www.crysys.hu

# Spam és DoS

Boldizsár Bencsáth, PhD, OSCP - Tamás Holczer, PhD
Laboratory of Cryptography and System Security
Department of Networked Systems and Services
{bencsath,holczer}@CrySyS.hu

© 2023 Laboratory of Cryptography and System Security, Budapest University of Technology and Economics

# Denial of Service Attacks

- The **Denial of Service (DoS)** is an action that prevents or impairs the authorized (normal) use of networks, systems or applications
- Typically by exhausting resources such as CPU, memory, bandwidth, and disk space
- „**Magic packet**” DoS attacks: A simple error in a software component can be used for DoS. A single packet might cause DoS.
- Always an **active attack**
- Accidental „hangs” of the system are not considered as attacks.
- **Distributed DoS** = multiple (often zombie, relay) sources used for the attack

# Unintentional DoS

- A DoS condition, when there is no attacker
- E.g. “slashdot”(.org), digg(.com) effect; special events (WTC, ticket sites, etc.)
- Television-URL effect (e.g. http://www.realgravy.co.uk/)
- From the administrator’s point of view, hard to distinguish if there is an attack or there is a software/hardware failure or just too much traffic – even detection of a DoS attack is problematic
- DoS is a problem that is very hard to handle

- In the nature, DoS is generally not a problem, multiple “servers”, physical separation
- In the world of computers, every resource is individual and thus attackable

# Examples for DoS

- Ping of death („magic packet”) / an IP packet larger than $2^{16}$ bytes by fragmentation

A malformed, single ICMP packet could freeze a computer.

Solution: Upgrading to a corrected version of the software component (IP stack in this case)

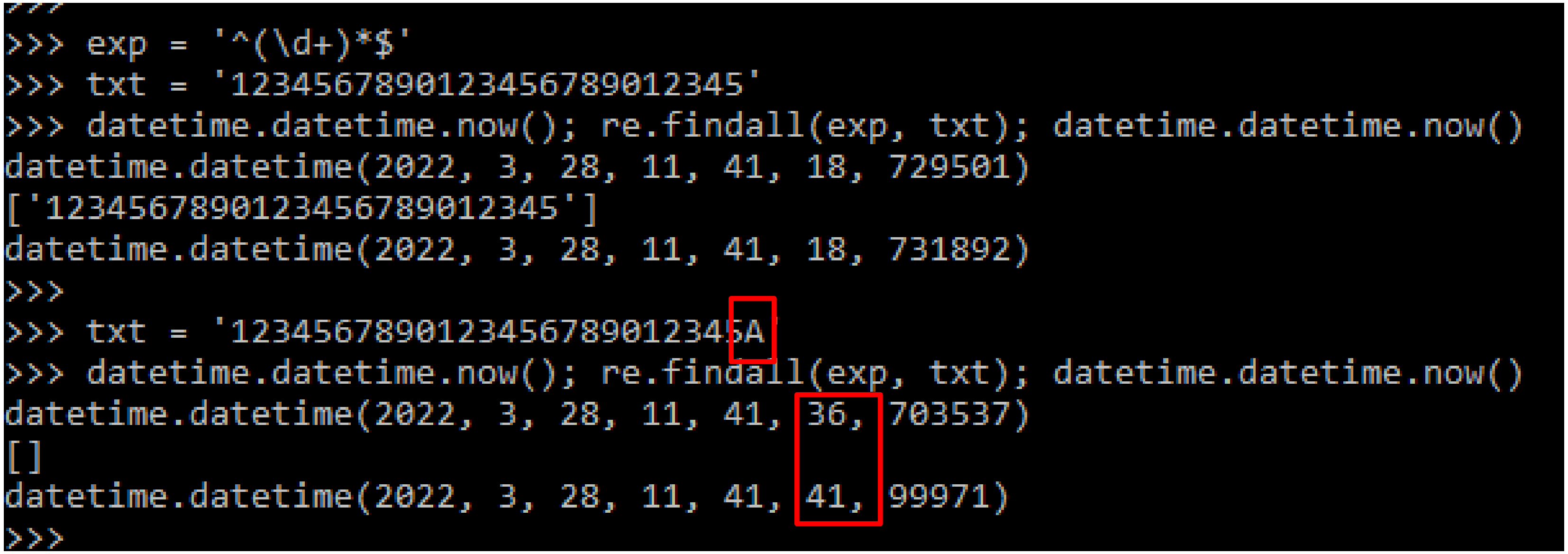
| OS | Result of PoD |
|---|---|
| Win 95 | crash |
| AIX | OS dump |
| Linux <2.0.13 | Reboot / kernel panic |
| Win NT | Mixed results |
| HPUX | crash |

- Flooding attack
- SYN Flood

# Regular expression bombs

- Regular expression: search/match expression/pattern in string
- Used for validating inputs (and others) in (web)apps

- Example: check if number (in a bad way)
- Problem: backtracking in 2^n time (repetitive in repetitive, greedy seach, all possible subset), problem in live services



```
>>>
>>> exp = '^(\d+)*$'
>>> txt = '1234567890123456789012345'
>>> datetime.datetime.now(); re.findall(exp, txt); datetime.datetime.now()
datetime.datetime(2022, 3, 28, 11, 41, 18, 729501)
['1234567890123456789012345']
datetime.datetime(2022, 3, 28, 11, 41, 18, 731892)
>>>
>>> txt = '1234567890123456789012345A'
>>> datetime.datetime.now(); re.findall(exp, txt); datetime.datetime.now()
datetime.datetime(2022, 3, 28, 11, 41, 36, 703537)
[]
datetime.datetime(2022, 3, 28, 11, 41, 41, 99971)
>>>
```

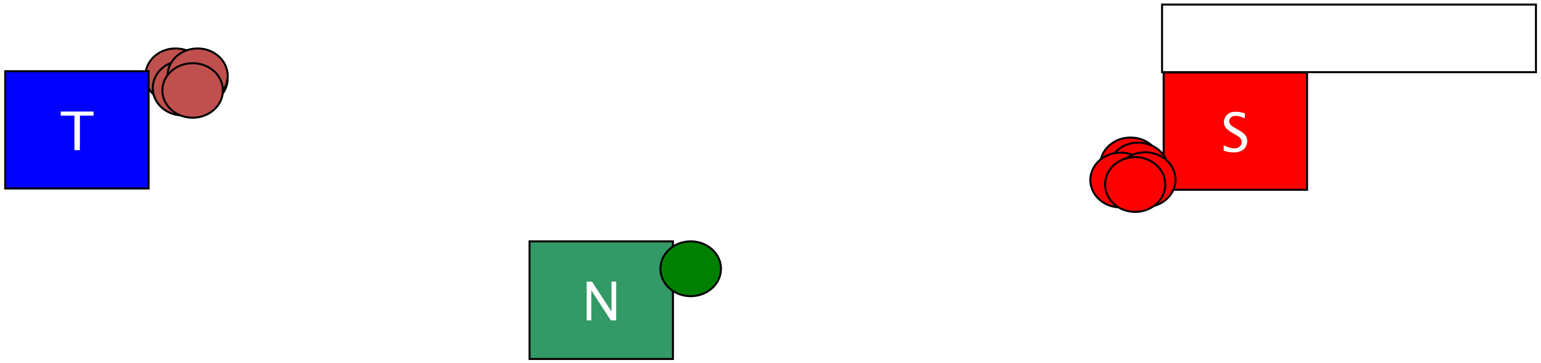
# “Normal” Client-server model

- Client-server model

- E.g. HTTP query: answer is a web page. SMTP connection: query is a mail, answer is “ok, received”
- Typically multiple steps

# DoS-Denial-of-Service

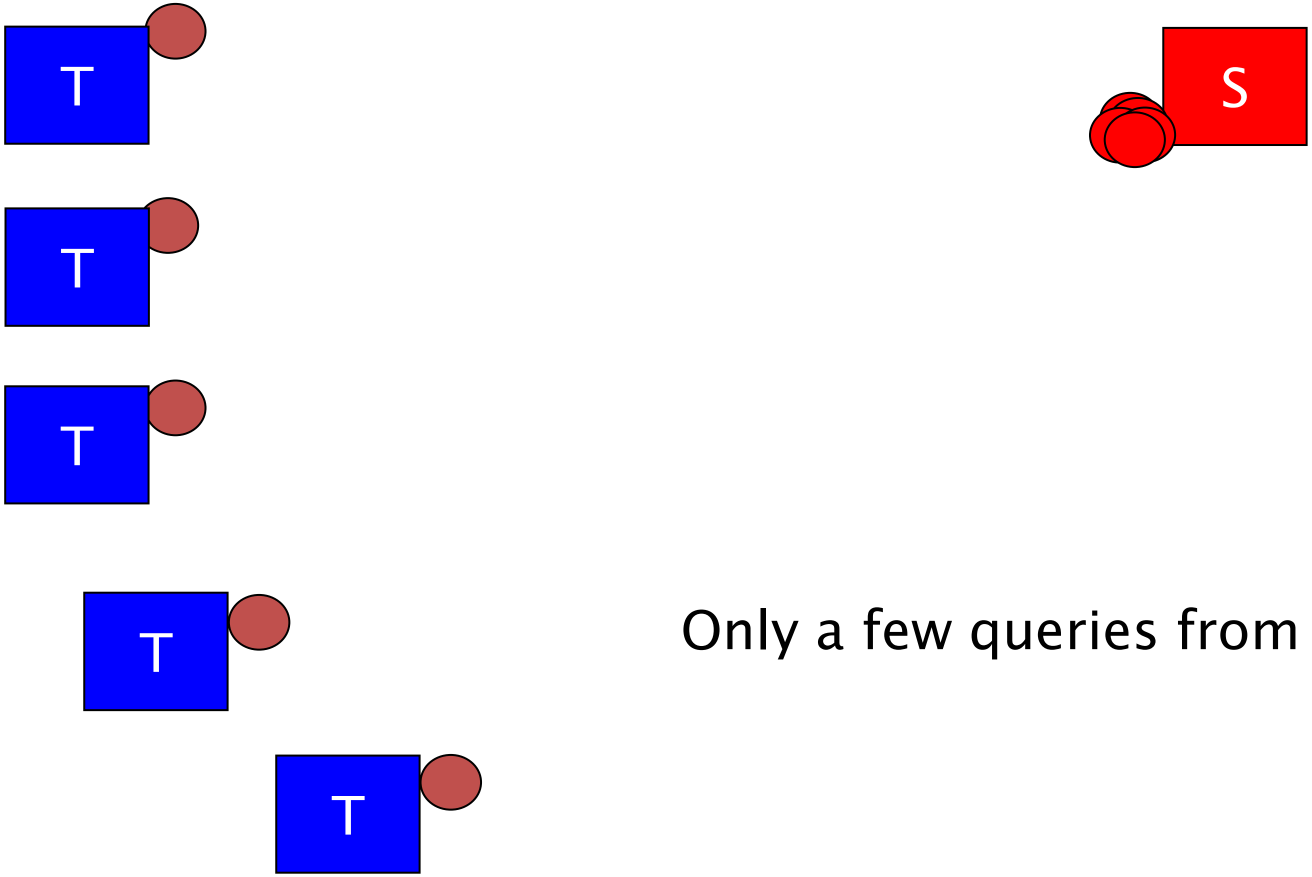
- Many queries from one source: slower answer, overloading



- Above a limit, users do not accept the service quality

# DDoS – Distributed Denial-of-Service

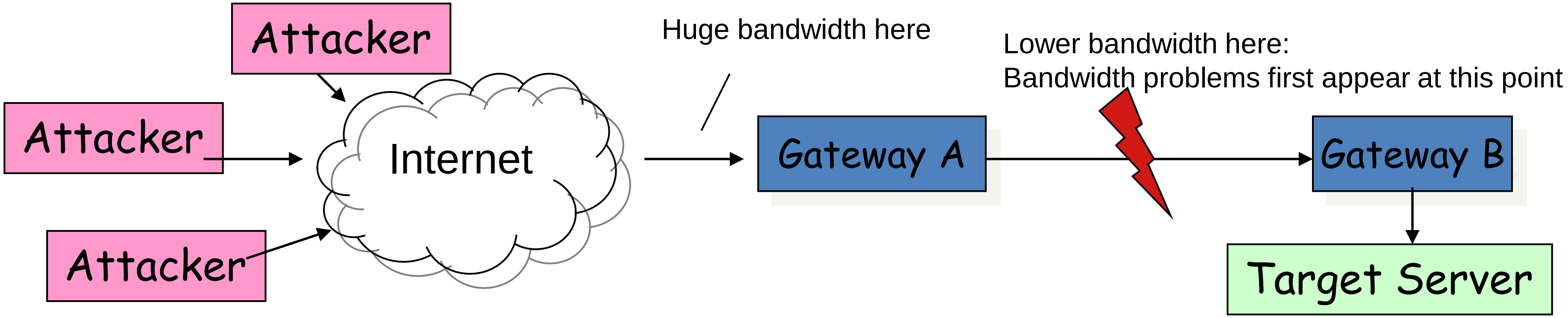
- Many attacking sources, huges traffic:DoS



Only a few queries from every host

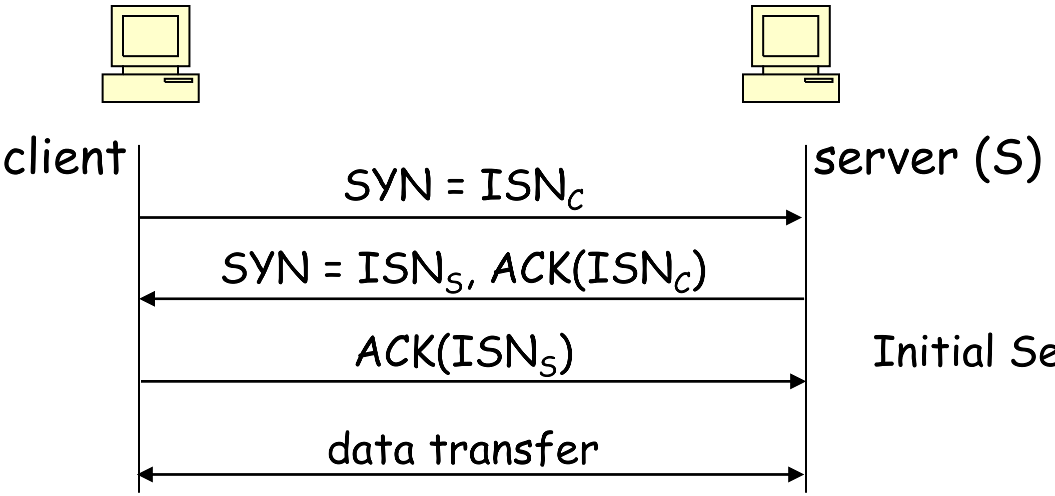
# Flooding attack

- Flooding attack
- Lot of packets (typically ICMP, UDP) are sent to a destination
- Bandwidth depletion, usually spoofed addresses and distributed (botnet, zombie network)
- A typical case is drawn below.
- Filtering, as a solution against flooding attacks:
  - **Not possible at gateway B (already in trouble)**
  - **Who manages Gateway A?**
  - **Not easy to filter out spoofed sources – Filtering might cause another DoS attack (spoofed legitimate host might result filtered legitimate clients)**
  - **Filtering is not easy if the attack is highly distributed**



# SYN flooding

TCP 3-way handshake, normal communication



client

server (S)

SYN = $ISN_C$

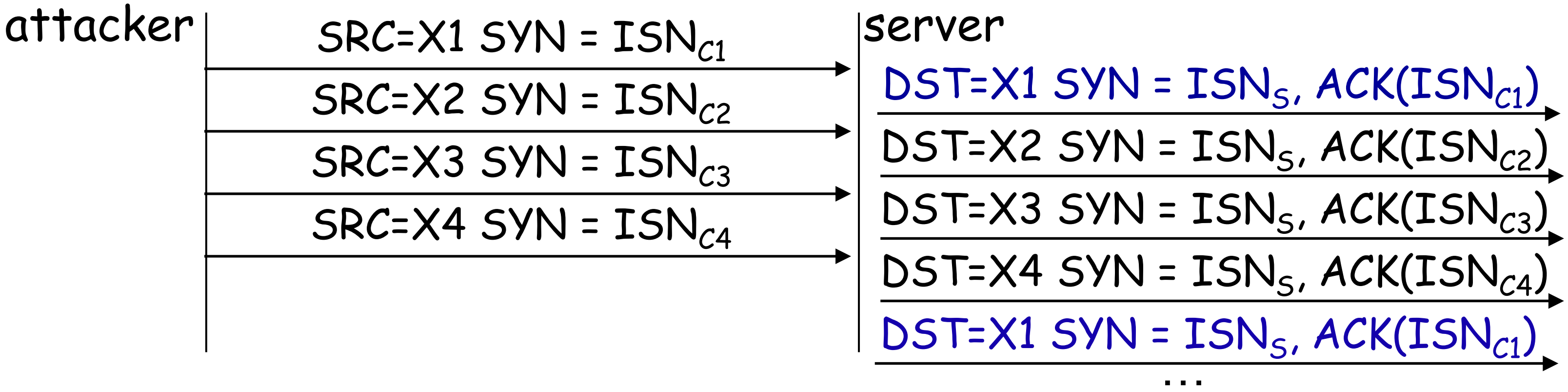
SYN = $ISN_S$, ACK($ISN_C$)

ACK($ISN_S$)

data transfer

ISN –
Initial Sequence Number

Syn flood from spoofed source



attacker

server

SRC=X1 SYN = $ISN_{C1}$

SRC=X2 SYN = $ISN_{C2}$

SRC=X3 SYN = $ISN_{C3}$

SRC=X4 SYN = $ISN_{C4}$

DST=X1 SYN = $ISN_S$, ACK($ISN_{C1}$)

DST=X2 SYN = $ISN_S$, ACK($ISN_{C2}$)

DST=X3 SYN = $ISN_S$, ACK($ISN_{C3}$)

DST=X4 SYN = $ISN_S$, ACK($ISN_{C4}$)

DST=X1 SYN = $ISN_S$, ACK($ISN_{C1}$)

…

# SYN Flood (2.)

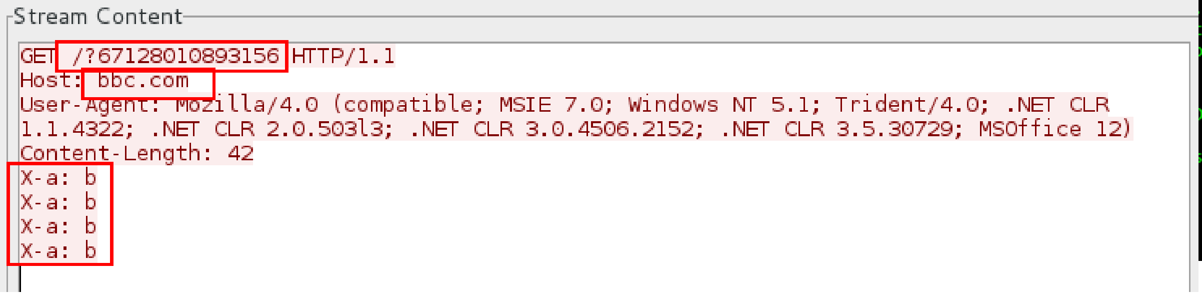
- The attacker sends thousands of SYN messages from a (possibly) spoofed source address
- The server (S) answers with SYN-ACK messages to the spoofed sources
- This depletes the bandwidth at the server (S)
- More important:
  - The server (S) should maintain the status for those half-open connections.
  - This will only be deleted after a number of resend attempts
  - The server will be loaded with half-open connetions
  - legitimate users might not be able to establish new connections. -> DoS
- **Backscatter/reflection attack**: The answers sent from the server (S) to the “spoofed” source addresses might deplete bandwidth at remote targets
- These third parties might think that the attack is originating from the server (S)
- Also possible with spoofed e-mails, ICMP packets, DNS queries, etc.
- Example: Squid web proxy: Even  if access is denied, an unauthorized connection might result in a DNS query to remote hosts

# Slowloris attack

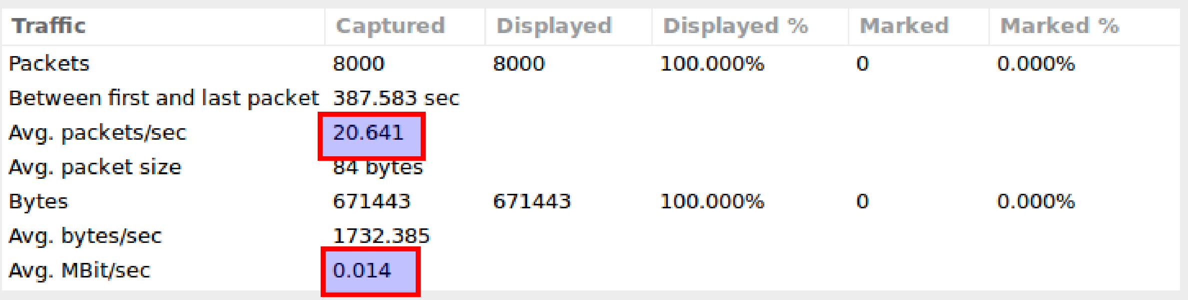
- Against web servers
- Send partial http request
- Add new headers
- Never completing the request

- Connection pool on victim is filled

```
ow bandwidth, yet greedy and poisonous HTTP client

Desktop\slowloris.pl -dns [www.example.com] -options

s\[redacted].Desktop\slowloris.pl' for help with options.
```



```
GET /?67128010893156 HTTP/1.1
Host: bbc.com
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR
1.1.4322; .NET CLR 2.0.503l3; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; MSOffice 12)
Content-Length: 42
X-a: b
X-a: b
X-a: b
X-a: b
```



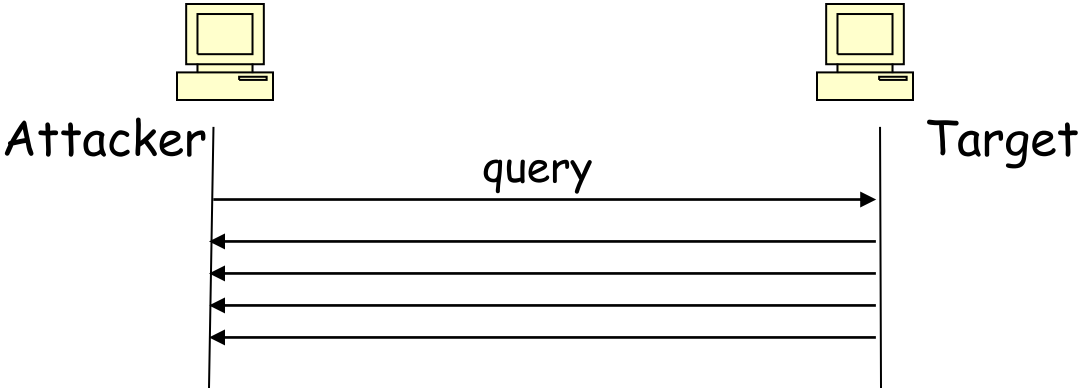
| Traffic | Captured | Displayed | Displayed % | Marked | Marked % |
|---|---|---|---|---|---|
| Packets | 8000 | 8000 | 100.000% | 0 | 0.000% |
| Between first and last packet | 387.583 sec | | | | |
| Avg. packets/sec | 20.641 | | | | |
| Avg. packet size | 84 bytes | | | | |
| Bytes | 671443 | 671443 | 100.000% | 0 | 0.000% |
| Avg. bytes/sec | 1732.385 | | | | |
| Avg. MBit/sec | 0.014 | | | | |

Mitigation:

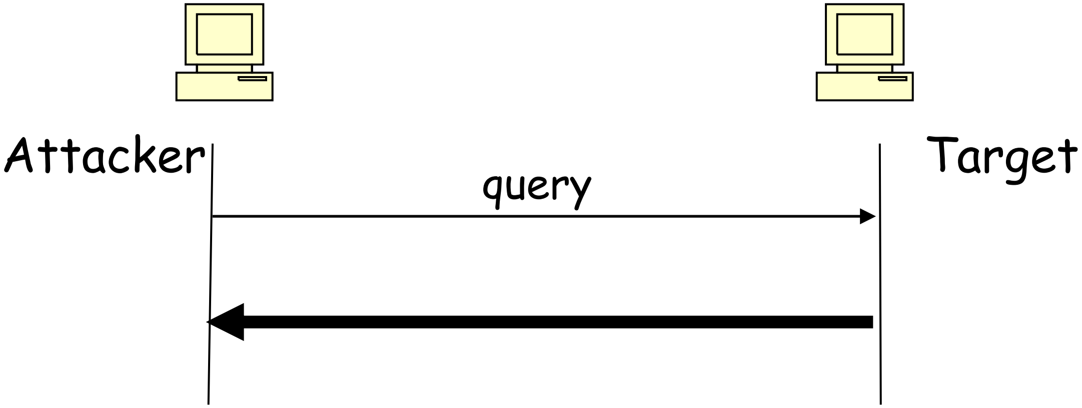
- Parameter configuration (max client, min bandwidth etc.)
- Unaffected servers (e.g. nginx)

# DoS amplification

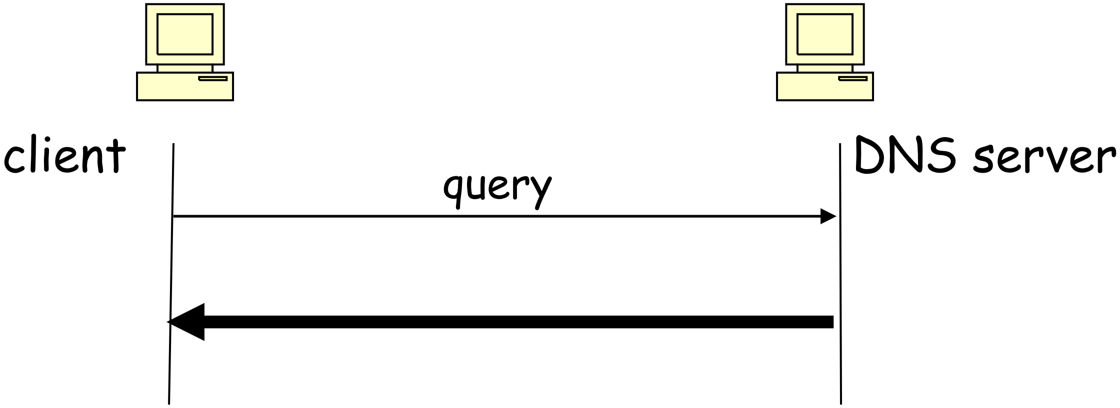
- Two basic ideas
  - Multiple replies to a single query



  - Reply is much larger than the query



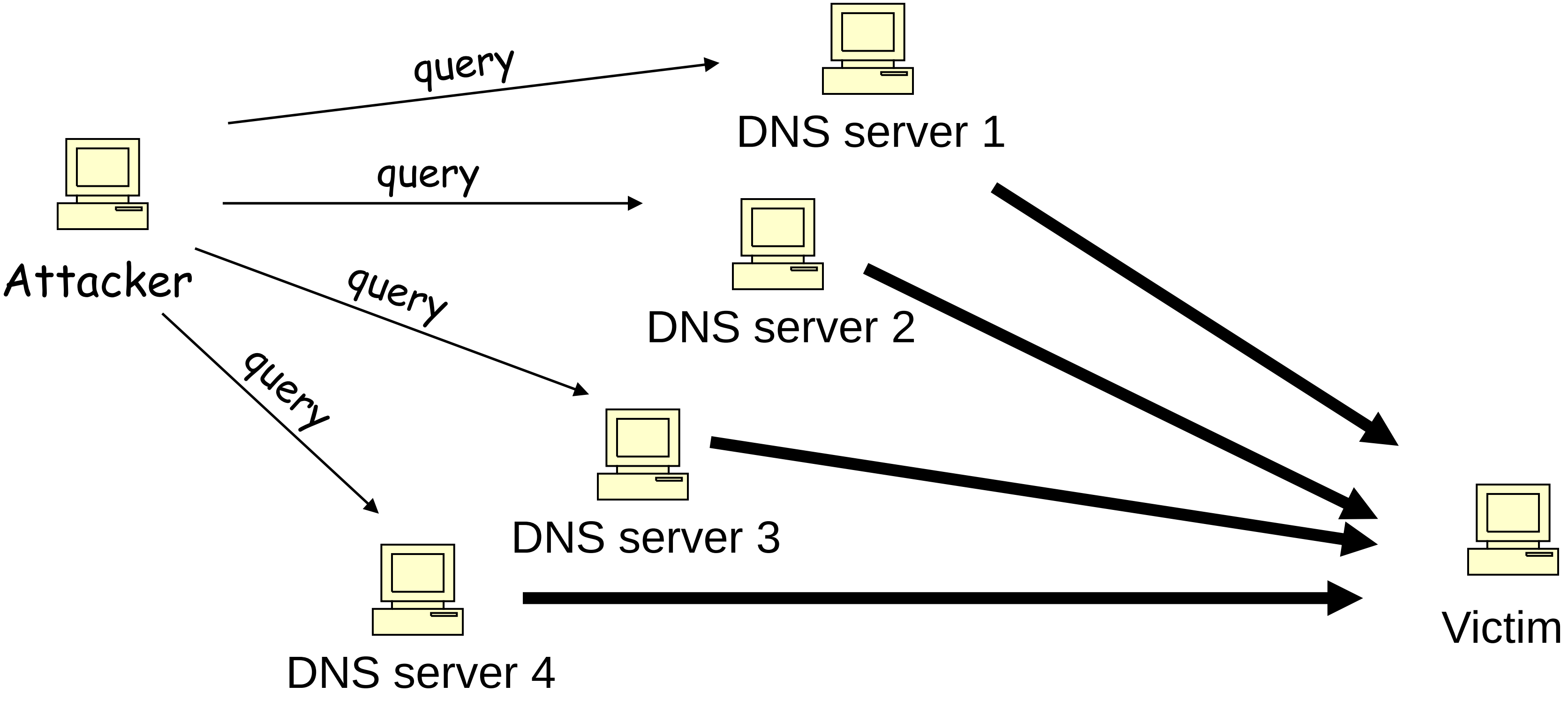
# DoS Amplification with DNS



Query: ~60 bytes
Answer: possibly 4000 bytes, e.g.
“xxx.special-domain.sthg IN TXT AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
…

Amplification: 66x-73x

# DoS - DNS amplification



- The attacker sends spoofed queries
- The servers answer to the spoofed source address, to the Victim
- The victim receives huge traffic
- Countermeasure: Do not allow recursive queries to DNS servers to anyone

(the DNS server would deny to answer the query,
sends only a small packet back to the Victim

# NTP based amplified DoS attack

- Network Time Protocol uses port 123 for data transfer

```
tcpdump -n -e -i lo port 123
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
13:56:41.939197 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 234:
127.0.0.1.55451 > 127.0.0.1.123: NTPv2, Reserved, length 192
13:56:41.939374 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 482:
127.0.0.1.123 > 127.0.0.1.55451: NTPv2, Reserved, length 440
13:56:41.939394 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 482:
127.0.0.1.123 > 127.0.0.1.55451: NTPv2, Reserved, length 440
13:56:41.939405 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 482:
127.0.0.1.123 > 127.0.0.1.55451: NTPv2, Reserved, length 440
13:56:41.939415 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 482:
127.0.0.1.123 > 127.0.0.1.55451: NTPv2, Reserved, length 440
13:56:41.939425 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 482:
127.0.0.1.123 > 127.0.0.1.55451: NTPv2, Reserved, length 440
13:56:41.939434 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 482:
127.0.0.1.123 > 127.0.0.1.55451: NTPv2, Reserved, length 440
…..
```

# NTP monlist

boldi@machine:~$ ntpdc -c monlist localhost

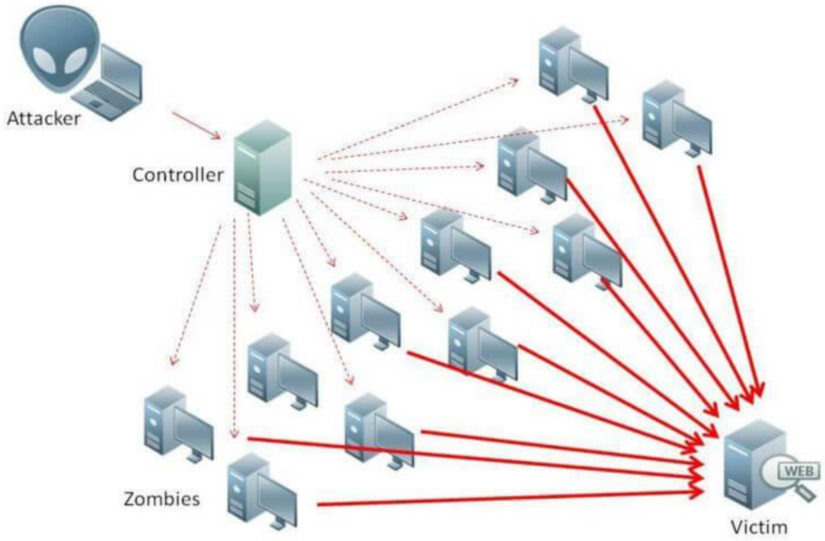
```
remote address          port local address      count m ver rstr avgint  lstint
===============================================================================
254C2813.nat.pool.tele 57709 195.228.75.149        30 3 4   1d0      32       1
business-088-079-133-1   123 195.228.75.149        35 3 4   1d0      55       1
deibp9eh1--blueice3n2.   281 195.228.75.149        16 3 4   1d0      83       1
51B65CC9.dsl.pool.tele 11533 195.228.75.149         1 3 4   1d0       1       1
atom.chemaxon.com      60098 195.228.75.149      3938 3 4   1d0      26       1
195.7.107.141            123 195.228.75.149        15 3 4   1d0     122       1
lbck-5f779388.pool.med   123 195.228.75.149         1 3 3   1d0       1       1
BC0675E7.catv.pool.tel 19949 195.228.75.149       246 3 4   1d0      14       1
sd440720c.adsl.online. 33920 195.228.75.149         7 3 4   1d0     107       1
85.255.233.229         55550 195.228.75.149         6 3 3   1d0      14       1
osaka.w5.hu              123 195.228.75.149         1 3 4   1d0       2       2
5400EABA.dsl.pool.tele   123 195.228.75.149       537 1 3   1d0      15       2
95.9.52.111.static.ttn 55356 195.228.75.149      1389 3 4   1d0      31       2
catv-89-132-133-31.cat   123 195.228.75.149      1191 1 3   1d0      16       2

boldi@machine:~$ ntpdc -c monlist 195.228.75.555
195.228.75.555: timed out, nothing received
***Request timed out
```

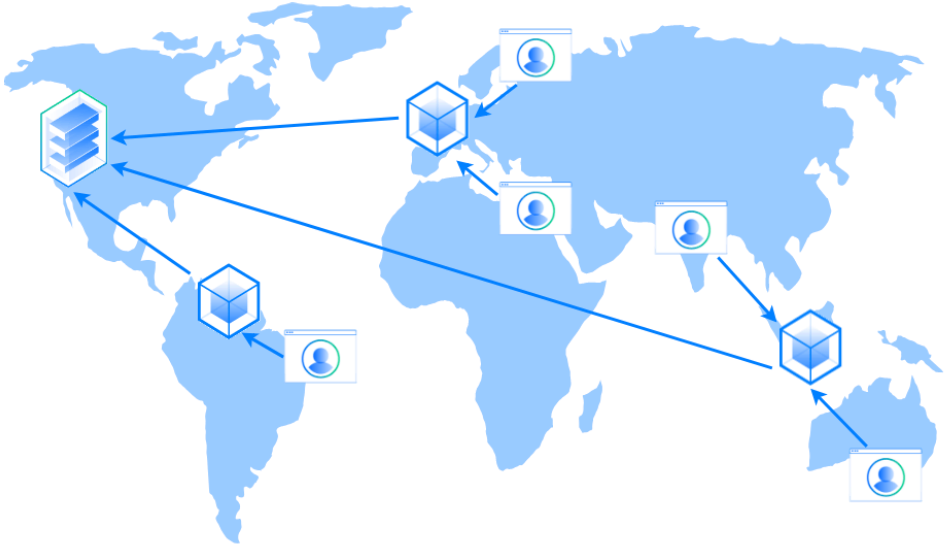
# NTP based amplified DDoS

- NTP supported “Monlist” command to receive list of Ips recently used a server
- The problem was the high amplification ratio of monlist (query: ~200 bytes, answer: ~40000 or more bytes)
- Also: monlist raises privacy concerns
- Supporting “monlist” was removed from most distributions
- Some attackers used NTP based technique 20/2014 in the wild to create some ~400 GBps attacks
- (check http://krebsonsecurity.com/2014/02/the-new-normal-200-400-gbps-ddos-attacks/ )

# Protection: Filtering / CDN

- Request can be filtered (DNS, NTP etc.), but normal HTTP?
- Copies of your content stored by CDN server all around the world
- Content is served from colsest server
- Traffic is shared between all CDN servers

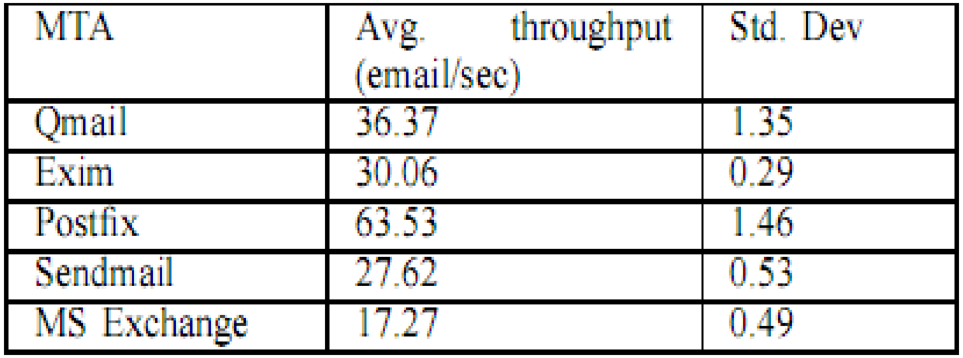




Content Delivery Network (CDN)

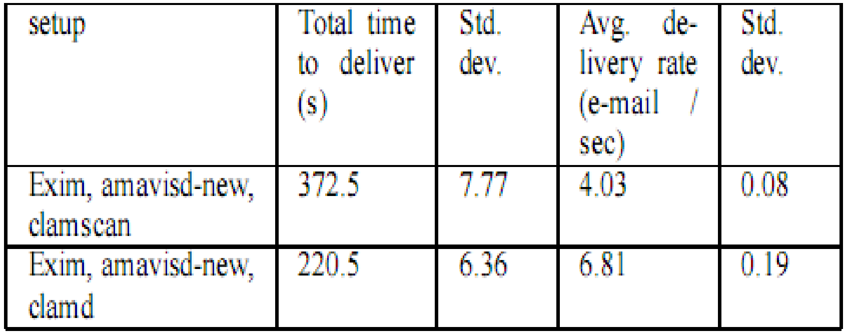
# SMTP server performance – DoS and SPAM

- Don’t think that e-mail delivery is so fast!



| MTA | Avg. throughput (email/sec) | Std. Dev |
|---|---|---|
| Qmail | 36.37 | 1.35 |
| Exim | 30.06 | 0.29 |
| Postfix | 63.53 | 1.46 |
| Sendmail | 27.62 | 0.53 |
| MS Exchange | 17.27 | 0.49 |

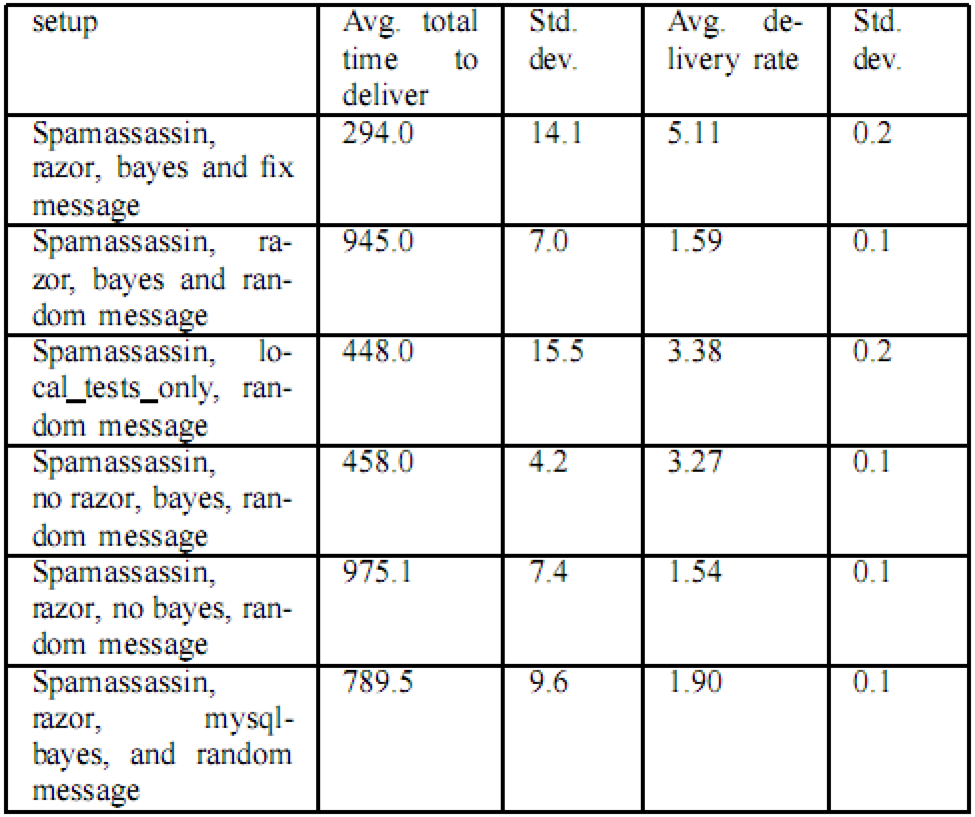
- With virus scanning



| setup | Total time to deliver (s) | Std. dev. | Avg. delivery rate (e-mail / sec) | Std. dev. |
|---|---|---|---|---|
| Exim, amavisd-new, clamscan | 372.5 | 7.77 | 4.03 | 0.08 |
| Exim, amavisd-new, clamd | 220.5 | 6.36 | 6.81 | 0.19 |

# Spam filtering is slow

- Speed with spam filtering



| setup | Avg. total time to deliver | Std. dev. | Avg. delivery rate | Std. dev. |
|---|---|---|---|---|
| Spamassassin, razor, bayes and fix message | 294.0 | 14.1 | 5.11 | 0.2 |
| Spamassassin, razor, bayes and random message | 945.0 | 7.0 | 1.59 | 0.1 |
| Spamassassin, local_tests_only, random message | 448.0 | 15.5 | 3.38 | 0.2 |
| Spamassassin, no razor, bayes, random message | 458.0 | 4.2 | 3.27 | 0.1 |
| Spamassassin, razor, no bayes, random message | 975.1 | 7.4 | 1.54 | 0.1 |
| Spamassassin, razor, mysql-bayes, and random message | 789.5 | 9.6 | 1.90 | 0.1 |

# That means: very vulnerable

Test messages: body size of 4kb, random text

Exim, virus+spam filtering = 1.58 e-mails/sec

1.58e-mails/sec*4kb=6.32 kb/sec payload, ~50kbps, with overhead ~64kbps

Using only 64kbps we can overload an SMTP server with content filtering!

# SPAM

- **Spam** is the abuse of electronic messaging systems to send unsolicited bulk messages
- Spamming remains economically viable because advertisers have no operating costs beyond the management of their mailing lists, and it is difficult to hold senders accountable for their mass mailings.
- **Ham**: a message that is not spam
- **Filtering**: decision if ham or spam (with some confidence)
  - False positive/negative decision
- **Spammer**: send spam for profit
  - Mass operation
  - Not targeted (not interested in losing some emails)

# The first e-mail SPAM - history

Mail-from: DEC-MARLBORO rcvd at 3-May-78 0955-PDT
Date: 1 May 1978 1233-EDT
From: THUERK at DEC-MARLBORO
Subject: ADRIAN@SRI-KL
To: DDAY at SRI-KL, DAY at SRI-KL, DEBOER at UCLA-CCN,
…
ZOSEL@LLL-COMP
DIGITAL WILL BE GIVING A PRODUCT PRESENTATION OF THE NEWEST MEMBERS OF THE
DECSYSTEM-20 FAMILY; THE DECSYSTEM-2020, 2020T, 2060, AND 2060T. THE
DECSYSTEM-20 FAMILY OF COMPUTERS HAS EVOLVED FROM THE TENEX OPERATING SYSTEM
AND THE DECSYSTEM-10 <PDP-10> COMPUTER ARCHITECTURE. BOTH THE DECSYSTEM-2060T
AND 2020T OFFER FULL ARPANET SUPPORT UNDER THE TOPS-20 OPERATING SYSTEM.
THE DECSYSTEM-2060 IS AN UPWARD EXTENSION OF THE CURRENT DECSYSTEM 2040
AND 2050 FAMILY. THE DECSYSTEM-2020 IS A NEW LOW END MEMBER OF THE
DECSYSTEM-20 FAMILY AND FULLY SOFTWARE COMPATIBLE WITH ALL OF THE OTHER
DECSYSTEM-20 MODELS.

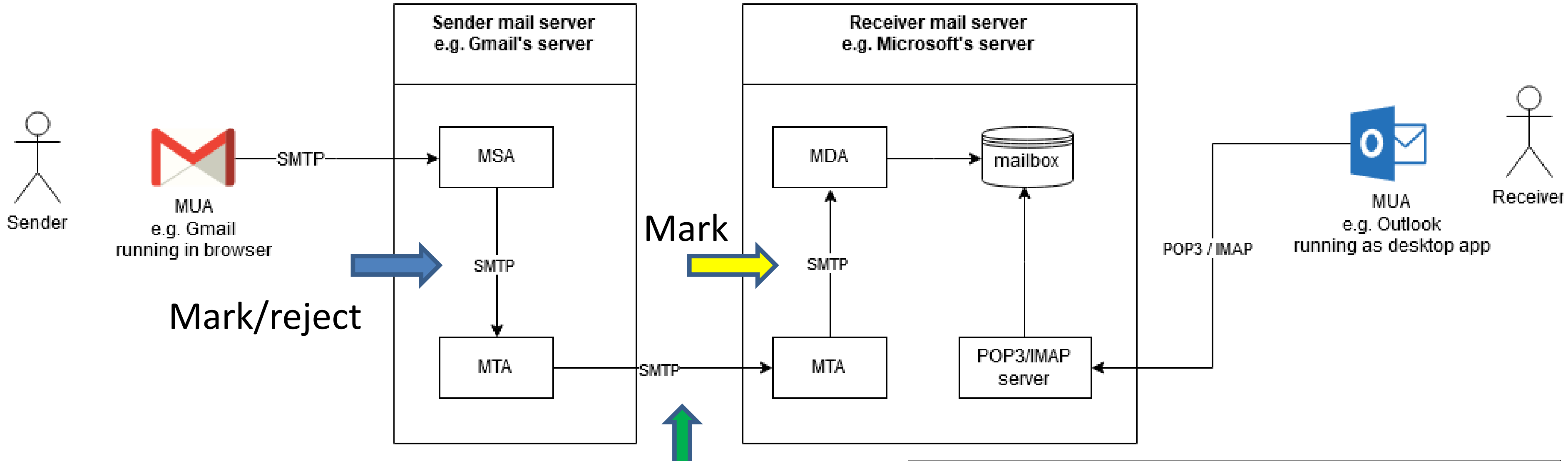
WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY
AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS
MONTH. THE LOCATIONS WILL BE:

TUESDAY, MAY 9, 1978 - 2 PM
HYATT HOUSE (NEAR THE L.A. AIRPORT)
LOS ANGELES, CA

THURSDAY, MAY 11, 1978 - 2 PM
DUNFEY'S ROYAL COACH
SAN MATEO, CA
(4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)

A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER
DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND,
PLEASE FEEL FREE TO CONTACT THE NEAREST DEC OFFICE
FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY.

# Email basics – When to filter

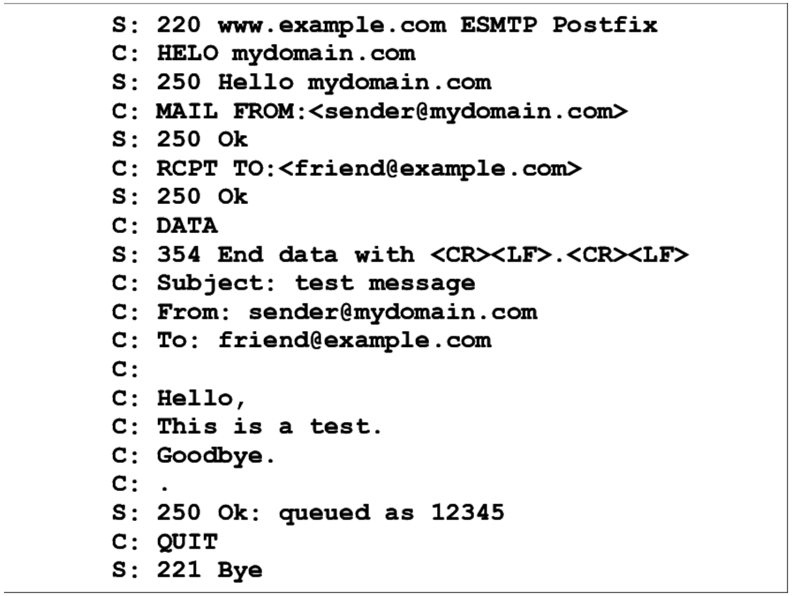


Mark

Mark/reject

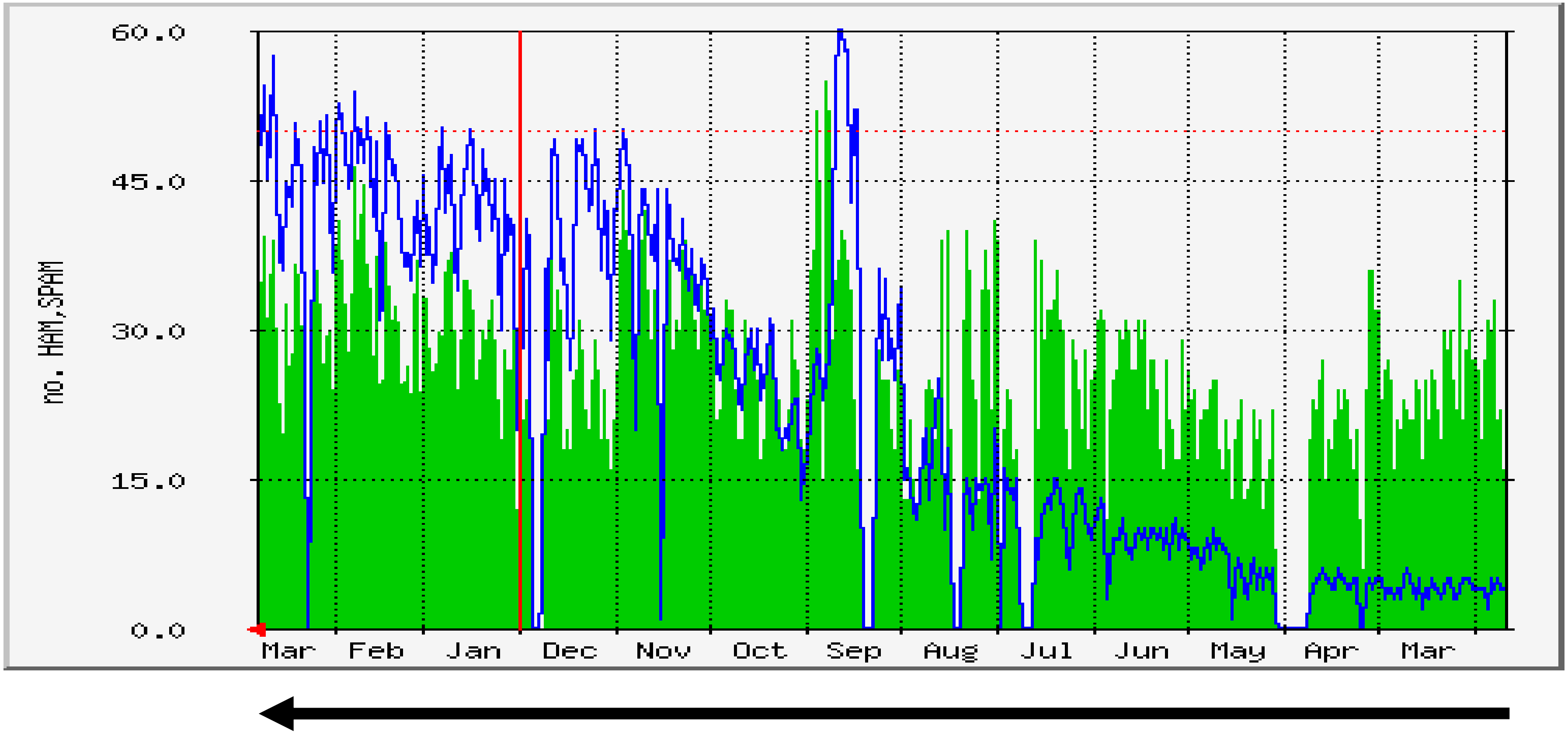
Mark/reject

- MSA - Mail submission agent
- MTA - Mail transfer agent
- MDA - Mail delivery agent
- MUA - Mail user agent
- TLS/STARTTLS usage
- Backscatter
- Secondary MX



```
S: 220 www.example.com ESMTP Postfix
C: HELO mydomain.com
S: 250 Hello mydomain.com
C: MAIL FROM:<sender@mydomain.com>
S: 250 Ok
C: RCPT TO:<friend@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: test message
C: From: sender@mydomain.com
C: To: friend@example.com
C:
C: Hello,
C: This is a test.
C: Goodbye.
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
```

# SPAM - importance



2007 idő 2006

Kék: SPAM Zöld: Normál levelek száma

# Full header of an email

```
Return-path: <nlaw518@gmail.com>
Envelope-to: boldi@crysys.hu
Delivery-date: Wed, 11 Mar 2009 08:18:14 +0100
X-Spam-Flag: YES
X-Spam-Score: 72.787
X-Spam-Level: ************************************************************************
X-Spam-Status: Yes, score=72.787 tagged_above=0.1 required=6.3 tests=[AWL=8.853, BAYES_99=10, DCC_CHECK=2.17,
DRUGS_ANXIETY=0.343, DRUGS_ANXIETY_EREC=0.001, DRUGS_ANXIETY_OBFU=0.155, DRUGS_DIET=0.001,
DRUGS_DIET_OBFU=0, DRUGS_ERECTILE=2.2, DRUGS_ERECTILE_OBFU=1.229, DRUGS_MANYKINDS=0.13,
 DRUGS_SLEEP_EREC=1.09, FB_CIALIS_LEO3=1.441, FB_MED1CAT=1, FRT_DISCOUNT=1.81, FRT_VALIUM1=1.59,
FRT_VALIUM2=1.301, FRT_WEIGHT2=2.121, FUZZY_AMBIEN=1.026, FUZZY_CPILL=0.001, FUZZY_MEDICATION=2.717,
FUZZY_MERIDIA=2.374, FUZZY_VLIUM=0.001, OBFU_1=0.5, OBFU_BAYES=5, RCVD_IN_BL_SPAMCOP_NET=1.96,
RCVD_IN_NOMOREFUNN=1.3, SARE_OBFU_CODEINE=0.833, SARE_OBFU_MEDS=2.777, SARE_OBFU_PART_IUM=0.978,
 SARE_OBFU_PHARM=2.222, SARE_OBFU_PHARM_POX=1.666, SARE_OBFU_VALIUM=1.666, SARE_OBFU_XANAX=2.222,
SARE_SUB_MEDS_LEO=2.222, SPF_NEUTRAL=0.686, TVD_VISIT_PHARMA=0.001, URIBL_BLACK=4.2,
URIBL_JP_SURBL=1.501, URIBL_SBL=1.499]
..
Received: from shamir.crysys.hit.bme.hu ([10.105.1.254]) by localhost (ss.crysys.hu [10.105.1.55]) (amavisd-new, port 10023) with
 ESMTP id w4x56cZmEL2r; Wed, 11 Mar 2009 08:18:11 +0100 (CET)
Received: from 80-218-100-154.dclient.hispeed.ch ([80.218.100.154]) by shamir.crysys.hit.bme.hu with smtp (Exim 4.63)
 (envelope-from <nlaw518@gmail.com>) id 1LhIhd-0001V8-Ti; Wed, 11 Mar 2009 08:18:10 +0100
From: "Trinidad Pickett" <boldi@crysys.hit.bme.hu>
To: "Shelly Bullock" <boldi@crysys.hit.bme.hu>
Message-ID: <SXCcunj52fmunmboldi@crysys.hit.bme.hu>
Content-Type: text/plain;
Content-Transfer-Encoding: 7Bit
Date: Wed, 11 Mar 2009 00:17:56 -0800
Subject: ***SPAM*** ***SPAM*** The only med1cation for we1ght l0ss that does work
```

# SPAM filtering

- From filter rules to heuristic, “scoring” methods – decision after multiple tests.
- Discarding errorous e-mails and connections (sometimes direct filtering, no additional tests):
    - Missing (mandatory) “Date:” field in header, missing FQDN after HELO in SMTP connection, bad reverse-DNS for the host, etc.

Special rules for most common spams

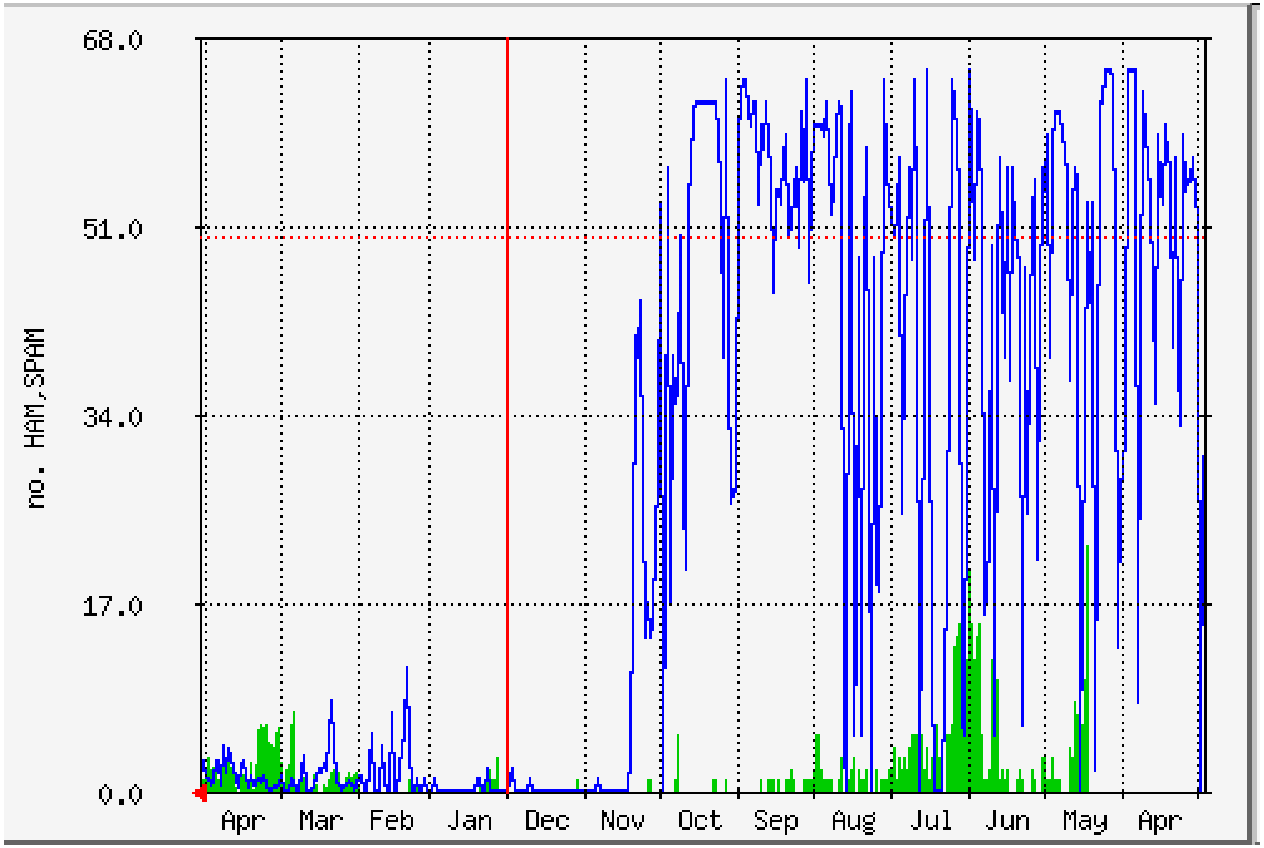
Filtering words like “VIAGRA”, identifying obfuscations

# Header checks

- Local part
  - If the recipient exist (user, alias, db, ad, ldap…)
  - What if it is down? (can result in permanent reject which is a problem)
- Domain part
  - Is it ours?
- Sender
  - Check if exists (remote call)
- Envelope syntax
  - Missing mandatory fields?
  - Proper HELO?
- Header syntax
  - E.g. missing date
  - But normal clients can also create bad header
- DNS
  - Reverse dns exist? Dynamic?
- …

# Effect of some tools can achieve high gains

- Introduction of mandatory reverse DNS, proper HELO and greylisting



# Statistical filtering: The bayesian method

- Bayes' theorem $$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

- In case of spam:

$$\Pr(S|W) = \frac{\Pr(W|S).\Pr(S)}{\Pr(W|S).\Pr(S) + \Pr(W|H).\Pr(H)}$$

W: a word is in the email e.g. “Viagra”

S: the email is spam

H: the email is ham (good message, not spam)

Pr(S|W): The message is spam, if it contains the word W.

Pr(W|S): In a spam message, the probability of the existance of the particular word

etc.

# Statistical filtering: The bayesian method 2.

- About 80% of the internet e-mails are spam

$$\Pr(S) = 0.8; \Pr(H) = 0.2$$

- However, many bayesian spam filter make the assumption

$$\Pr(S) = 0.5; \Pr(H) = 0.5$$

- In this case

$$\Pr(S|W) = \frac{\Pr(W|S)}{\Pr(W|S) + \Pr(W|H)}$$

# Combining individual probabilities

- We can assume that the appearance of individual words are independent event.(this is generally not true, but still, we can assume that)
- In this case:

$$p = \frac{p1.p2.\cdots.pN}{p1.p2.\cdots.pN + (1-p1).(1-p2).\cdots.(1-pN)}$$

- p: probability that the suspect message is spam
- p1:Pr(S|W$_1$), p2: Pr(S|W$_2$)

# Bayesian filtering

- We collect statistics about individual words in spam and ham messages into a database
- During filtering, retrieve Pr(S|W) Pr(H|W) for every word in the email
- Calculate the probability of the event that the e-mail is spam
- A separate database can be used for every user (different e-mails, different statistics)
- Spammers can attempt to decrease effectiveness:
  - Adding common words to the e-mail
  - Poisioning the database

- Bayesian filtering is one of the most usable methods currently

# RBL (Real-Time Blacklist)

- **RBL:** Originally a list that contained the blacklisted SMTP servers
- Now: dozens of RBLs available, from different organizations and providing different information
- Some specialties:

  -Computers with dialup IP address (DUL) (can be submitter!)

  -RFC ignorant hosts

  -URIBL: Blacklisted URIs
- **DNSBL:** Most of the RBLs use DNS to communicate.
- Advantage: DNS is a distributed service, caching is possible, easy to transfer through firewalls, easy to implement
- Example:

**"dig lowlyenjoy.com.multi.uribl.com in a":**

**Answer "lowlyenjoy.com.multi.uribl.com. 1762 IN A 127.0.0.2"**

**Understanding: 127.0.0.2 means "black", URI (URL) used by spammer.**

**([http://www.uribl.com/about.shtml](http://www.uribl.com/about.shtml))**

# Other techniques against spam

- Greylisting
  - (From, To, IP address)
  - Do not let the first ‘trial’ (temporary reject)
  - Most spammers don’t try twice
- Authenticating senders (SPF, DKIM)
  - The sender proves that the message is not spoofed
  - SPF: The domain DNS record contains valid SMTP servers (as sending host)
    - Lot of problems, e.g. forwarding
  - DKIM: The DNS record contains public key to check signature on email.
    - The signature is generally put by the mail server
  - Wide deployment would be crucial

# SPF: Sender Policy Framework

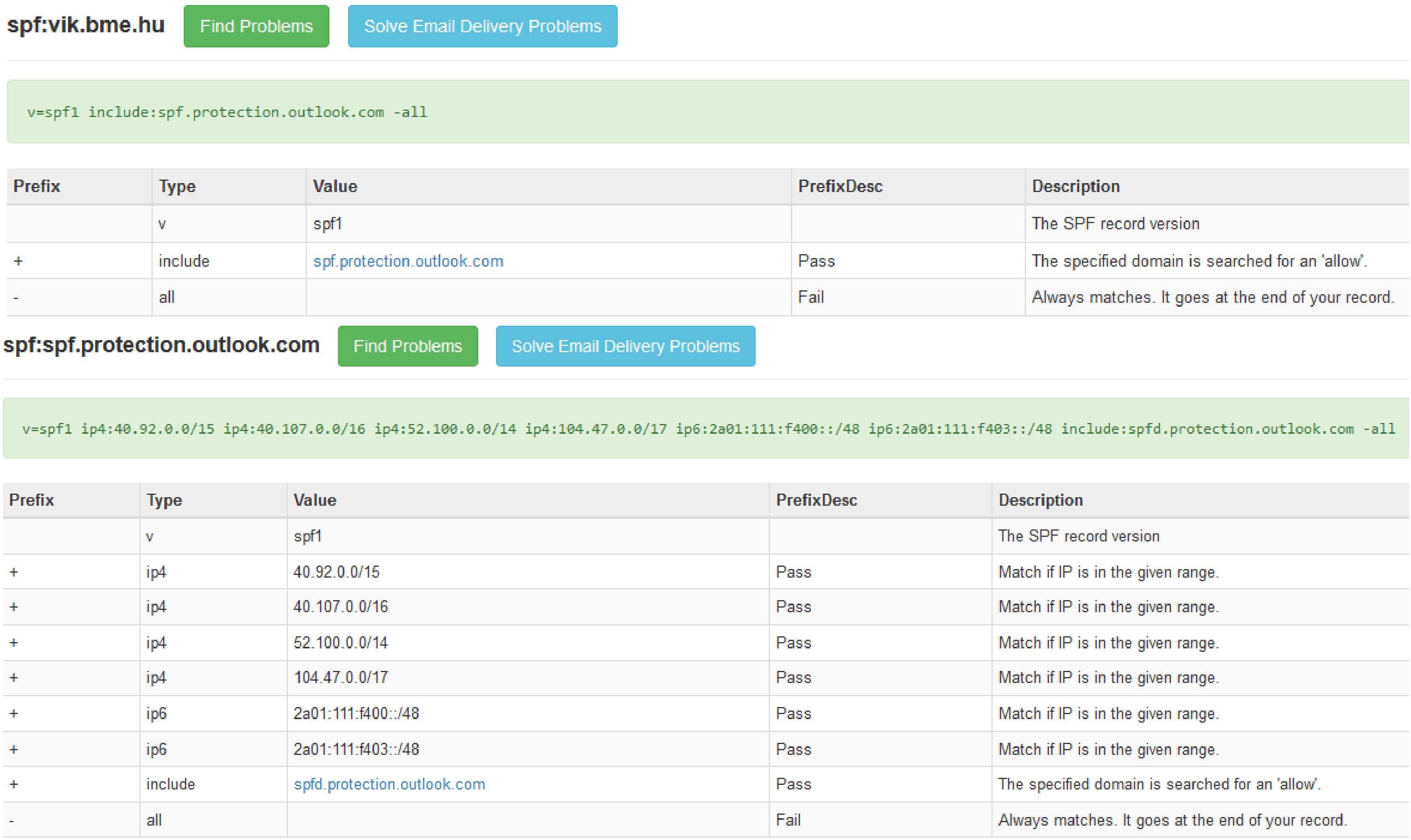
- RFC4408
- A DNS record is created like this and can be used to validate if a server is intended to send out mails for the given domain::
- $dig dc.hu in any|grep spf
- dc.hu. 5000 IN TXT **"v=spf1 a mx ip4:152.66.249.135 ip4:195.228.45.175 -all"**
- For domains that do not send any mail: **„v=spf1 -all”**
- ALL Matches always; used for a default result like -all for all IPs not matched by prior mechanisms.
- A If the domain name has an address record (A or AAAA) that can be resolved to the sender's address, it will match.
- IP4 If the sender is in a given IPv4 address range, match.
- IP6 If the sender is in a given IPv6 address range, match.
- MX If the domain name has an MX record resolving to the sender's address, it will match (i.e. the mail comes from one of the domain's mail servers).
- PTR If the domain name (PTR record) for the client's address is in the given domain and that domain name resolves to the client's address (forward-confirmed reverse DNS), match.
- EXISTS If the given domain name resolves to any address, match (no matter the address it resolves to). This is rarely used. Along with the SPF macro language it offers more complex matches like DNSBL-queries.
- INCLUDE If the included (a misnomer) policy passes the test this mechanism matches. This is typically used to include policies of more than one ISP.

# SPF qualifiers

- + for a PASS result. This can be omitted; e.g., +mx is the same as mx.
- ? for a NEUTRAL result interpreted like NONE (no policy).
- ~ for SOFTFAIL, a debugging aid between NEUTRAL and FAIL. Typically, messages that return a SOFTFAIL are accepted but tagged.
- - for FAIL, the mail should be rejected (see below).

- SPF has a number of controversies and problems
- In a survey published in 2007, 5% of the .com and .net domains had some kind of SPF policy.
- 2021 Alexa Top million: 63.4% exist but only 52.9% valid
- Problems with bad configuration
  - Generating SPF record
  - Subdomains not covered generally (2021: only 1% uses SPF for subdomain)
  - Sending company mail from home

# SPF examples



spf:vik.bme.hu
Find Problems
Solve Email Delivery Problems

```
v=spf1 include:spf.protection.outlook.com -all
```

| Prefix | Type | Value | PrefixDesc | Description |
|---|---|---|---|---|
| | v | spf1 | | The SPF record version |
| + | include | spf.protection.outlook.com | Pass | The specified domain is searched for an 'allow'. |
| - | all | | Fail | Always matches. It goes at the end of your record. |

spf:spf.protection.outlook.com
Find Problems
Solve Email Delivery Problems

```
v=spf1 ip4:40.92.0.0/15 ip4:40.107.0.0/16 ip4:52.100.0.0/14 ip4:104.47.0.0/17 ip6:2a01:111:f400::/48 ip6:2a01:111:f403::/48 include:spfd.protection.outlook.com -all
```

| Prefix | Type | Value | PrefixDesc | Description |
|---|---|---|---|---|
| | v | spf1 | | The SPF record version |
| + | ip4 | 40.92.0.0/15 | Pass | Match if IP is in the given range. |
| + | ip4 | 40.107.0.0/16 | Pass | Match if IP is in the given range. |
| + | ip4 | 52.100.0.0/14 | Pass | Match if IP is in the given range. |
| + | ip4 | 104.47.0.0/17 | Pass | Match if IP is in the given range. |
| + | ip6 | 2a01:111:f400::/48 | Pass | Match if IP is in the given range. |
| + | ip6 | 2a01:111:f403::/48 | Pass | Match if IP is in the given range. |
| + | include | spfd.protection.outlook.com | Pass | The specified domain is searched for an 'allow'. |
| - | all | | Fail | Always matches. It goes at the end of your record. |

# DCC #1

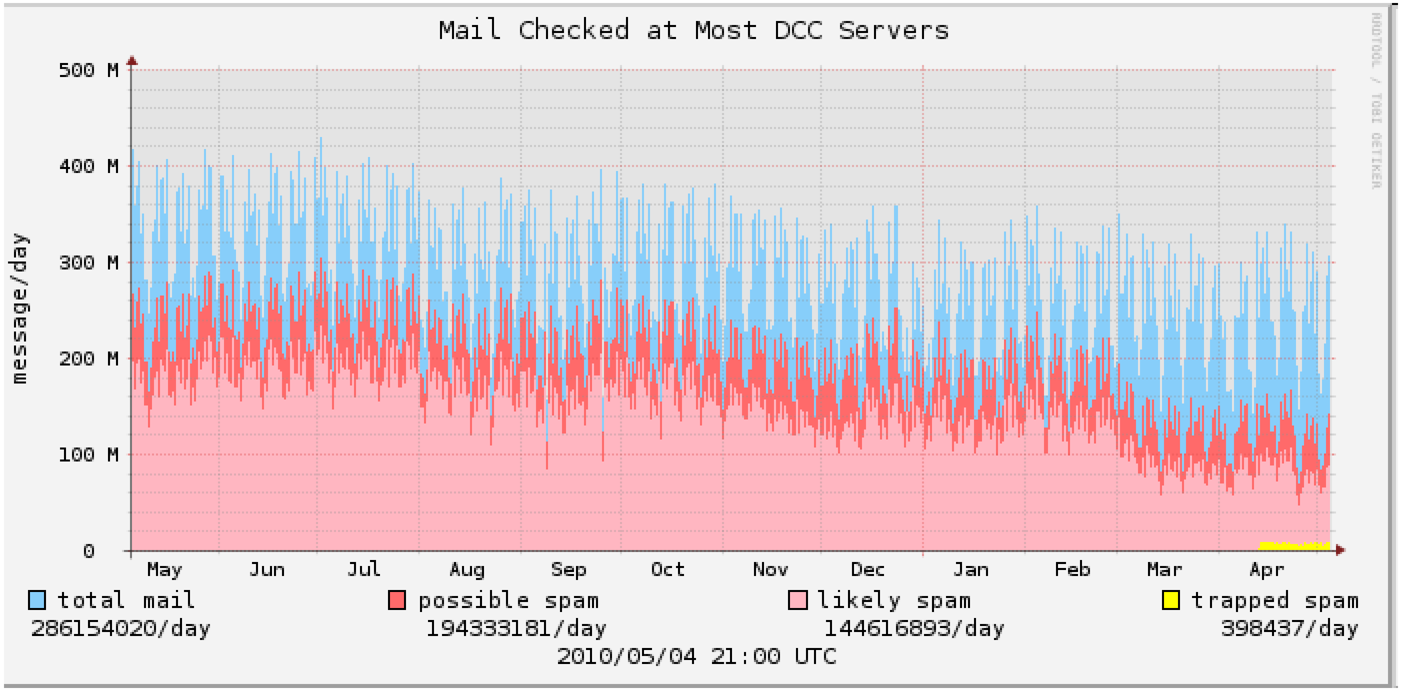
- The idea of DCC is that if mail recipients could compare the mail they receive, they could recognize unsolicited bulk mail. A DCC server totals reports of **checksums** of messages from clients and answers queries about the total counts for checksums of mail messages. A DCC client reports the checksums for a mail message to a server and is told the total number of recipients of mail with each checksum. If one of the totals is higher than a threshold set by the client and according to local whitelists the message is unsolicited, the DCC client can log, discard, or reject the message.
- Because simplistic checksums of spam would not be effective, the main DCC checksums are fuzzy and ignore aspects of messages. The fuzzy checksums are changed as spam evolves. Since DCC started being used in late 2000, the fuzzy checksums have been modified several times.

# DCC

- NAME
- DCC - Distributed Checksum Clearinghouse
- [Does not matter if a message is spam or ham. If the content is the same, then it is a ‘bulk email’, therefore most likely a spam]

- DESCRIPTION
- The Distributed Checksum Clearinghouse or DCC is a cooperative, distributed system intended to detect "bulk" mail or mail sent to many people. It allows individuals receiving a single mail message to determine that many other people have received essentially identical copies of the message and so reject or discard the message.
- How the DCC Is Used
- The DCC can be viewed as a tool for end users to enforce their right to "opt-in" to streams of bulk mail by refusing bulk mail except from sources in a "whitelist." Whitelists are the responsibility of DCC clients, since only they know which bulk mail they solicited.
- The only false positives (mail marked as "bulk" by a DCC server that is not) occur when one of the recipients of a message report it to a DCC server as having been received many times or when the "fuzzy" checksums of differing messages are the same. The fuzzy checksums ignore aspects of messages in order to compute identical checksums for substantially identical messages. The fuzzy checksums are designed to ignore only differences that do not affect meanings.

# DCC


Mail Checked at Most DCC Servers
message/day
500 M
400 M
300 M
200 M
100 M
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
🟦 total mail
286154020/day
🟥 possible spam
194333181/day
□ likely spam
144616893/day
🟨 trapped spam
398437/day
2010/05/04 21:00 UTC
RRDTOOL / TOBI OETIKER

# Razor

- Vipul's Razor is a distributed, collaborative, spam detection and filtering network.
- Through user contribution, Razor establishes a distributed and constantly updating catalogue of spam in propagation that is consulted by email clients to filter out **known spam**.
- Detection is done with statistical and randomized signatures that efficiently spot mutating spam content.
- User input is validated through reputation assignments based on consensus on report and revoke assertions which in turn is used for computing confidence values associated with individual signatures.

# Pyzor

- Pyzor initially started out to be merely a Python implementation of Razor, but due to the protocol and the fact that Razor's server is not Open Source or software libre, Frank Tobin decided to implement Pyzor with a new protocol and release the entire system as Open Source and software libre.
- **Protocol**
- The central premise of Pyzor is that it **converts** an **email** message to a **short digest** that uniquely identifies the message. Simply hashing the entire message is an ineffective method of generating a digest, because message headers will differ when the content does not, and because spammers will often try to make a message unique by injecting random/unrelated text into their messages.

# Pyzor protocol

- To generate a digest, the 2.0 version of the Pyzor protocol:
- Discards all message headers.
- If the message is greater than 4 lines in length:
  - Discards the first 20% of the message.
  - Uses the next 3 lines.
  - Discards the next 40% of the message.
  - Uses the next 3 lines.
  - Discards the remainder of the message.
- Removes any 'words' (sequences of characters separated by whitespace) that are 10 or more characters long.
- Removes anything that looks like an email address (X@Y).
- Removes anything that looks like a URL.
- Removes anything that looks like HTML tags.
- Removes any whitespace.
- Discards any lines that are fewer than 8 characters in length.

# Spamassassin configuration

- # more /etc/mail/spamassassin/local.cf
- loadplugin     Mail::SpamAssassin::Plugin::Razor2
- score RAZOR2_CHECK 3.0
- razor_timeout 5
- …
- use_bayes 1
- bayes_store_module               Mail::SpamAssassin::BayesStore::MySQL #use DB for bayes db
- bayes_sql_dsn                  DBI:mysql:spamassassin;mysql_read_default_file=/etc/mylatin2.cnf
- bayes_sql_username               spamassassin
- bayes_sql_password              aaa
- auto_whitelist_factory           Mail::SpamAssassin::SQLBasedAddrList
- user_awl_dsn                  DBI:mysql:spamassassin ->use database for while list
- user_awl_sql_username             spamassassin
- user_awl_sql_password            aaaaa
- rbl_timeout 10 #->rbl timeout, important for
- pyzor_timeout 5 #pyzor-python razor
- dns_available yes
- dcc_home /var/dcc #distributed checksum clearinghouse
- …

# Re-Defining specific score

- score BAYES_80  5.0
- score BAYES_90  6.0
- score BAYES_99  10.0
- score DRUGS_ERECTILE 2.2
- score FORGED_RCVD_HELO 2.0
- score DNS_FROM_RFC_ABUSE 0.6
- score FORGED_YAHOO_RCVD 2.6
- score SARE_MLH_Stock1 2.2
- score  NO_REAL_NAME 0.4
- score RCVD_IN_SORBS_DUL 2.3

# White list

- Problematic, should set in config file
- Only administrator might change it, needs restart
- E.g.
- whitelist_from …@index.hu
- whitelist_from ….@mailbox.hu
- …
- …

# Adding new RBLs

- header RCVD_IN_NOMOREFUNN eval:check_rbl('nomorefunn', 'no-more-funn.moensted.dk', '127.0.0.[234567]')
- describe RCVD_IN_NOMOREFUNN Received via a relay in No more funn
- tflags RCVD_IN_NOMOREFUNN net
- score RCVD_IN_NOMOREFUNN 2.3

- header RCVD_IN_ABUSEAT eval:check_rbl('abuseat', 'cbl.abuseseat.org', '127.0.0.[234567]')
- describe RCVD_IN_ABUSEAT Received via a relay in cbl.abuseseat.org
- tflags RCVD_IN_ABUSEAT net
- score RCVD_IN_ABUSEAT 2.3

- header RCVD_IN_surriel eval:check_rbl('surriel', 'psbl.surriel.com', '127.0.0.[234567]')
- describe RCVD_IN_surriel Received via a relay in psbl.surriel.com
- tflags RCVD_IN_surriel net
- score RCVD_IN_surriel 2.3

# Spamassassin “positive” rules examples

- #:/etc/mail/spamassassin# more 70_boldi_pozitiv.cf

- body  BOLDI_AUTOGEN_1    /THIS DOCUMENT IS AUTOMATICALLY GENERATED/
- score  BOLDI_AUTOGEN_1    -2.167

- body  BOLDI_UTAZAS_1    /AT FERIHEGY AIRPORT/
- score  BOLDI_UTAZAS_1    -2.167

- body  BOLDI_UTAZAS_2    /SERVICE.*FROM.*TO.*DEPART.*ARRIVE/
- score  BOLDI_UTAZAS_2    -2.2

- body  BOLDI_HERTZ2   /Renting Location/i
- describe BOLDI_HERTZ2  hertz
- score  BOLDI_HERTZ2  -2.0

- body  BOLDI_HERTZ3   /Rate is Guaranteed/i
- describe BOLDI_HERTZ3  hertz
- score  BOLDI_HERTZ3  -2.0

- body  BOLDI_HERTZ4   /Rate is Guaranteed/i
- describe BOLDI_HERTZ4  hertz
- score  BOLDI_HERTZ4  -2.0

- body  BOLDI_WIZZ   /Wizz Air Hungary Kft/i
- describe BOLDI_WIZZ  Automatic ordering system message -wizz
- score  BOLDI_WIZZ  -3.2

- body  BOLDI_UTAZAS13   /minutes before scheduled departure/i
- describe BOLDI_UTAZAS13  Automatic ordering system message -wizz
- score  BOLDI_UTAZAS13  -2.2

# Example spamassassin rules extension

- body TAMAS_CFORM1 /https...docs.google.com.forms/i
- describe TAMAS_CFORM1 Contains Google Form link
- score TAMAS_CFORM1 1.8

- rawbody TAMAS_CFORM2 /background-color.rgb/i
- describe TAMAS_CFORM2 Custom rgb defined background in html mail
- score TAMAS_CFORM2 2.2

- meta TAMAS_CFORM_M (TAMAS_CFORM1 && TAMAS_CFORM2)
- score TAMAS_CFORM_M 2

- header TAMAS_CRYSYS_AT_CRYSYS To =~ /crysys@crysys\.hu/i
- describe TAMAS_CRYSYS_AT_CRYSYS Emails sent to crysys@crysys.hu are regularly spam
- score TAMAS_CRYSYS_AT_CRYSYS 2.0

# Spamassassin problems

- AWL / auto while list database might go too large
- Cleaning up e.g.: delete from awl where totscore>0 and count<3;
- E.g.
- mysql> delete from awl where totscore>0 and count<3;
- Query OK, 245531 rows affected (24.47 sec)
- Bayes-seen database might do too large – cleaning might be needed
- At start, bayes database is clean and unrealiable – if You do not force learning (sa-learn), then the first messages might turn the filter into a wrong direction,…
- Database corruptions might cause huge problems – regular backups are the best – take care of character encoding

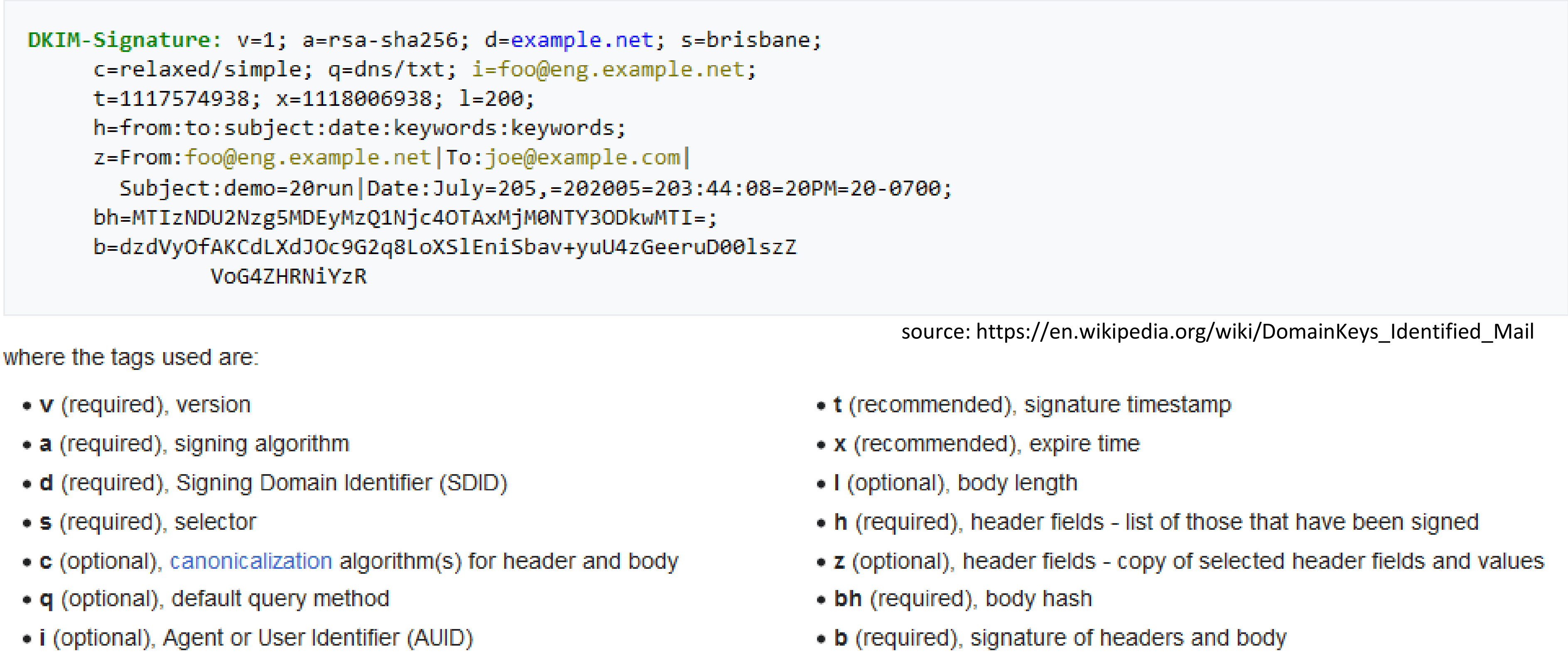
# Setting up DKIM for mail servers – how it works

```
# dig gamma._domainkey.gmail.com in any
; <<>> DiG 9.7.2-P3 <<>> gamma._domainkey.gmail.com in any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39164
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4
;; QUESTION SECTION:
;gamma._domainkey.gmail.com.    IN      ANY

;; ANSWER SECTION:
gamma._domainkey.gmail.com. 195 IN      TXT     "k=rsa\; t=y\;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDIhyR3oItOy22ZOaBrIVe9m/iME3RqOJeasANSpg2YTHTYV+Xtp4xwf5gTjCmH
QEMOs0qYu0FYiNQPQogJ2t0Mfx9zNu06rfRBDjiIU9tpx2T+NGlWZ8qhbiLo5By8apJavLyqTLavyPSrvsx0B3YzC63T4Age2CDqZYA+OwS
MWQIDAQAB“

;; AUTHORITY SECTION:
gmail.com.          345520  IN      NS      ns1.google.com.
gmail.com.          345520  IN      NS      ns3.google.com.
gmail.com.          345520  IN      NS      ns4.google.com.
gmail.com.          345520  IN      NS      ns2.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.     121886  IN      A       216.239.32.10
ns2.google.com.     121886  IN      A       216.239.34.10
ns3.google.com.     121886  IN      A       216.239.36.10
ns4.google.com.     121886  IN      A       216.239.38.10
```

# DKIM

- Domain Keys Identified Mail - http://www.dkim.org/
- RFC 4871
- Used to sign all emails from a domain
- Endpoint can check if the signature matches domain record
- If not, the email might be fake/spam
- For valid domain registrations, it might not work.
- Most spammers register their own domains for purchase, website, but not for sending out emails as it is easy to filter out.

# DKIM signature example



```
DKIM-Signature: v=1; a=rsa-sha256; d=example.net; s=brisbane;
     c=relaxed/simple; q=dns/txt; i=foo@eng.example.net;
     t=1117574938; x=1118006938; l=200;
     h=from:to:subject:date:keywords:keywords;
     z=From:foo@eng.example.net|To:joe@example.com|
       Subject:demo=20run|Date:July=205,=202005=203:44:08=20PM=20-0700;
     bh=MTIzNDU2Nzg5MDEyMzQ1Njc4OTAxMjM0NTY3ODkwMTI=;
     b=dzdVyOfAKCdLXdJOc9G2q8LoXSlEniSbav+yuU4zGeeruD00lszZ
              VoG4ZHRNiYzR
```

source: https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail

where the tags used are:

- **v** (required), version
- **a** (required), signing algorithm
- **d** (required), Signing Domain Identifier (SDID)
- **s** (required), selector
- **c** (optional), canonicalization algorithm(s) for header and body
- **q** (optional), default query method
- **i** (optional), Agent or User Identifier (AUID)
- **t** (recommended), signature timestamp
- **x** (recommended), expire time
- **l** (optional), body length
- **h** (required), header fields - list of those that have been signed
- **z** (optional), header fields - copy of selected header fields and values
- **bh** (required), body hash
- **b** (required), signature of headers and body

- Important tags: **a**lgorithm, **d**omain, **h**eader fields, **b**ody **h**ash, signature (**b**)
- b=base64(sign(bh+hash(h)))

# DKIM slides http://www.dkim.org/

- **D. Crocker** ~ bbiw.net
- dkim.org
- Consortium spec
  - –Derived from Yahoo DomainKeys and Cisco Identified Internet Mail
- IETF published revision – RFC 4871

Allows an organization to claim responsibility for transmitting a message, in a way that can be validated by a recipient

- Validate identifier and msg data integrity
  - DNS identifiers
  - Public keys in DNS
- End-to-end
  - Between origin/receiver administrative domains
  - Not path-based

# DKIM2

- Based on message content, itself
  - Not related to path
- Transparent to end users
  - No client User Agent upgrades *required*
  - But extensible to per-user signing
- Allow signature delegation
  - Outsourcing
- Low development, deployment, use costs
  - Avoid large PKI, new Internet services
  - No trusted third parties (except DNS)

# DKIM 3

- Signs body and selected parts of header
- Signature transmitted in DKIM-Signature: header
- Public key stored in DNS
  - In _domainkey subdomain
  - Uses TXT RR
- Namespace divided using selectors
  - Allows multiple keys for aging, delegation, etc.

# DKIM

```
# openssl genrsa -out dkim.private.key 1024
Generating RSA private key, 1024 bit long modulus
..............++++++
....++++++
e is 65537 (0x10001)

# more dkim.private.key
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCy8aeu0qt+dZAL6MN5q7/HXsEU4zo/8E2VjzOw4PHFReC+Zeki
wUtG8cimItrJuGTWd+OULQZtZwuo5mJuRqNpOV5eEvqnmJZkO5Ko+Cbelig7HiOx
Iw9Z5qpafJLQoNrhoJSQx6nlFdVDZX6UN/dQJEKyNYjgDxV0jmLvCOYVpwIDAQAB
AoGAB4ZqCswJMjQ3bojHT6KNWhD+BabYmD++w39WKSYMhMM/hEI8351JUR5x1aq5
kKoUc4BvLZgHISqoLt4hXYnS7VPerbiYP3JSuExjOIbU6omRijyqU6NIgUt/PLb5
MYQfWwImXyTll5IjI+hQmvFhNPD3aPfIpR6kNzXSRkZFFEECQQDbCAMlJhQtmrke
j8+P2z+itDDS0nEOCDHXYYMCP5v1cI6MPsMA9j2LNx0QJpPo5A3XA3/aNMMo4Vmt
p8S8NdZ5AkEA0SWIspIQZHcm7GHhwzEdDFcTv+7CuCBeTmYsusxJWrUWYGAWTAVa
SJnfatigoqkmmt5pywQlI8HdwQLdtYbFHwJBAMvElnh3BoquyM3/6I/i3z7U+B8K
HJd7VDMVyrXKy6L5RgR/VxeL/hGIAofF/BDMKXwBC27La0ya6b0+uS6Hv7ECQEq9
WwhVYj5ExkgbAo66cmMCizA/pp4eExV5NerbLiuYIxL1w4IPN6BSPKD2IRF/2Sfm
6299X7hTg2eCGrDQJIkCQQDV1nCJUiyK34IohkowiqXAgqE3cAWkZ9XY4zgYFX9L
PhxYMW4zndm3AL9NUd4ib9D3xDkc6hZbDHHeEBkTLMSV
-----END RSA PRIVATE KEY-----
```

//Of course this is not the real key… ;)

# Generate keys for signing

```
# openssl rsa -in dkim.private.key -out dkim.public.key -pubout -outform PEM
writing RSA key
# ls -la
total 44
drwxr-xr-x  2 root root  4096 Mar  7 23:25 .
drwxr-xr-x 14 bind root 32768 Mar  7 23:22 ..
-rw-r--r--  1 root root   887 Mar  7 23:24 dkim.private.key
-rw-r--r--  1 root root   272 Mar  7 23:25 dkim.public.key
# more dkim.public.key
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDPiAQp5v2WLboPeciMd5b4G+jr
+88/hykseyyOY0Y26jhVNJOJOWtlKrpTMdPjX5cvhv0yWdhMIXY3ZkGoNkcxCVUa
wup53IOJwI3tEBlg0IDZfmjNx3BkPdlSdXN7ezTYEBddrAKnjzUJf0qlo4cleoCA
Ara3UUARe4fwvGNl/QIDAQAB
-----END PUBLIC KEY-----
```

# Modifying DNS record for DKIM

- http://www.dnswatch.info/dkim/create-dns-record
- **Verifier service: check-auth@verifier.port25.com**
- shamir._domainkey   TXT v=DKIM1; t=s; k=rsa;
  p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDPiAQp5v
  2WLboPeciMd5b4G+jr+88/hykseyyOY0Y26jhVNJOJOWtlKrpTMdP
  jX5cvhv0yWdhMIXY3ZkGoNkcxCVUawup53IOJwI3tEBlg0IDZfmjN
  x3BkPdlSdXN7ezTYEBddrAKnjzUJf0qlo4cleoCAAra3UUARe4fwvG
  Nl/QIDAQAB
- eternal._domainkey   TXT v=DKIM1; t=s; k=rsa;
  p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCkI1xqg49
  bAvPSB2QiEDRenlcPSxp3O53JqLOEzorTOMUU/8b6kuWSWvFbTo
  Mw8kA3QDHE2c+No+OIJG3Ru7vaQkKJd08tlOe/jtGO23G0ajPQR
  7vVnnE3uIEVoqJNa6e4KPJR4MfoOQY3XDZWj/dhY6CUXwSKYY25
  7l3f7ZvYqQIDAQAB

# Exim - router

- dnslookup_dkim:
- debug_print = "R: dnslookup_dkim for $local_part@$domain"
- driver = dnslookup
- domains = ! +local_domains
- transport = remote_smtp_dkim
- senders = lsearch*@;/etc/exim4/dkim_senders
- same_domain_copy_routing = yes
- # ignore private rfc1918 and APIPA addresses
- ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8 : 192.168.0.0/16 :\
- 172.16.0.0/12 : 10.0.0.0/8 : 169.254.0.0/16 :\
- 255.255.255.255
- no_more

- # more /etc/exim4/dkim_senders
- *@crysys.hu:    crysys.hu

# Exim - transport

- remote_smtp_dkim:
-   debug_print = "T: remote_smtp_dkim for $local_part@$domain"
-   driver = smtp

- dkim_domain = ${lookup{$sender_address}lsearch*@{/etc/exim4/dkim_senders}}
- dkim_selector = shamir
- dkim_private_key = /etc/exim4/keys/dkim.shamir.key
- dkim_canon = relaxed
- dkim_strict = false
- #dkim_sign_headers = DKIM_SIGN_HEADERS

# Result in mail  header

- DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/relaxed; d=crysys.hu;
  s=shamir; h=From:Message-ID:Date;
  bh=r//ncCXDqN/FtRvXIA4ipCVtrDt5wpwP4VxqpzQYvc0=;
  b=mQUM28E1EibM+STbi7HCk2mYWrrj3UFDWxrAcsy9wxICp+4IuUShKzRh32u9ps3Dka
  1v0y7frQEnFzERaLKs5RPf2AKcUslZZtEaS3Mbt86UY5zU4TRI3B2YrhmzN5UVsvCBxrp6TX
  X+mTmUS/3+gfM3rrMnK7Vh08YzwhQHKa4=;

# DMARC

- **Domain-based Message Authentication, Reporting and Conformance** or **DMARC**
- **Defines what to do with SPF/DKIM protected emails when received**
- **Quarantine, reject, collect and report information back to the domain owner**

```
dig _dmarc.crysys.hu @dc.hu in any

; <<>> DiG 9.7.3 <<>> _dmarc.crysys.hu @dc.hu in any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9274
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;_dmarc.crysys.hu.          IN      ANY

;; ANSWER SECTION:
_dmarc.crysys.hu.      3000    IN      TXT     "v=DMARC1\; p=quarantine\; pct=5\; rua=mailto:dmarc@crysys.hu\; sp=r\;
aspf=r"
_dmarc.crysys.hu.      3000    IN      RRSIG   TXT 8 3 3000 20130510062720 20130425181400 741 crysys.hu.
xAI3Nc0OqRiUJirU67SLwCYG9uRUQFLC6pgQa1S19B4qn1WZNRQVXIoG
WlZQxABiAFkPN2d6p8nDR7zHruAhEgzM/RZ2GSRO2tSjFYy/WHyephNR
X69WWoNViNW9DKiXR2YH4g6rBTNHk0MSVStTVYyfNAP7anaKkZTaAEjI
GytbqkL/C31o9ZMt+LqwtiCDEwDbxh5aMK2GmORsYp+XVYilHU0ba9UF
ue9B4NuHzGKcECt1voEU0+8pJzxUjDzPzr7ufJ9Z96sxrzxfbZ/u79Q7
EDb2iIq0NZxS6kBDtqc2sudt1rps4Vj++Ax4uliDWo6kUXWVwdSm7oBZ DE4izg==
```

# Control questions

- What is the difference between DoS and DDoS attacks?
- What are the main types of DoS attacks
- Define magic packet DoS and give an example
- How SYN attacks work
- How bandwidth consumption attacks work? Why do the attackers need multiple computers for that? Who do they perform the attack?
- Give an example how would you extend spam filtering with a new rule on spam detection! The same on “positive rule”? (any format, pseudolanguage etc. is accepted)
- Name two options to help avoiding false positives in spam filtering!

# Control questions

- Define the function of the following solutions:
  - Antivirus-antispam middleware (amavisd)
  - Heuristic anti-spam solutions (spamassassin)
  - Greylisting
  - RBL, DNSBL, URIBL
  - DCC, Razor, Pyzor
  - DKIM
  - SPF
  - DMARC
  - Auto Whitelisting (AWL)
- What is bayes filtering, how does it work? (give typical equations too)
- How effective is greylisting+reverse DNS checking?
- What is a reflective attack?
- Define reflective DoS amplification attack, give 2 examples
- Tell us solutions that use RSA keys in spam or DoS protection