Carsten Rosenberg <c.rosenberg@heinlein-support.de>

Automatisiertes Lernen und KI bei Anti-Spam

- **Heinlein Support**
  - IT-Consulting und 24/7 Linux-Support mit ~45 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

History of unwanted electronic messages (aka Spam)

- 1864 Complaints about unwanted Telegraph messages in the Times newspaper
  - https://blog.knowbe4.com/here-is-a-spam-message-from-1864-as-old-as-the-victorian-internet
- 1937 introduction of Spam
- 1978 first unwanted newsletter from a DEC marketer to dozens of people in the ARPANET
- 198x's usage of the term spamming in Multi User Dungeons – text based
  - https://en.wikipedia.org/wiki/Spam_(Monty_Python)

History of unwanted electronic messages (aka Spam)

- 1993 by accident 200 mails to a USENET group -> first time called spam
- 1994 commercial spam to USENET "Green Card Lottery- Final One?"
- 1997 Blocking Spam with MAPS "blackhole list"
- 1998 First DNS based RBL's
- 1999 "Happy99" worm, "Melissa" worm
- Outlook Worms
  - 2000 "Iloveyou"
  - 2001 "Anna Kournikova virus"

History of unwanted electronic messages (aka Spam)

- 2002 Paul Graham "A plan for spam" Bayesian filtering
- In January 2004 Bill Gates of Microsoft announced that "spam will soon be a thing of the past."
- 2004 first postgrey release
- 2005 Idea of phishing using ebay.com fake mails
- 2006 IronPort released a study which found 80% of spam emails originating from zombie computers.
- 2008+ Spam got more dynamically
  - Daily changing campaigns
  - Targeted phishing waves
  - But still - viagra spam

How to avoid Spam in the 90's

- Add some rules manual rules to your MTA
- Add an IP Block List

How to avoid Spam in th 200x's

- Add many rules manual rules to your MTA
  - https://www.postfixbuch.de/upload/header_checks
- Add multiple RBL's
- Add a mail content filter like spamassassin
- Add an Antivirus scanner
- Add Greylisting
- Write additional rules for spamassassin
- Train mails in SA's bayes filter

How to avoid Spam in th 201x's

- Puh, let's say its getting more complicated ...

Spamassassin

- Initially written 1997 filter.plx
- 2001 Perl Daemon Release
- Mail Content Filter with many features and plugins
- Features
  - Naive Bayes (window of 2 tokens)
  - AWL / Txrep
  - Big Ruleset
- Eloborate test-system for rule creation
- Automated system to verify scores of all rules
- Many external rulesets and plugins

Hashing tool ideas to detect Spam

- Create a generic hash of the mail and query a remote database
- Vipul's Razor
  - 2000
- DCC (Distributed Checksum Clearinghouse)
  - 2000
- Pyzor (reimplementing the Razor idea in Python)
  - 2002
- Ixhash (heise.de)
  - procmail 2003
  - later SA Plugin
  - RBL

Amavis

- Initially written 1997 as bash script to connect MTA's with AntiVirus software
- Rewritten and forked many times into the now known amavid-new 2002 by Mark Martinec
- Just includes Spamassassin (and possibly others) as Content Scanners
- Some kind IP reputation
- pen pals – replies function
- Manual reputation rules

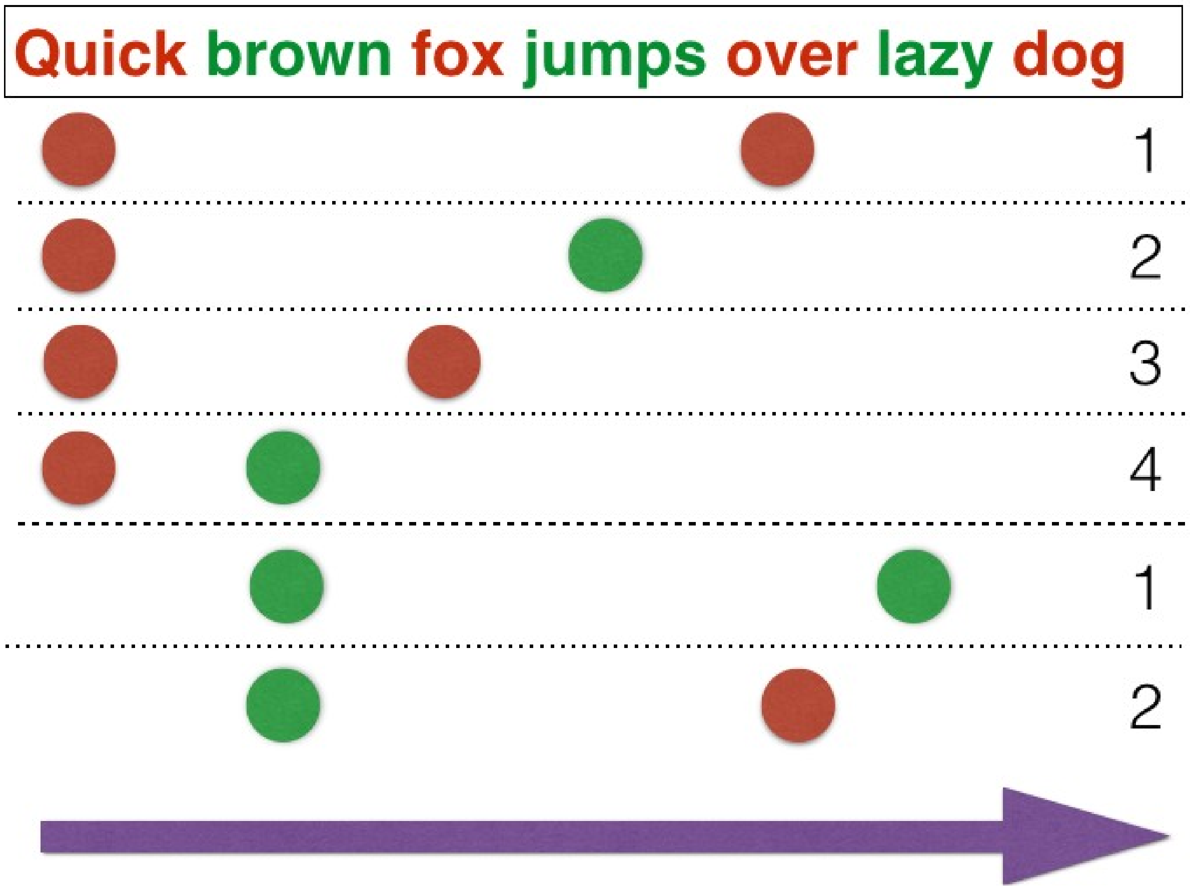
CRM114 - the Controllable Regex Mutilator

- http://crm114.sourceforge.net/
- Advanced Bayes Learning
- Markov Bayes (OSB ++)
- Spamassassin Plugin / Amavis Patches
- Released 2002 - 2010

Bayes Filtering

- https://en.wikipedia.org/wiki/Recursive_Bayesian_estimation
- Im Fall von Mail Texten
- Statistic probability of 2 words to be seen either on the spam side or on the ham side
- Also included: weighting of the distance of the 2 words in the text
- When applied to all word tuples in a text - an algorithm could calculate a propabilty the text is more likely spam or ham

Bayes Filtering in Rspamd



- Inspired by the CRM114 implementation
- Window of 5 tokens
- Also add some kind of header processing

Fuzzy Hashes in Rspamd

- Another idea of text processing / hashing – but not based on probability
- Hashes of a processed mail are added to on category (maybe ham/spam/maybe)
- Uses the shingles algorithm to process text
- Could calculate text similarity
- But final score is also based on a sum of learned weight
- Initially implemented to be used in automated spamtraps
  - A spam mail should hit the spamtrap several times to be counted as hit

- Very good detection rate !

Fuzzy Hashes in Rspamd

- Create all possible Triples of a text
- Do some hashing magic
- Compare the hashes of 2 text to see if they are equal or nearly equal

Reputation in Rspamd

- Create reputations based on a generic input key
  - IP, URL, Sender, User
  - DKIM
  - X-Mailer
- Rspamd calculates the average score of all scanned mails for an input key

- IP_REPUTATION_SPAM(2.45){asn: 48347(0.40), country: RU(0.01), ip: 193.124.117.175(0.00);}
- URL_REPUTATION(3.99){0.99940753247423;}

Other Learning methods in Rspamd

- Ratelimit
  - adaptive rates for any useable key
  - IP, Sender, User, X-Mailer
  - Spam/Ham Multiplier
    - e.g. ham mail 100 * 1.01 = 101

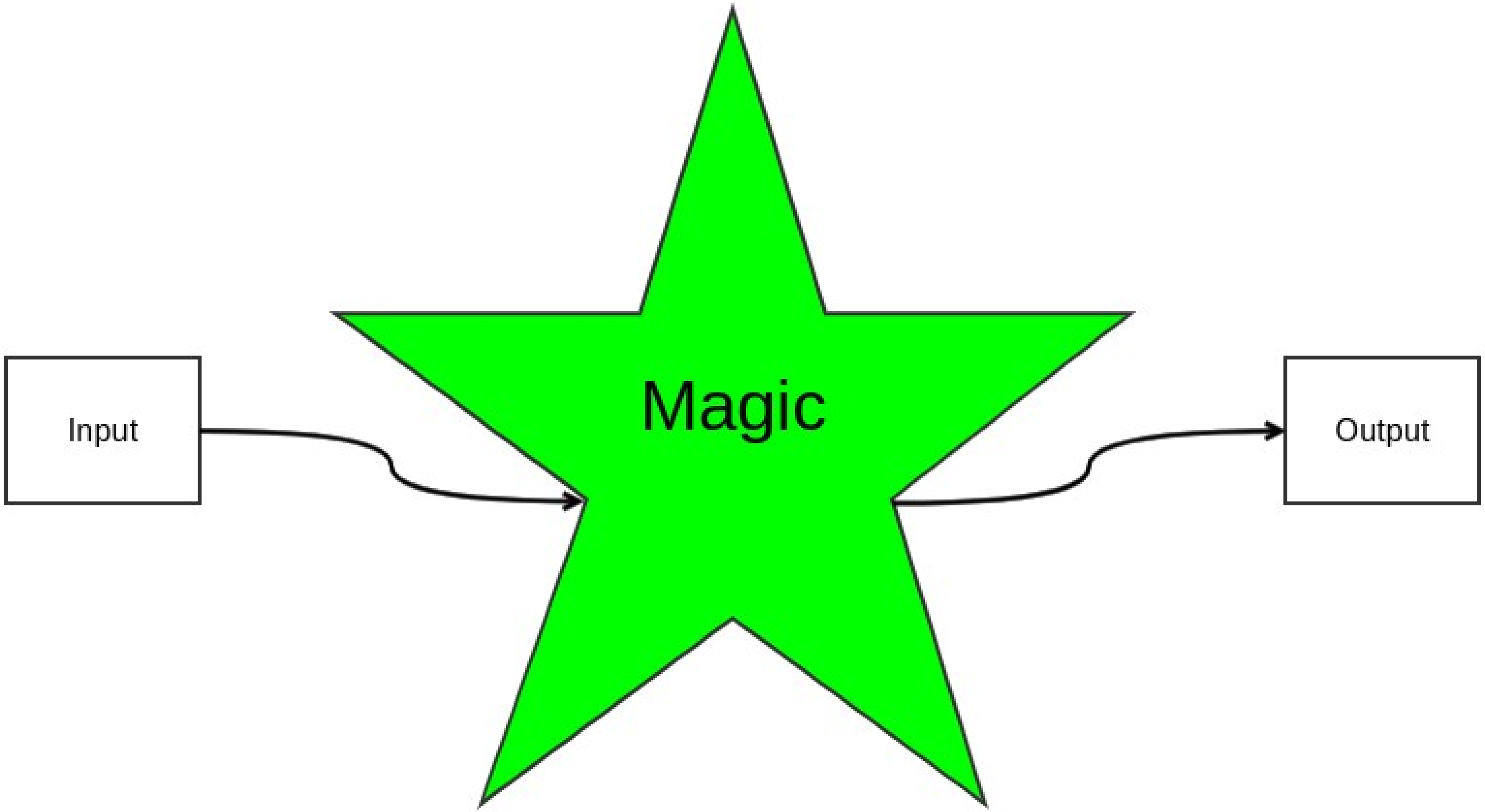
- Spamtrap
  - Learn mails send to addresses in a map directly as spam

- Neural Network

KI / AI / ML / NL Buzzwords

- As Wikipedia stated: First we need to define what intelligence in detail before we can say what artificial intelligence really is
- Artificial intelligence is known today as complex algorithms working on big pre-processed data sets to return the assumed best solution for an input
- Machine Learning: Finding generic principle in an input data set using some specific algorithm
- Neural Network: Sub-group of machine learning. The idea is inspired by biological neural networks of brains. Neurons connect to each other and having a connection weight. Often Neural networks are having multiple layers with different data transformations which outputs are combined later. E.g. a neural network could be trained to recognize pictures of cats and pictures without cats.

Neural Networks



Carsten Rosenberg <c.rosenberg@heinlein-support.de>

Neural Network in Rspamd

- Deep Learning (Multiple Layer) Neural Network based on KANN library
- Works like humans looking up thousands of distinct Spam and Ham reports to find coherences
- So not about how many times specific symbols has been seen in equal reports, but the same symbol sets in different scan reports
- The neural network plugin is collecting distinct spam and ham reports to a create a data set of the configure size
- This set will be learned using the neural network
- While using the current learned set, a new set will be collected to create a new set
- As different Rspamd configurations would make a learned set inadequate - the set ist also attached to the Config-ID (Config hash) and/or User Profile
- So running different configurations in a cluster will result in different learned neural network sets

Neural Network in Rspamd

- Many options to adjust balancing, excluding symbols, iterations, data set age
- Defaults seem to work good
- Neural data set could become invalid by config change, profile change
  - New training need to be done
- But also max_age, max_trains could invalid the data
  - New ANN set is needed - but maybe not completely collected
- Also running multiple train sets with different max_ages (and different data size) is a good idea to have better results
  - Long: 90 days / 5k samples
  - Short: 2 days / 200 samples

Neural Network in Rspamd – Real live example

- Symbols: `SPAMHAUS_ZEN(7.00), FORGED_RECIPIENTS(2.00), DMARC_POLICY_QUARANTINE(1.50), LOCAL_FUZZY_DENIED(9.31)`
  - Sum: 19,81 -> Reject
- Symbols: `FORGED_RECIPIENTS(2.00), DMARC_POLICY_QUARANTINE(1.50), LOCAL_FUZZY_DENIED(9.31)`
  - Spamhaus Symbol is missing – SUM: 12,81 – no reject :(
- Now we add the neural network
- Symbols: `SPAMHAUS_ZEN(7.00), FORGED_RECIPIENTS(2.00), DMARC_POLICY_QUARANTINE(1.50), LOCAL_FUZZY_DENIED(9.31), NEURAL_SPAM(3.00)`
  - Sum: 22,81 -> Reject
- Symbols: `FORGED_RECIPIENTS(2.00), DMARC_POLICY_QUARANTINE(1.50), LOCAL_FUZZY_DENIED(9.31), NEURAL_SPAM(3.00)`
  - Although Spamhaus is missing, the neural network is still recognizing the other symbols as typical for current spam – Sum: 15,81 → Reject :)

Problems using learning and AI in automated systems

- It’s all about your training data
- Zeit newspaper article about discriminating AI software
  - https://www.zeit.de/digital/internet/2018-05/algorithmen-rassismus-diskriminierung-daten-vorurteile-alltagsrassismus
- Face recognition Software works best for white men
  - https://www.heise.de/newsticker/meldung/Gesichtserkennung-funktioniert-am-besten-bei-weissen-Maennern-3965561.html
- https://en.wikipedia.org/wiki/Tay_(bot)
  - Microsoft’s AI self learning chat bot becames a discriminating offensive racist in hours
  - Tried 2 times :)
- First Training is often based on personal or local data

How to get good training data to start

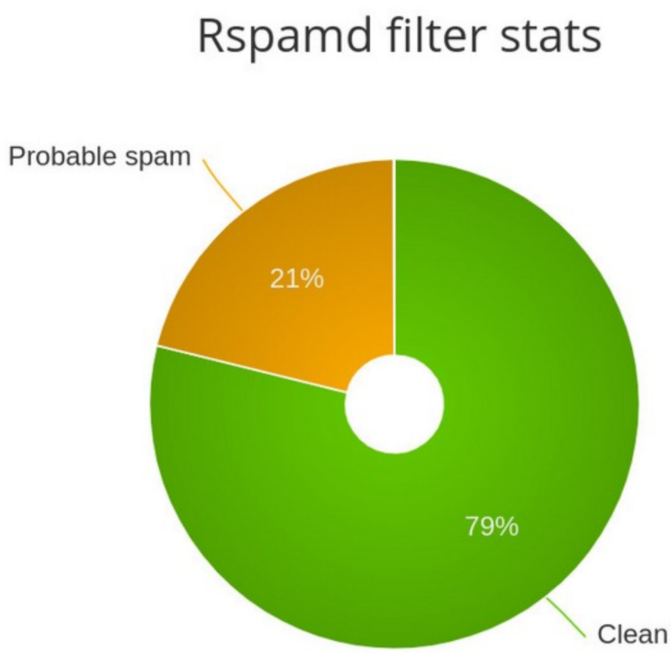
- Do not learn you personal 10k ham/spam mails residing in your Admin mailbox
  - You are the white man
- Do not learn all spams from 2005 to 2020
  - They will not represent the current spam
- Do not ...

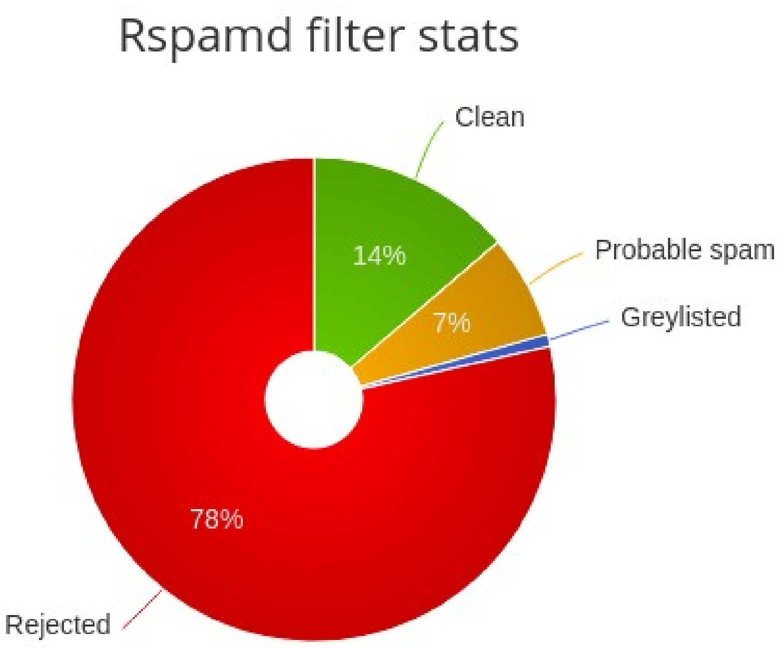
- Just activate the system on a normal Tuesday morning and let the system scan the normal traffic coming in
- Learn the current unrecognized spam mails manually

And open the gates to the learning systems

- Every mail you reject at MTA level before the learning system was able to scan it will shift your data set into a wrong direction
- Next time when the spam mail is not originating from an IP listed on a RBL – the mail is completely unknown in the learning system – so no reputation, no hash data, no neural help, maybe no reject
- Every learning system should be able to learn transparently from all incoming traffic
- This does not imply you should not reject mails listed on RBL’s anymore, but you should really consider to reject those after learning them - and beeing listed on a hight quality RBL is a quite good indication to learn a mail
- Especially local IP reputation algorithms benefit from the higher traffic

And open the gates to the learning systems





Common Problems with self learning systems

- Using high scores in rules to reject mails
  - e.g. add 20 points to really reject this type of mail
  - Could lead to learning false positives if the rule is not only matching on spam
- Run 10 years old self written rules in current content filters
- IP address with normally a good reputation is sending out spam for a short time
  - RBL hit
  - Local reputation could be inverted with some really bad mails
- Autolearning has awkward thresholds
  - Balance of learning significant ham and spam is not given anymore
- Normally a shifted recognition of one functions will be negated by other indicators
- But its also possible the bad data in the one function will push other mechanisms to learn False Positive data
- Learned data could become completely poisoned
- Attackers are trying to trick reputation systems with adding hidden text of typical mails from big tech companies (Amazon, Paypal etc.)

Solutions

- Old data should expire over the time
- e.g. Rspamd Bayes Statistics Expiry
  - Bayes Tokens are stored with a TTL in the redis database
  - The expiry script takes a look to the hit count of tokens and set a new TTL for relevant tokens
  - All other unused tokens will be removed by redis when the TTL is 0
- Go for multiple different profiles of the same type and don't add too much score for a single one
  - Bayes: Maybe different Bayes profiles for different customer types
    - While it should work technically – we had problems running German,English and Finnish speaking users in one bayes database
    - Creating one for German / English and one for Finnish worked better
  - Neural: Long, Short
  - Ratelimit: Multiple Rates
  - Reputation: IP, URL, Sender, User
  - Fuzzy: Multiple Profiles with different scores and weights

Solutions

- Do not learn Spam archives just because you found them in the internet
- Learn all local false negative mails in Bayes and Fuzzy
- When the self learning plugin still seem to decide wrong
  - Consider to wipe all learned data and have a fresh start instead of try to fix and adjust the existing data
- Consider clean rules for your policies
  - If your rule is to reject .exe files do not give this rule a score just force the reject
  - Else you're in high risk to learn FP mails when a colleague sends out the newest Firefox installer as attachment
  - Otherwise it's fine to have a score, but also reject RBL mails - as been listed on a RBL is also high spam indicator
  - Consider bypassing self learning modules for ugly/bad mails that are whitelisted by policy (example report could be: action: accept, score 30.00 / 15.00)
    - In Rspamd: prefilter, passthrough, want_spam are your friends

Crouwdsourcing – user triggered learning

- A user should not be able to influence the global mail filter with just learning one or two mails
  - Hitting the Junk button is often more comfortable than using the delete button as when deleting a mail you have to confirm it
  - So newsletter and other mails often go to the Junk folder
  - Also users don't unsubscripe from ML or newsletters and just send them to the Junk folder
  - ...
- Users could get a personal Bayes profile
- Using the Fuzzy mechanism for user based learning is a good solution as the admin could set the weight per learned mail
- Learning with a weight of 1.00 the mail needs to be trained at least 30 times to start to add any score for the hash

How self learning modules works best

- They have enough transparent data to be trained
- If they are just one extra indicator of many others
  - Basic rules, remote databases (RBL, Hash), small local adjustments and self learning modules (with multiple profiles)
- If the admin is looking for anomalies in the reports and maybe adjust settings or wipes the data of one module / profile before all data is poisoned

Soweit, so gut.

Gleich sind Sie am Zug:
Fragen und Diskussionen!

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.