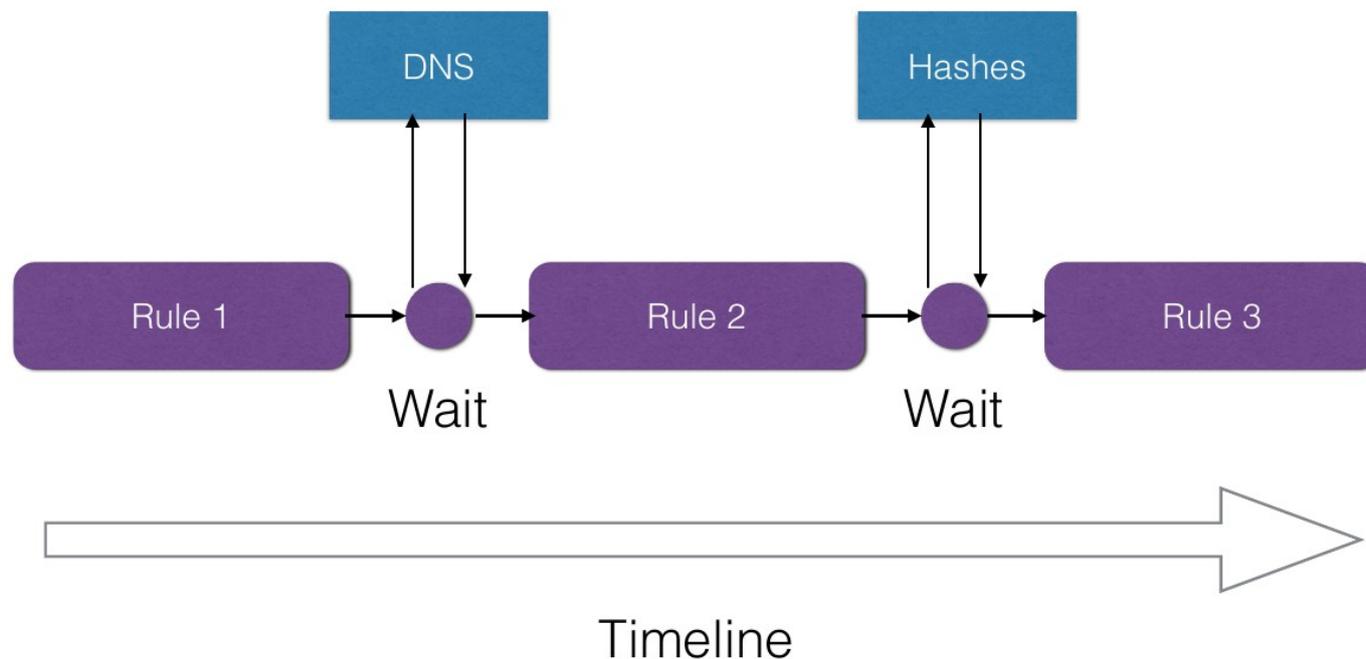
- Configs sind in Universal Configuration Language (UCL)
- Updateproblematik? Ersetzen? Diff? Patch?
- MERGE! Default Config + User Config
- Rspamd / UCL unterstützt includes
- z.B. Antivirus.conf:

```
.include(try=true,priority=5) "${DBDIR}/dynamic/antivirus.conf"  
.include(try=true,priority=1,duplicate=merge)  
                                "$2_CONFDIR/local.d/antivirus.conf"  
.include(try=true,priority=10)  
                                "$LOCAL_CONFDIR/override.d/antivirus.conf"
```

Rspamd Detail - Event Driven Model

Sequential processing

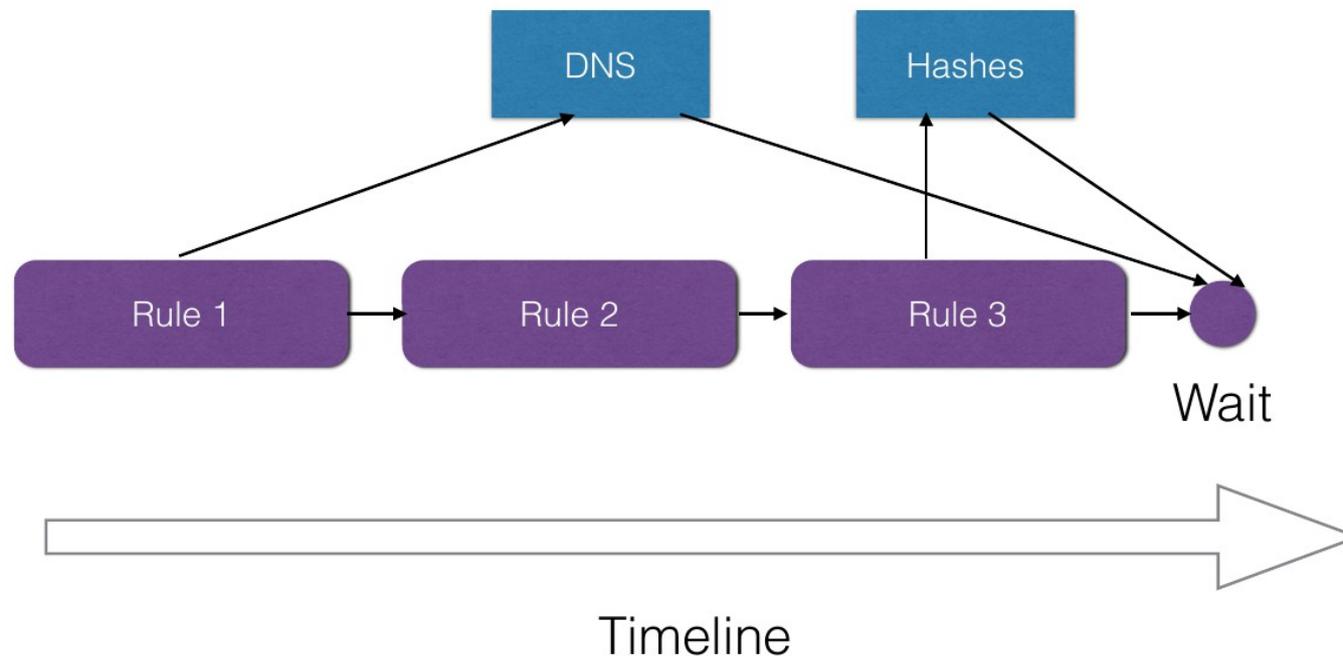
Traditional approach



Rspamd Detail - Event Driven Model

Event driven model

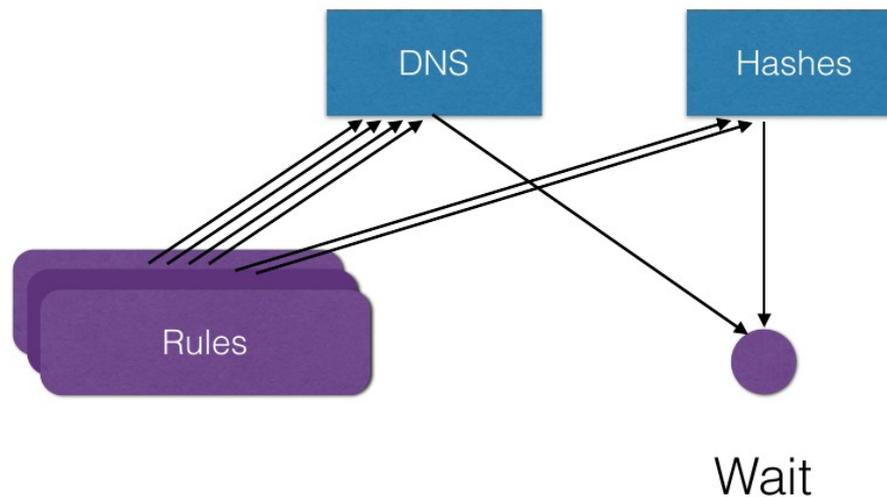
Rspamd approach



Rspamd Detail - Event Driven Model

Event driven model

What happens in the real life



Rspamd Stages / Processing

- **Pre-filters:** checked every time and can stop all further processing by calling `task:set_pre_result()`
- **All symbols*:** can depend on each other by calling `rspamd_config:add_dependency(symbol1, symbol2)`
- **Statistics:** is checked only when all symbols are checked
- **Composites:** combine symbols to adjust the final results; pass 1
- **Post-filters:** are executed even if a message is already rejected and symbols processing has been stopped
- **Composites:** combine symbols to adjust the final results; pass 2
- **Idempotent:** execute rules that cannot change result (e.g. data exporters)

Rspamd - was gibt's nicht?

- SQL, LDAP Support
- Quarantäne
- Externe Programme mit Shellaufruf (clamscan ...)
- Entpacken von Anhängen
- ~~Body einer Mail ändern~~
 - Ab 1.9.0 möglich
- Reject bei mail from oder rcpt to
 - Es wird immer End of Data abgewartet
- Brauchen wir eigentlich aber auch nicht wirklich ;-)
- <https://www.helein-support.de/sites/default/files/rechtsfragen-fuer-postmaster.pdf>

Rspamd: Kommandozeile

- `rspamdadm configtest` - zusammengefasste Config und Werte auf Fehler prüfen
- `rspamdadm configdump` - zusammengefasste Config anzeigen
- `rspamc -learn_spam < spam.txt` - Mail als Spam lernen (Bayes)
- `rspamc -learn_ham < ham.txt` - Mail als Ham lernen (Bayes)
- `rspamc stat` - Einige Statistiken anzeigen
- `rspamc < file.txt` - Spamerkennung auf Text ausführen

Rspamd - Lessons Learned

- Dynamische Module (Bayes, IPScore ...) benötigen einige Zeit zum Lernen - Erkennung wird immer besser
- UCL Config Dateien müssen sauber geschrieben werden - sonst wird der Wert nicht erkannt oder richtig gemerged oder schlimmer die ganze Config ist defekt
 - Manchmal wird eine defekte Config von *rspamadm configtest* nicht erkannt
- Redis wird bei großen Installationen unangenehm
 - Heinlein 120 Mio Keys - 25 GB Database
 - Aufteilung der Daten auf mehrere Redis-Instanzen hilft

Rspamd vs. Amavis Erkennungsrate

- And The Winner Is: ähmm Depends.
- Frisch Installiertes Amavis/Spamassassin vs. frisches Rspamd?
 - Amavis erkennt durch die vielen statischen Regeln erstmal besser
 - Nach kurzer Lernphase ist Rspamd besser
- Optimiertes Amavis/Spamassassin vs. Angelerntes Rspamd?
 - **Rspamd !!!**
- Insgesamt Vorteile für Rspamd durch dynamischere Charakterisierung vor Allem bei neuen Spamwellen
- Statische Regeln oder Signaturen hinken der Realität immer etwas hinterher
- Zukunft? Definitiv Rspamd!

Rspamd vs. Amavis Administration, Flexibilität, Erweiterbarkeit

- Alles steckt in einem System
- Alles läßt sich zur Laufzeit anpassen
- Kleine eigene Funktionen können alles während eines Scan-Vorgangs anpassen oder weiter verarbeiten
- Redis: alle Systeme sind auf dem gleichen Wissenstand
- Scale-Out sehr einfach möglich
- Eigenes Framework für Erweiterungen

Rspamd - Installation

- <https://rspamd.com/downloads.html>
- Hauptentwickler möchte keine durch Distributionen „optimierte“ Pakete
- Alle Abhängigkeiten im Git verfügbar
- Offizielle Pakete für Fedora/CentOS, Ubuntu/Debian
- Ebenfalls verfügbar für Alpine Linux, Arch Linux, Gentoo Linux, OpenSUSE, Void Linux, BSD
- Offizielle Pakete sind in Stable (1.7.7) und Experimental (Nightly Builds - 1.7.8-0~git175~d14da77f0~sid) verfügbar

Amavis / Rspamd Migration

- Milter und SMTP-Proxy arbeiten nicht zusammen
- Amavis auf Milter umstellen (amavisd-milter)
- Die Milter Schnittstelle kann mehrere Milter nacheinander ausführen
- Dabei wird bei jedem SMTP Zustand erst Milter1 danach Milter2 kontaktiert
- Milter2 kann also auf die Ergebnisse von Milter1 zugreifen
- z.B: **smtpd_milters=inet:RSPAMD:11332,inet:AMAVIS:10030**
- Milter1 sollte nur scannen und Header einfügen
- Milter2 muss als letzte Instanz die Aktion ausführen

Amavis / Rspamd Migration #2

- Milter2 = Amavis kann nun die Header von Milter1 = Rspamd auswerten
 - X-Spamd-Bar: ++++++
 - X-Spamd-Result: default: False [12.99 / 200.00]
 - TO_DN_NONE(0.00)[]
 - ...
- Amavis könnte Punkte für das Rspamd Ergebnis vergeben
 - header RSPAMD_12_13 X-Spamd-Result =~ /default: False \[1[2-3]\.\/i
 - describe RSPAMD_12_13 RSPAM 12.00-13.99
 - score RSPAMD_12_13 1

Amavis / Rspamd Migration #3

- Nach der Eingewöhnung setzt man
 - Milter1 = Amavis
 - Milter2 = Rspamd
- Jetzt muss Rspamd aber auch die Aktionen auslösen und ggf. die Header vom Amavis auswerten
- Eventuelle alte Spamassassin Rules können mit dem spamassassin Modul des Rspamd direkt eingebunden werden
 - Besser aber: in Rspamd Maps umwandeln

Rspamd im Enterprise (Multi-Server)

- Rspamd - Milter (worker-proxy) kann unabhängig vom Scanner (worker-normal) laufen
- Viele Module machen externe Calls (DNS, TCP, Maps, SMTP)
- Aufteilung der Last und Funktionen auf verschiedene Server
- Rules (Multimaps) oder Settings lassen sich von externen HTTP Servern laden
- Redis bietet eingebaute (Master-Slave) Replikation
- Worker-Proxy kann auch Entwicklungsserver zusätzlich befragen und das Ergebnis informativ mit loggen

Rspamd im Enterprise - Upstreams

- Lua Funktionen um multiple Gegenstellen zu verwalten
 - z.B. Liste von Rspamd Scanner Servern, Liste von ClamAV Servern ...
- Eingebaute HA Funktionalität
 - Nicht erreichbare Gegenstellen werden temporär deaktiviert und selbstständig später neu aufgelöst
- Alternierung
 - master-slave
 - round-robin
 - random
 - sequential
 - hash
- Angegebene Prioritäten fließen immer mit ein

Rspamd im Enterprise - Upstreams #2

- Beispiel worker-proxy
 - server = "master-slave: 10.0.0.1:11333:10, 10.0.0.2:11333:5, 10.0.0.3:11333:2"
 - Versucht immer zuerst 10.0.0.1
 - Wenn nicht erreichbar - Häufigkeit der Slaves nach Priorität 5x 10.0.0.2, 2x 10.0.0.3

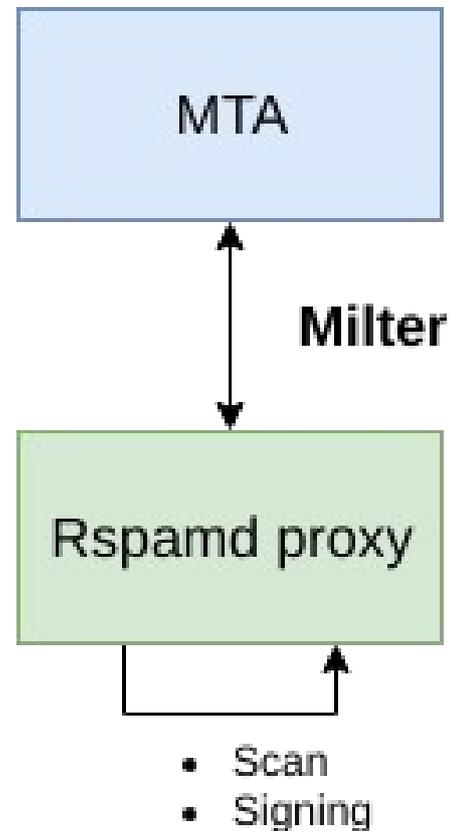
- Nutzung von DNS Records
 - Multiple Records werden für HA genutzt, ABER nicht zur Lastverteilung
 - Bei Ausfall der ersten IP, wird eine andere aus dem Record genutzt

- Nutzung von Upstreams:
 - Proxy ↔ Scanner
 - DNS Server
 - Redis Server
 - Anti-Virus Server
 - DCC Server

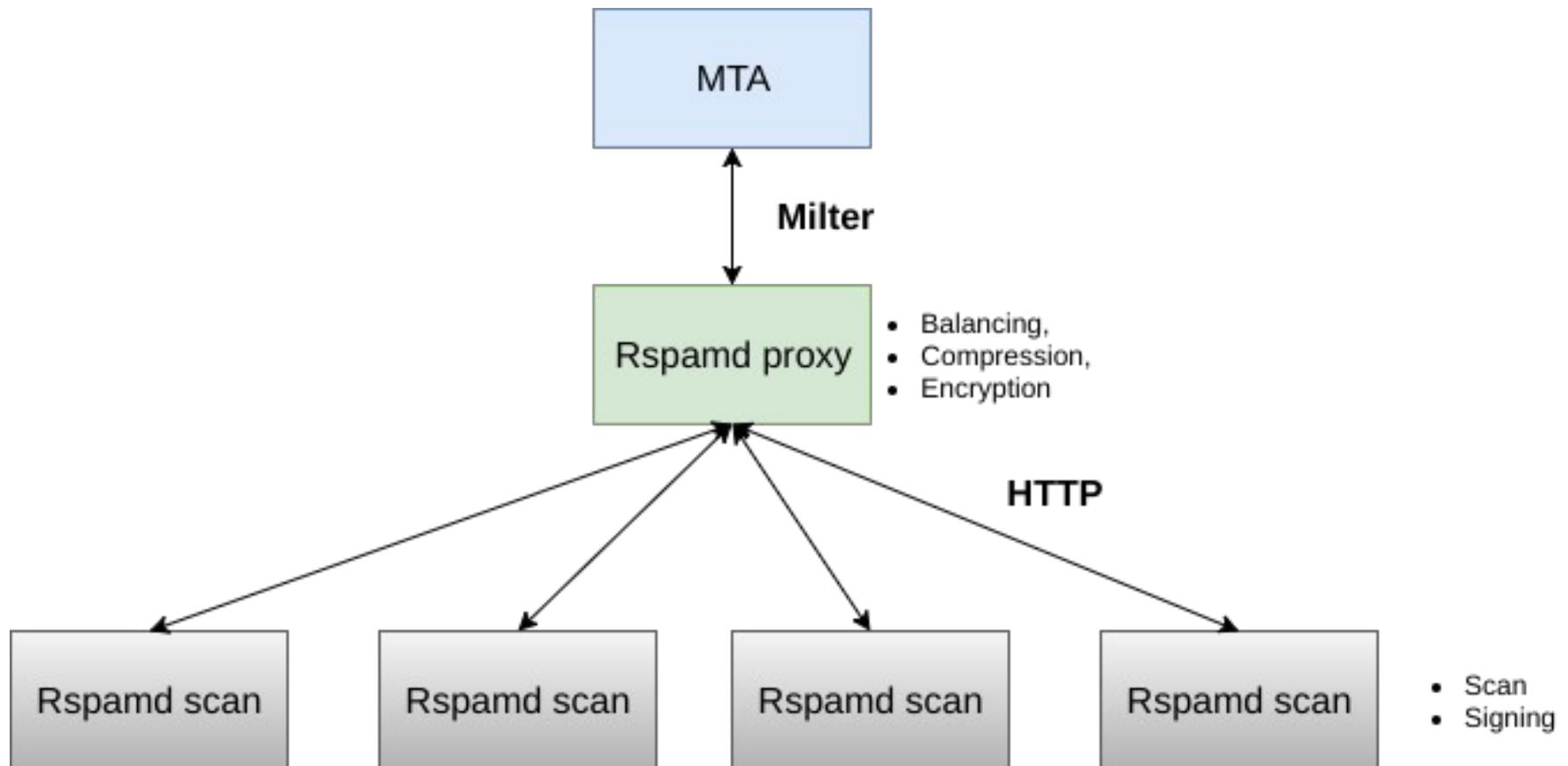
Rspamd im Enterprise - Redis

- Redis Master-Slave Replikation funktioniert super
- Redis Master-Master ist experimentell
- Alternative: Redis Sentinel, Netflix Dynamite
- Aber: Rspamd kann auch zwischen Read-Server und Write-Server unterscheiden
- Vorteil: Alle Rspamd Server haben die gleiche Datenbasis
- Lösung: Lese wenn möglich von Localhost, schreibe auf den Master Server
 - read-servers = „master-slave: 127.0.0.1, 10.0.0.2, 10.0.0.3“
 - Write-servers = „10.0.0.1“

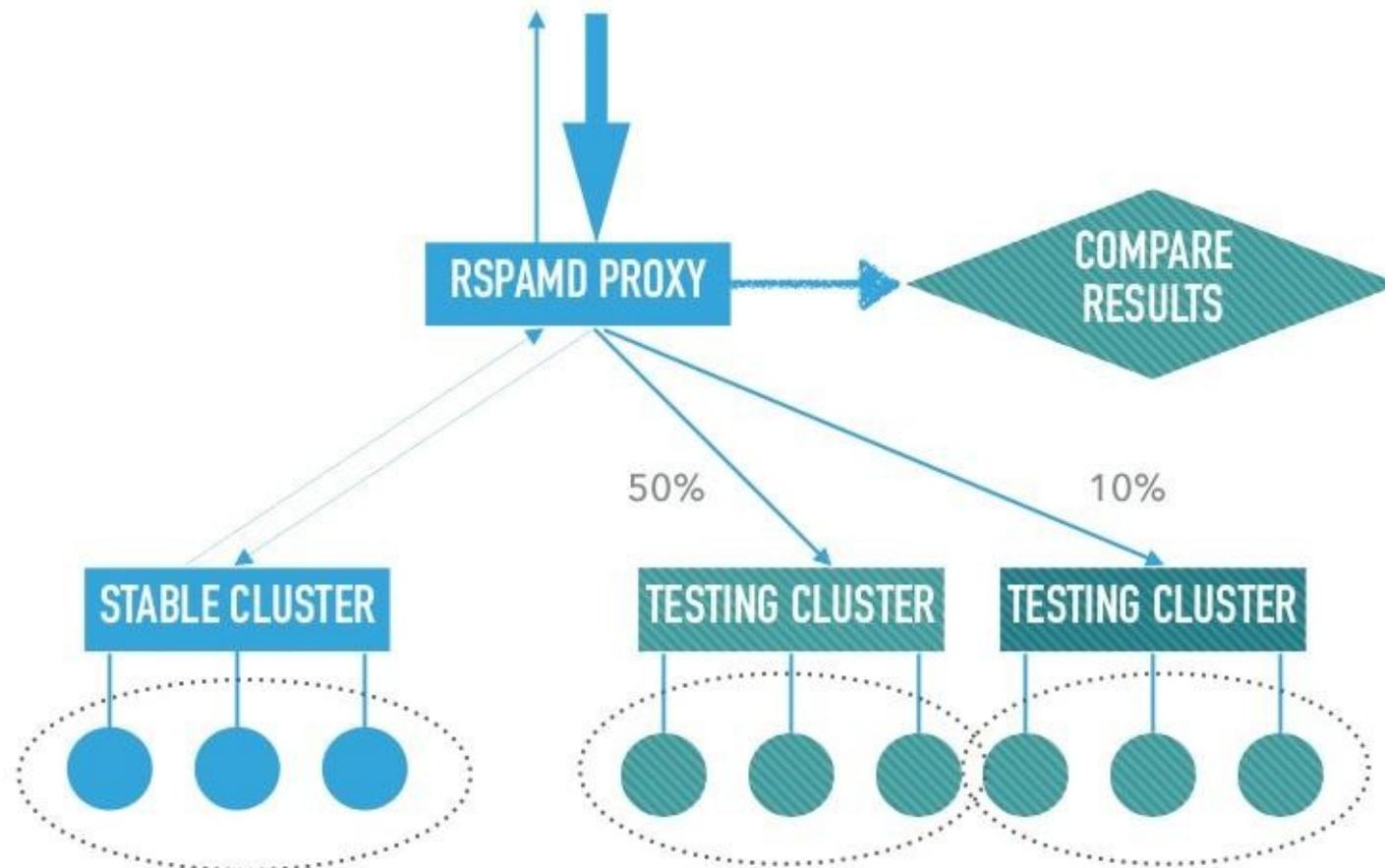
Rspamd - Singleserver



Rspamd - Multiserver

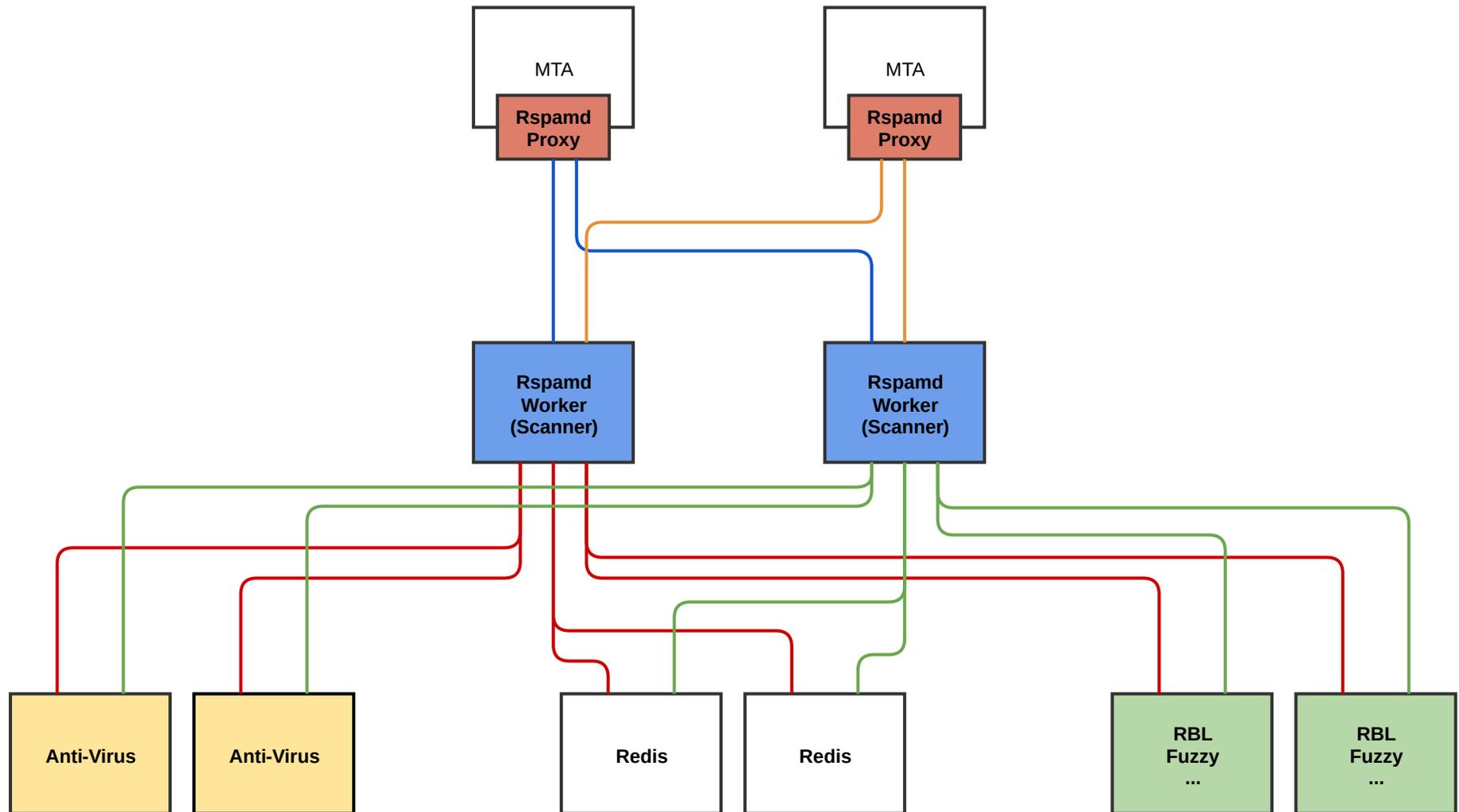


Rspamd - Produktiv und Entwicklung



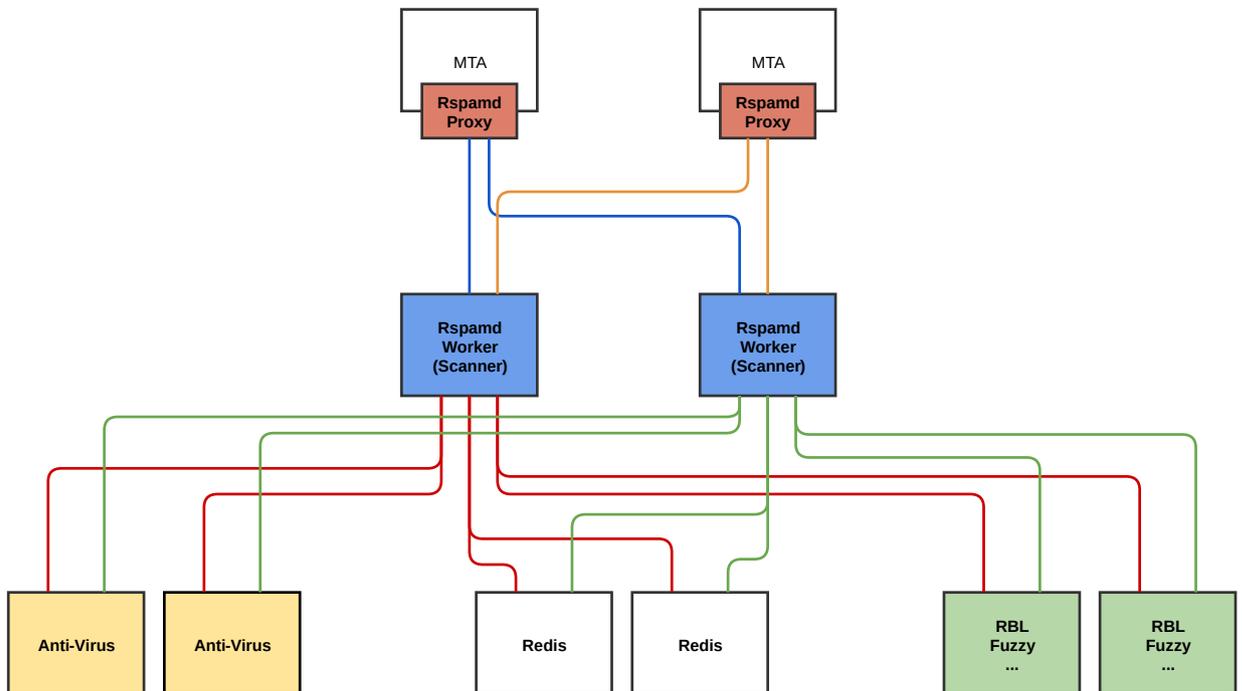
Balance within clusters

Rspamd - Cluster mit Upstreams



Rspamd im Enterprise - Scale out

- Einfach links und rechts zusätzliche Maschinen der jeweiligen Art hinzufügen
- Nutzung von DNS oder einer Automatisierung vereinfacht den Aufwand massiv.



Rspamd im Enterprise - Webinterface

- Zusammenfassen aller Rspamd Scanner durch Config Option neighbours
- Alle Server werden im Webinterface angezeigt
- Zusammenfassung oder einzelne Server anzeigbar
- Die gescannten Mails und Fehlermeldungen aller Server verfügbar
- Dynamische Einstellungen für alle Server ausführen

Rspamd & Auswertung / Monitoring

- Metadata Exporter → HTTP, Email, Redis
- Metric Exporter → Graphite
- Clickhouse → Clickhouse Database
 - „ClickHouse is an open source column-oriented database management system capable of real time generation of analytical data reports using SQL queries.“
<https://clickhouse.yandex/>
- ElasticSearch / Kibana → Metricdaten
- Nagios & Co: Via REST API abfragbar

Rspamd @ Heinlein

- Heinlein entwickelt Rspamd aktiv mit
 - <https://github.com/rspamd/rspamd>
- Ein Rspamd Scan-Cluster für alle Marken und Kunden
- Kundenspezifische Settings zur Anpassung
- Rollout, Config und Skalierung aller Komponenten mit Ansible
- Rspamd teilweise noch mit Amavis
 - Heinlein User-spezifische Settings noch nicht komplett für Rspamd
- Todo:
 - Export der Heinlein Spamassassin Regeln direkt als Map für Rspamd
 - Aufbau eines Spamtraps / eigene Fuzzy Datenbank

Rspamd - ein neuer Ansatz für moderne Mail - Infrastrukturen?

- Rspamd im Cluster stellt einen performanten Scan-Pool mit einheitlicher Datenbank zur Verfügung
- Rest API kann von MTA's oder anderen Tools genutzt werden
 - Skripte: Sieve, Bash ...
 - Webmailer: z.B. bei Ansicht Mail noch einmal scannen
- Rspamd lernt aus jeder (Spam-) E-Mail, die er zu Gesicht bekommt
 - Aufbau eigener Reputationen (IPScore, Neural Network ...)
- Rspamd kann auf einfache Weise externe Services anbinden und nutzen
- Rspamd stellt Funktionen bereit, die sonst fest mit dem MTA verbunden sind
 - Greylisting, MX-Checks, RBL, Ratelimit

Rspamd - ein neuer Ansatz für moderne Mail - Infrastrukturen? #2

- Viele Setups lehnen offensichtlichen Spam aber frühest möglich ab
- Aus diesen ganz schlechten Spams kann der Rspamd aber auch viel lernen
- Abschalten typischer Maßnahmen ...
 - Postscreen
 - PostGrey, Policyd-Weight, Postfwd
 - RBL Rejects
 - Restrictions: reject_non_fqdn_sender, reject_unknown_sender_domain ...
 - Ratelimit im MTA
- ... und Rspam nach dem Scannen der Mail rejecten lassen
- Aber Sicherstellen, dass Rspamd die Funktion dann abdeckt und forciert
- Zusätzlicher Ressourcenverbrauch vernachlässigbar

Soweit, so gut.

**Gleich sind Sie am Zug:
Fragen und Diskussionen!**

Tools für Mailserver - SWAKS

- Ihr arbeitet an Mailservern? Ihr braucht dafür die richtigen Werkzeuge zum schnellen und effizienten Testen
- Swaks - Swiss Army Knife for SMTP - <http://www.jetmore.org/john/code/swaks/>
- Swaks kann smtp/lmtp mit und ohne TLS/STARTTLS, Authentifizierung, Anhänge, besondere Header, Abbrechen des Tests an einem bestimmten Punkt (z.B. nach RCPT to) ...
- Zeigt komplette SMTP/LMTP Kommunikation auf der Konsole
- Benötigt Perl

```
apt / yum / dnf / zypper install swaks
```

Tools für Mailserver - Magics mit SWAKS

- Mail von root an kurs

```
swaks -s 127.0.0.1 -f root -t kurs
```

- Mail an remote Server

```
swaks -t kurs@carsten.post.fix
```

- Rawmail versenden

```
swaks -s 127.0.0.1 -t kurs@dennis.post.fix --data - < ./rawmail.txt
```

- Attachment anhängen

```
swaks -s 127.0.0.1 -t kurs@dennis.post.fix --attach  
./hello_world.exe -suppress-data
```

- SSL / Starttls

```
swaks -s 127.0.0.1 -t kurs@dennis.post.fix --protocol ESMTPS
```

Tools für Mailserver

- **etckeeper**
 - Erstellt aus /etc ein git Repository
 - Nach jeder Änderung einmal etckeeper commit
- **tig**
 - Git log viewer (zum einfachen Nachvollziehen von Änderungen)
- **Inav**
 - Moderner Log-File viewer
- **mutt**
 - Command-line Mail Client

How to start

- apt / yum / zypper install git
- git clone <https://github.com/HeinleinSupport/rspamd-slac-2019.git>

Wir suchen:

Admins, Consultants, Trainer!

Wir bieten:

Spannende Projekte, Kundenlob, eigenständige Arbeit, keine Überstunden, Teamarbeit

...und natürlich: Linux, Linux, Linux...

<http://www.heinlein-support.de/jobs>

Heinlein Support hilft bei allen Fragen rund um Linux-Server

HEINLEIN AKADEMIE

Von Profis für Profis: Wir vermitteln die oberen 10% Wissen: geballtes Wissen und umfangreiche Praxiserfahrung.

HEINLEIN HOSTING

Individuelles Business-Hosting mit perfekter Maintenance durch unsere Profis. Sicherheit und Verfügbarkeit stehen an erster Stelle.

HEINLEIN CONSULTING

Das Backup für Ihre Linux-Administration: LPIC-2-Profis lösen im CompetenceCall Notfälle, auch in SLAs mit 24/7-Verfügbarkeit.

HEINLEIN ELEMENTS

Hard- und Software-Appliances und speziell für den Serverbetrieb konzipierte Software rund ums Thema eMail.