

# Spam - verifying the sender

Bo Simonsen <[bosim05@imada.sdu.dk](mailto:bosim05@imada.sdu.dk)>

<http://imada.sdu.dk/~bosim05/>

DM71 - Computer Security

Institute for Mathematics and Computer Science

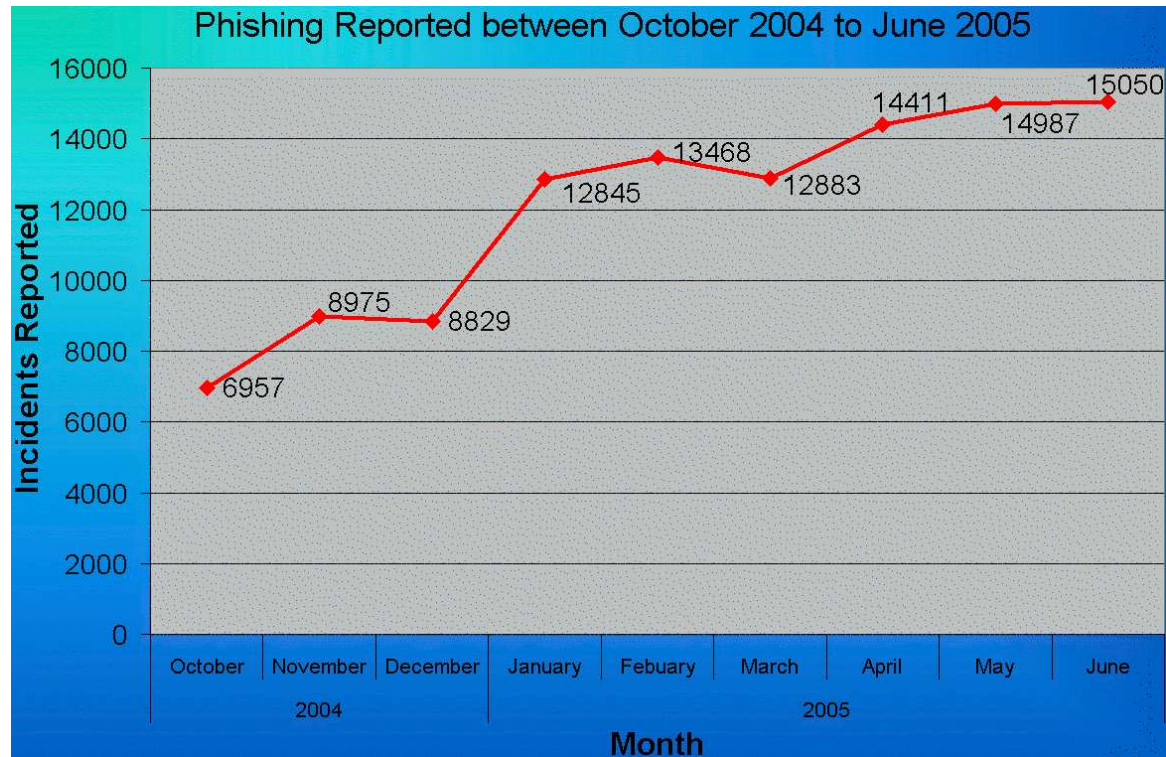
South Danish University, Odense

# Agenda

- Spam - the problem (formally UCE - Unsolicited commercial e-mail)
- Current methods
- Basic knowledge (SMTP & DNS)
- DKIM
- Sender ID
- Advantages
- Current status

# The problem

Phishing issues reported to <http://www.antiphishing.org>



Just a few problems about spam:

- SPAM means Phishing
- Overloaded mailboxes / traffic expenses

# Current methods

A few examples:

- Bayesian filters

$$Pr(spam|words) = \frac{Pr(words|spam)Pr(spam)}{Pr(words)}$$

- RBL (Realtime Blackhole List) / DNSBL (DNS Blackhole List)

62.242.216.110  $\Rightarrow$  110.216.242.62.relays.ordb.org

If a DNS lookup succeeds on above address, the client is listed, else it's not.

# Mail transfer

Mail is exchanged by using SMTP (**S**imple **M**ail **T**ransfer **P**rotocol):

```
220 localhost.localdomain ESMTP Exim 4.50 Wed, 08 Mar 2006 10:32:34 +0100
HELO geekworld.dk
250 localhost.localdomain Hello bo at localhost [127.0.0.1]
MAIL FROM: <fake@emailaddress.com>
250 OK
RCPT TO: <bo@geekworld.dk>
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
From: Spam sender <fake@emailaddress.com>
To: Bo Simonsen <bosim05@imada.sdu.dk>
Date: Tue, 7 Mar 2006 21:52:16 +0100 (CET)
Message-ID: <random-stuff@domain.dk>

Some text
.
250 OK id=1FGv2e-0000v1-Su
quit
221 localhost.localdomain closing connection
```

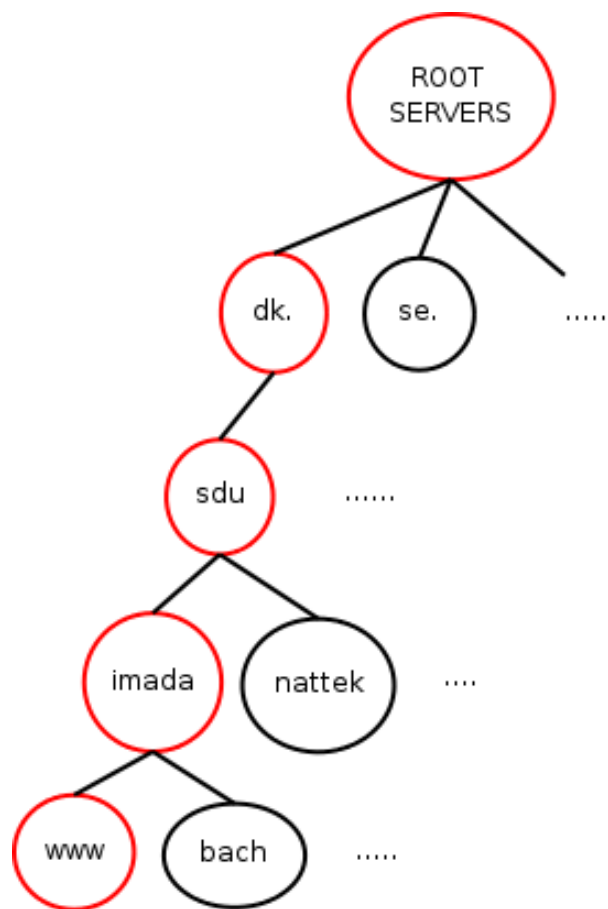
## Problems about normal mail transfer:

- Sender server isn't verified.
- Normally everything for local domains is accepted, which means you can fake the sender address.
- Normally a mailserver relays all mail from a given IP range. Like a ISP is relaying mail from their customers controlled by the IP.

**This makes life easy for spammers!**

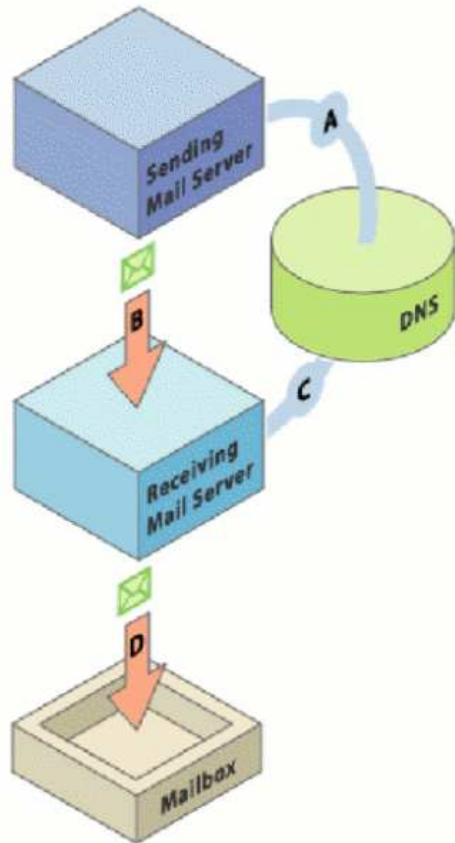
Domain Name System is a system which can translate hostnames to ip addresses

An example lookup of `www.imada.sdu.dk`



- A record - Hostname to IP fx.  
`$ host -t a mozart.imada.sdu.dk`  
`mozart.imada.sdu.dk has address 130.225.128.10`
- MX record - Hostname to mail exchanger  
`$ host -t mx imada.sdu.dk`  
`imada.sdu.dk mail is handled by 5`  
`berlioz.imada.sdu.dk.`
- CNAME record - Alias to hostname  
`$ host -t cname www.imada.sdu.dk`  
`www.imada.sdu.dk is an alias for`  
`mozart.imada.sdu.dk.`
- TXT record - Descriptive text - used for several purposes including SPF and domainkeys

Domain **K**ey **I**dentified **M**ail works in the following manner:



- A) Sender server publish public key in DNS.
- B) By sending a mail, a sum (not specified by standard but SHA1 is widely used) is calculated on selected headers. The sum is signed by the private key and encoded in base64.
- C) Reciever server looks public key up using DNS and calculate the sum on selected headers and, it decodes and verify the recieved sum.
- D) If the sum verifies, the sender server is verified, and the mail can be delivered.

# The DKIM header

An example DKIM header:

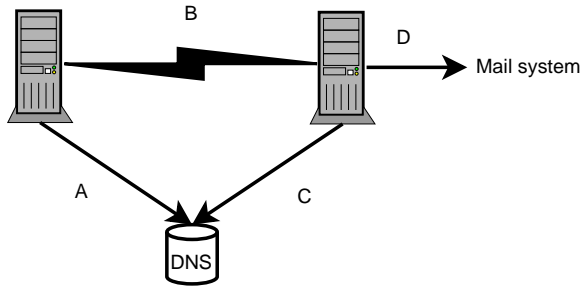
```
DomainKey-Signature: a=rsa-sha1; q=dns; c=noaws;  
s=beta; d=gmail.com;  
h=received:message-id:date:from:to:subject:mime-version:content-type;  
b=tHX1w3JH8T/INYEBNqeHXK1YkGaILaK8RbAAc7rT5Uqaga3P9su9I6vm/IMbyAfiCbbG4  
xWdq1BJWuJffMRZjam617v6W9k2Zz6dfg3U4NMpPRI9PxXyn5bcqIrfRCnuf5ZInXXA0e0l  
euLbv+bwZ/nbZeun3Ze+us+NKmiC4Xg=
```

- a= - Algorithm - RSA signed - SHA1 hash
- h= - Selected headers
- b= - Base64 encoded signed sum

The public key is found by using a DNS lookup like the one below:

```
bo@challenger:~$ host -t txt beta._domainkey.gmail.com  
beta._domainkey.gmail.com descriptive text "t=y k=rsa p=MIGfMAOGCSqGSIb3  
DQEBAQUAA4GNADCBiQKBgQC69TURXN3oNfz+G/m3g5rt4P6nsKmVgU1D6cw2X6BnxKJNlQKm10f8  
tMx6P6bN7juTR1BeD8ubaGqtzm2rWK4LiMJqhoQcwQziGbK1zp/MkdXZEWMCf1LY6oUITrivK7JN  
OLXtZbdxJG2y/RAHGswKKyVhSP9niRsZF/IBr5p8uQIDAQAB"
```

# Sender ID



- A) Sender publishes his SPF record.
- B) Sender start communication with the reciever sender,
- C) The reciever looks up the SPF record, and checks if the sender is allowed to send.
- D) If it's allowed the mail system accepts the mail

```
[columbia:~/cvs/dm71slides] bo % host -t txt hotmail.com
hotmail.com descriptive text "v=spf1 include:spf-a.hotmail.com
include:spf-d.hotmail.com ~all"
[columbia:~/cvs/dm71slides] bo % host -t txt spf-a.hotmail.com
spf-a.hotmail.com descriptive text "v=spf1 ip4:209.240.192.0/19
ip4:207.46.0.0/16 ip4:199.2.137.0/24 ~all"
```

Which means only mailservers which says they are sending from "hotmail.com" and are present in the above ip ranges is allowed to send.

# Advantages/Disadvantages

## Advantages about using DKIM:

- The sender server is verified, we can ensure that it matches the claimed domain. If a server sends spam, it's easy to track.
- Integrity, ensures the integrity due to the SHA1 sum, so we are sure, that the risk for a covert channel is very small.

## Disadvantages about using DKIM:

- Takes CPU power to verify every signature / to sign, and computation of the secure hash functions.

## Advantages about using Sender ID:

- Efficient & Simple.

## Disadvantages about using Sender ID:

- Doesn't ensure integrity.
- Only based on IP access control, so DNS poisoning may cause a weakness.
- It will fail if the sender doesn't use the associated SMTP server to the domain he's sending from.

# Current status

## DKIM:

- Used by gmail and yahoo - Two major providers of free e-mail addresses.
- Implementations for many MTAs exist. An open source library from Yahoo there should make implementations easier exists

## Sender ID:

- Hotmail rejects mail from servers, if there is a SPF record present on the domain, and it doesn't match the server ip.
- The OpenSPF framework makes it possible to use it with other than Microsoft products.

# Final words

- Sender ID can be easily implemented - because a lot of APIs and implementations for MTAs exist. And the domain owner has a simple task, creating a SPF record
- Domainkey can also be done - but a bit harder for the domain owner since he can generate keys, but implementations for major MTAs exist
- Both methods ensure the sender server is allowed to send, Sender ID by IP - Domainkey by verifying a signed "message"

Mail should anyway be accepted, if the domain doesn't have a SPF record or a Domainkey, but verified mail could bypass bayesian filters.