UNIVERSITÄT DUISBURG ESSEN

Internet Technology & Web Engineering

Email Security & Spam

Dr.-Ing. Matthäus Wander

Universität Duisburg-Essen

User Authentication

- IMAP and POP have authentication built-in
  - Original SMTP provided no authentication
  - Anyone could send emails via any MTA

- Idea: Check sender From address
  - Insecure, can be spoofed
- Authenticate by source IP address range
  - Works for closed groups, not for public email service provider
- SMTP after POP (or POP before SMTP)
  - Authenticate via POP, save client IP address
  - Allow SMTP afterwards if IP address matches

SMTP Authentication

- SMTP AUTH protocol extension
  - Requires Extended SMTP
- Plaintext login
  - Base64-encoded (for compatibility, **not** security)
  - Secure only if SMTP connection encapsulated by TLS
- Digest authentication (hashed password)
  - Server sends challenge (arbitrary, unique string)
  - Client „encrypts“ challenge with password, sends response
  - Server checks response with known password
  - Password hidden, but dictionary attacks on response possible

PLAIN Authentication by Example

```
S: 220-smtp.example.com ESMTP Server
C: EHLO client.example.com
S: 250-smtp.example.com Hello client.example.com
S: 250 AUTH GSSAPI DIGEST-MD5 PLAIN
C: AUTH PLAIN dGVzdAB0ZXN0ADEyMzQ=
S: 235 2.7.0 Authentication successful
```

- Base64-encoded username and password
  - base64(identity || 0x00 || $identity_{auth}$ || 0x00 || password)
  - Decoded example: test, test, 1234
- Base64 is not an encryption
  - Anyone can decode Base64 strings and retrieve the password

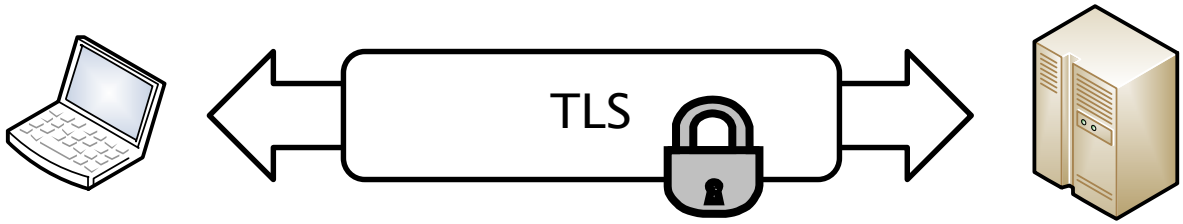
Digest Authentication (CRAM-MD5) by Example

```
S: 220-smtp.example.com ESMTP Server
C: EHLO client.example.com
S: 250-smtp.example.com Hello client.example.com
S: 250-AUTH DIGEST-MD5 CRAM-MD5
S: 250-ENHANCEDSTATUSCODES
S: 250 STARTTLS
C: AUTH CRAM-MD5
S: 334 PDQxOTI5NDIzNDEuMTI4MjgONzJAc291cmNlZm91ci5hbmRyZX
   cuY211LmVkdT4=
C: cmpzMyBlYzNhNTlmZWQzOTVhYmExZWM2MzY3YzRmNGIOMWFjMA==
S: 235 2.7.0 Authentication successful
```

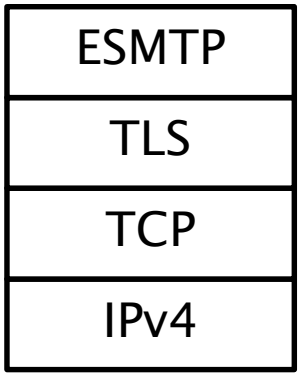
- Client response: username, encrypted password
  - encrypted password = HMAC-MD5$_{password}$(challenge)

TLS Encryption

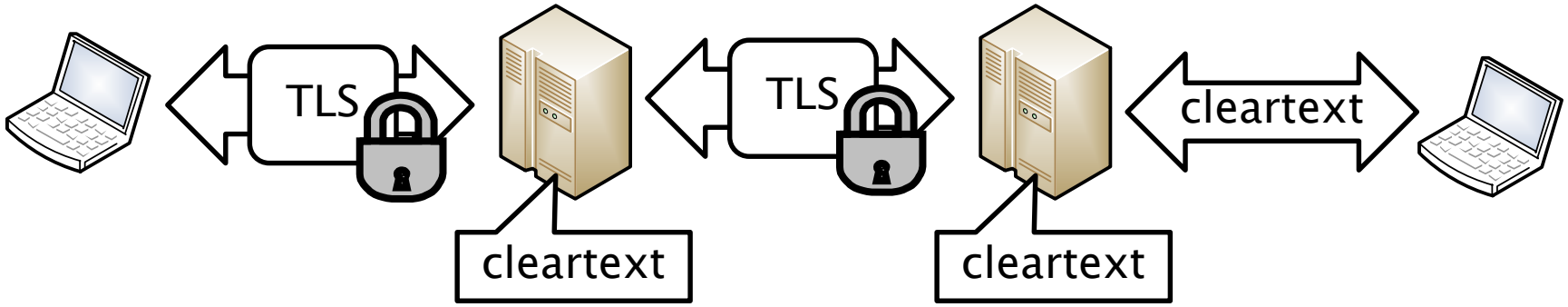
- All email protocols support insecure cleartext transfers



- Encapsulate connection with Transport Layer Security
  - TLS ensures server authenticity, encrypts data
  - Protects from passive surveillance and active man-in-the-middle attacks
- Two approaches:
  1. Connect to extra port for secure SMTP *(deprecated)*
  2. Use STARTTLS command within ESMTP connection

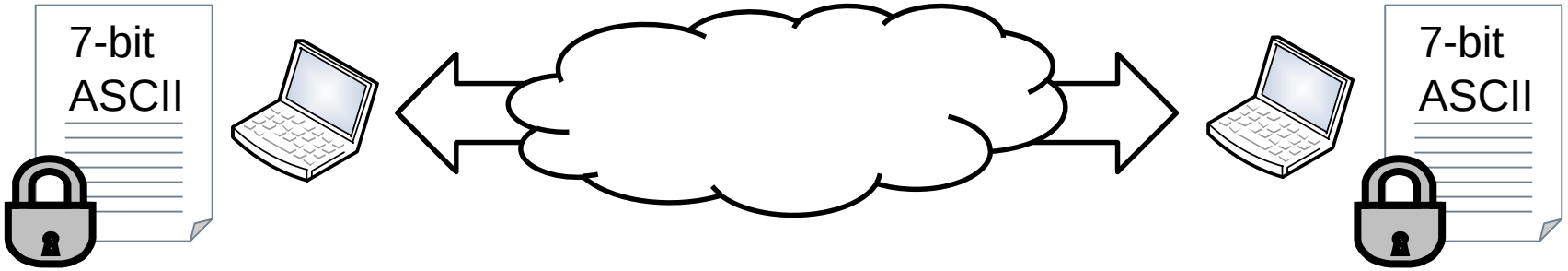


Point-to-Point Security



- TLS encrypts point-to-point between client/server or server/server
  - Server decrypts TLS and processes email as cleartext
- Downgrade attacks possible (attackers blocks TLS)
  - Protection: DANE/DNSSEC
- There is no guarantee that everybody uses TLS
  - e.g. receiver may use IMAP without TLS

End-to-End Security



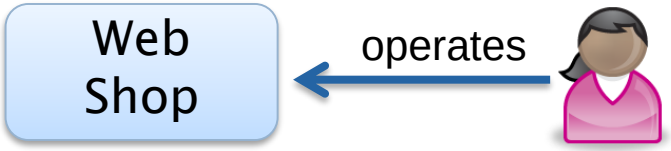
- Encrypt/decrypt emails on end user devices
  - Message content will be hidden from email servers
- Message body is encrypted, but message header is not
  - SMTP servers require cleartext headers to process email
  - Metadata remains visible (sender, receiver, message id, subject)
- Approaches:
  1. PGP or GnuPG: signed and encrypted part of message body
  2. S/MIME: signed and encrypted MIME data

EMAIL SPAM

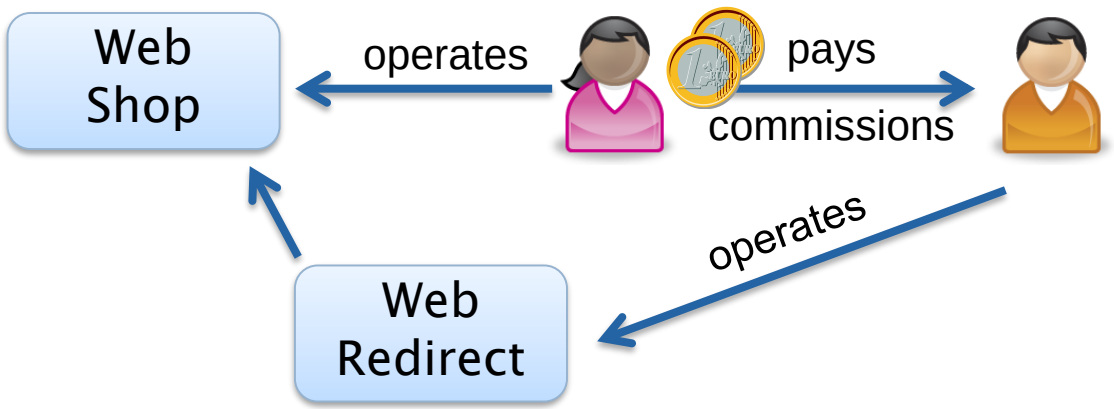
Definitions

- Spam: undesired mass message
- Ham: non-spam message
- Unsolicited Bulk Email (UBE): spam
- Unsolicited Commercial Email (UCE): commercial spam
- False positive: ham message classified as spam
- False negative: spam messages classified as ham
- Spam can occur in all communication platforms
  - Instant messaging, discussion boards, Wikis, SMS, VoIP (SPIT), …
  - We focus on email spam here
- Purpose: advertising, scam, phishing, hoax, …

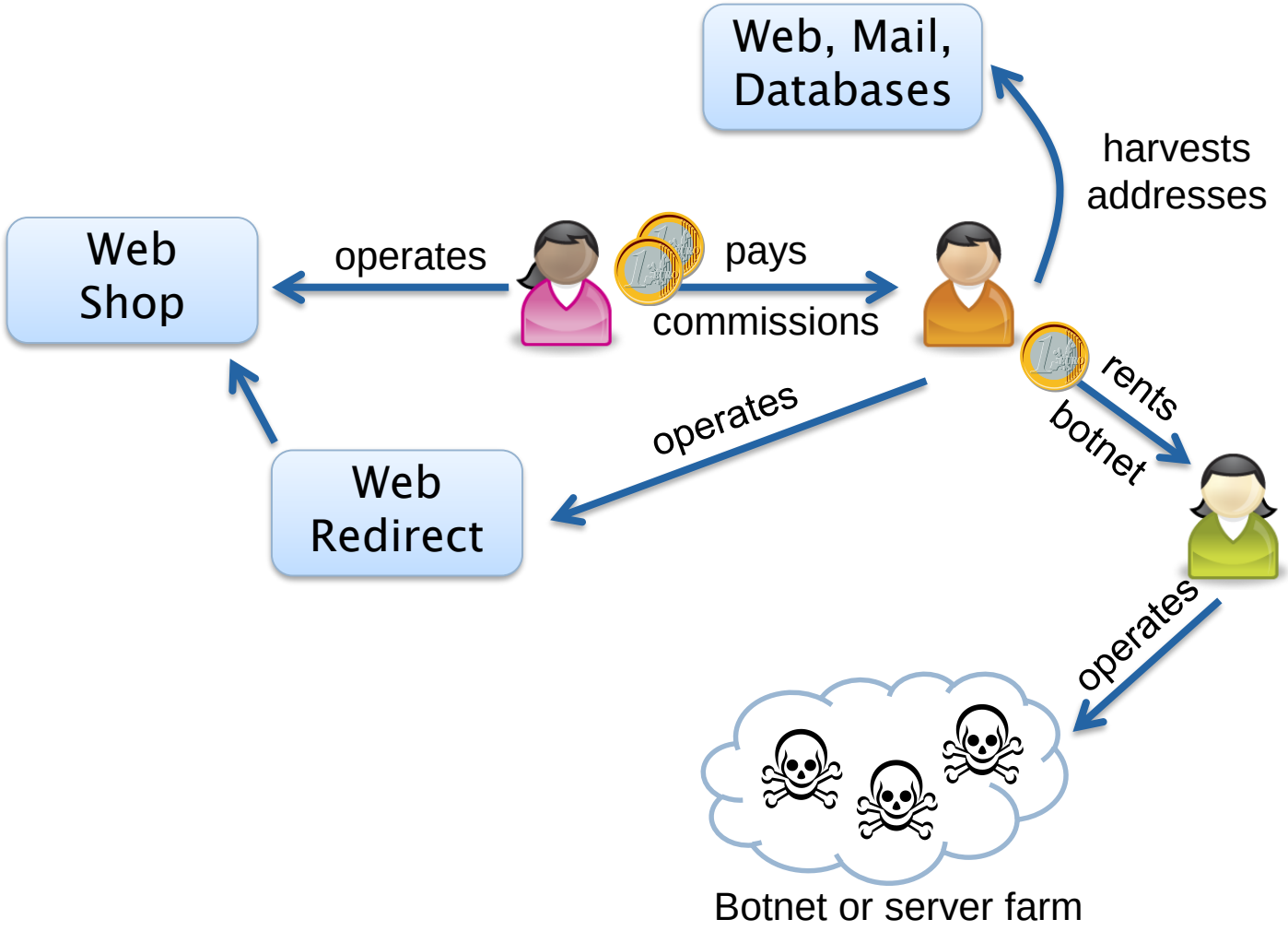
Ecosystem of Commercial Spam



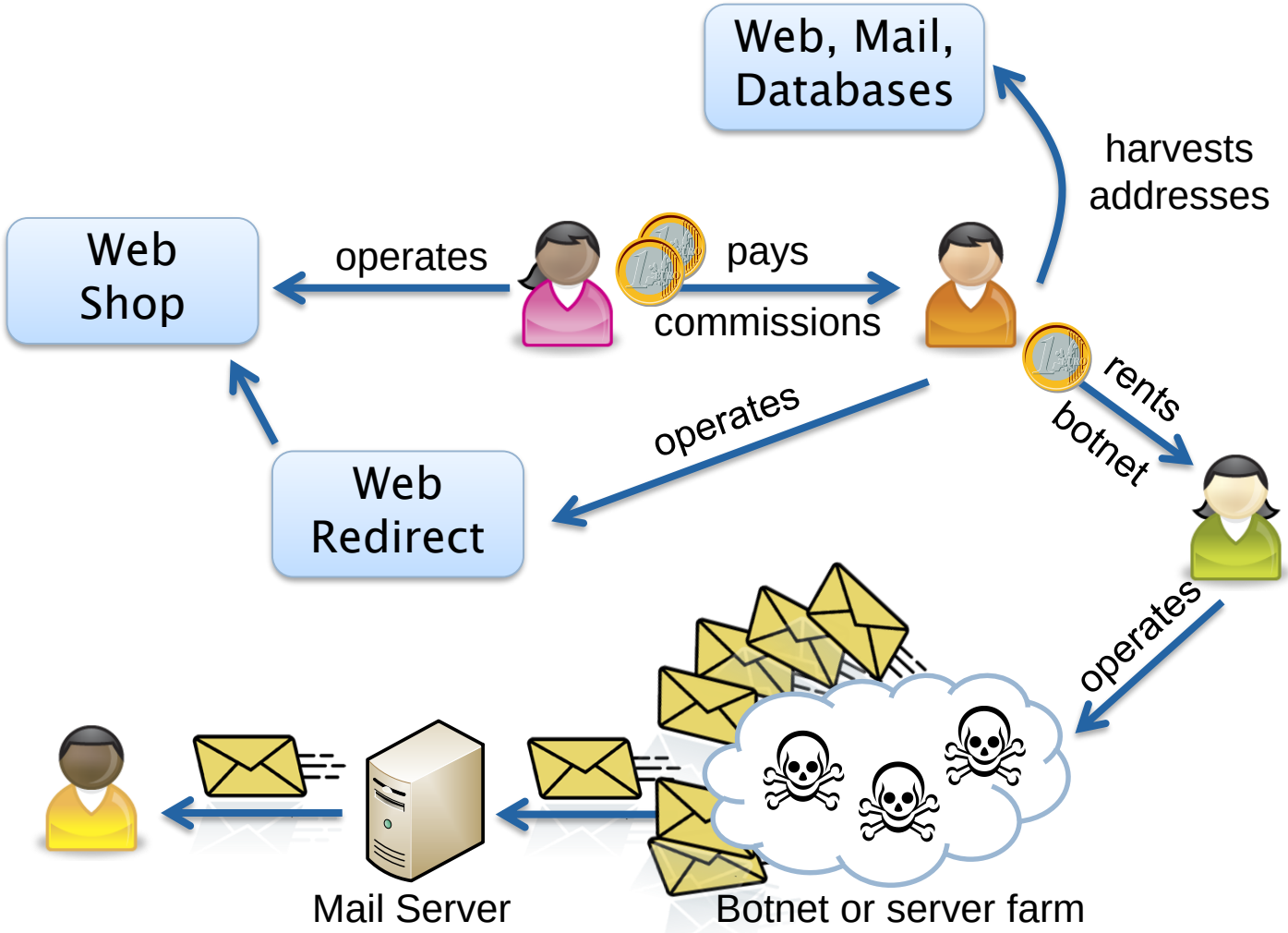
Ecosystem of Commercial Spam



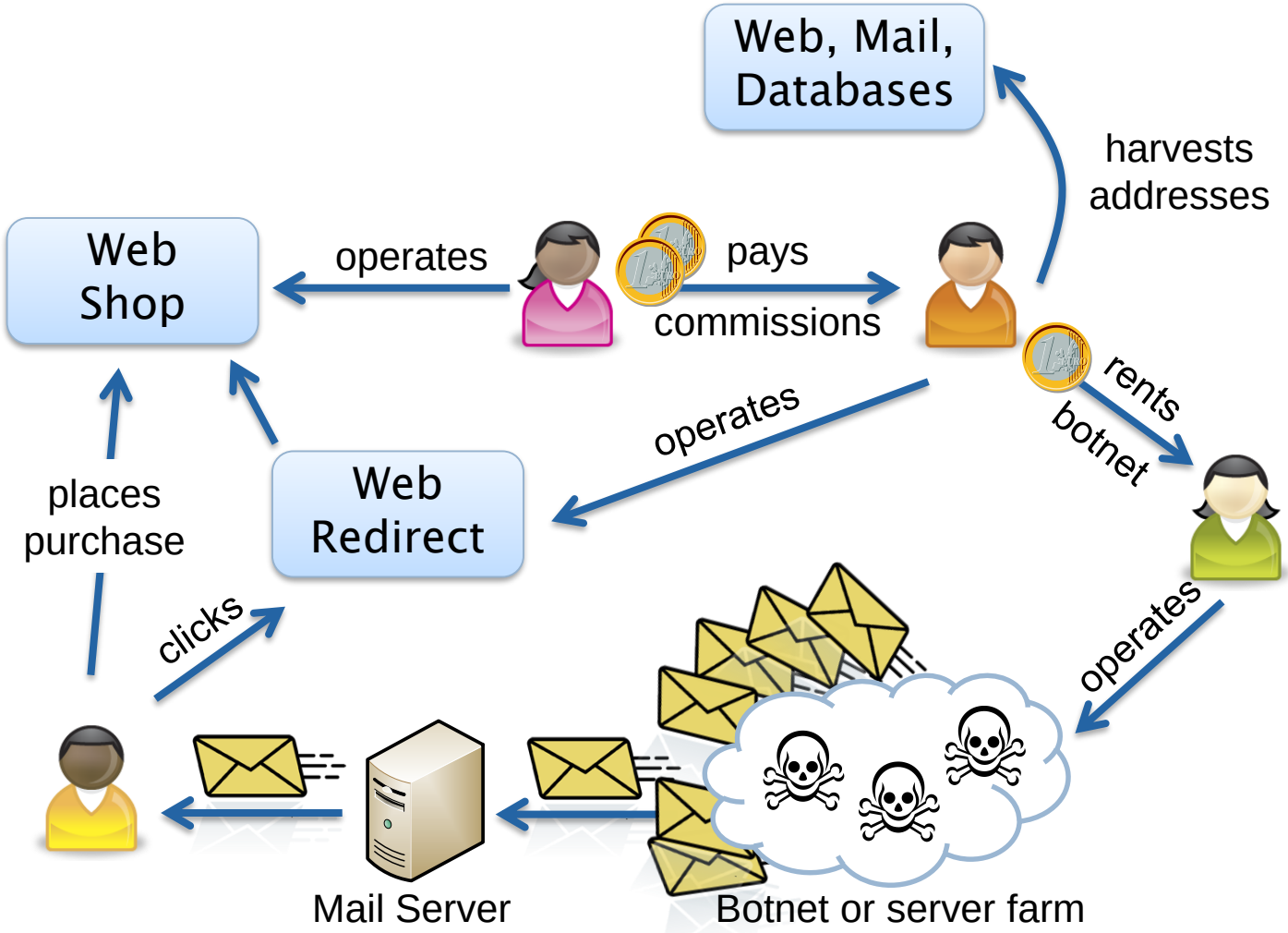
Ecosystem of Commercial Spam



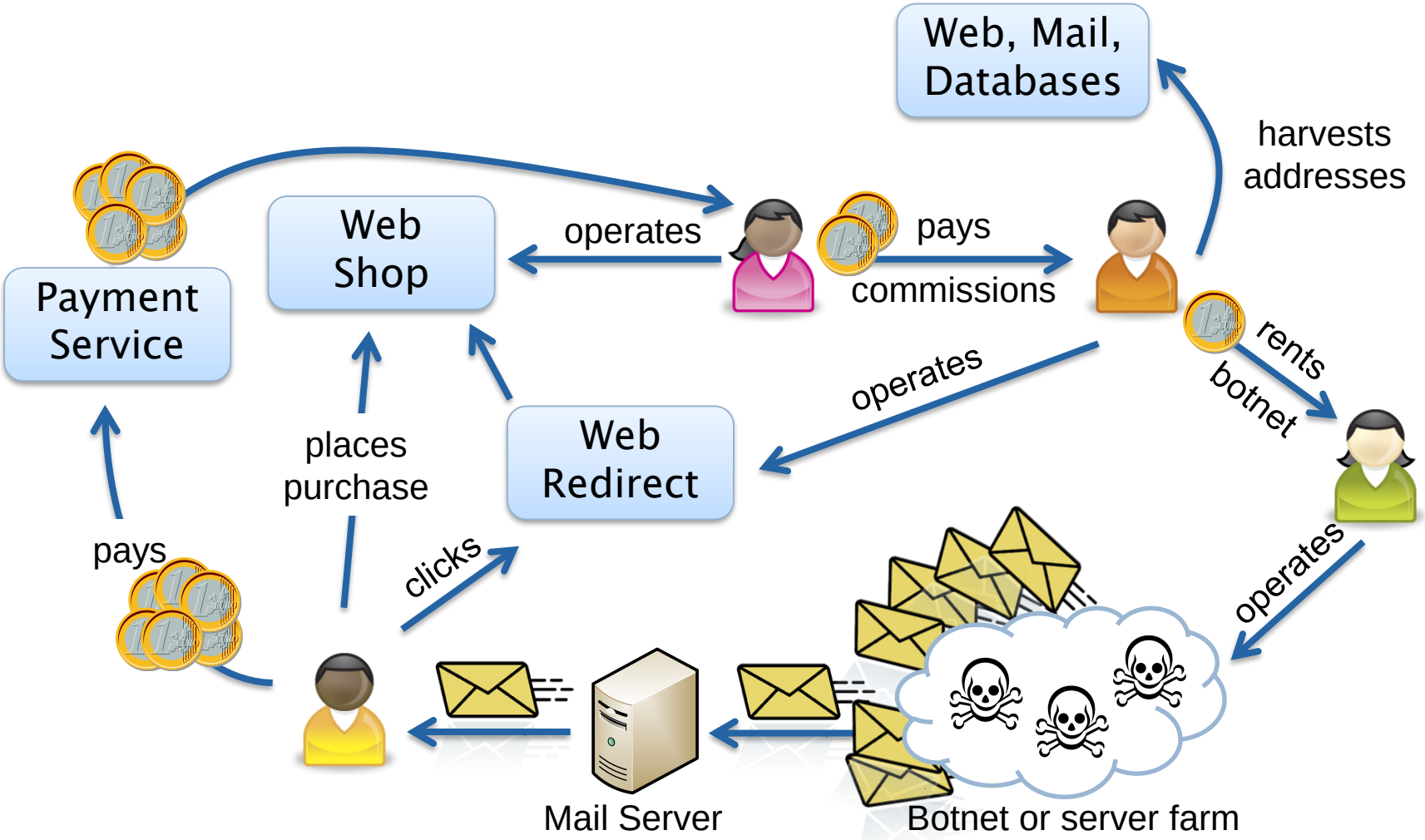
Ecosystem of Commercial Spam



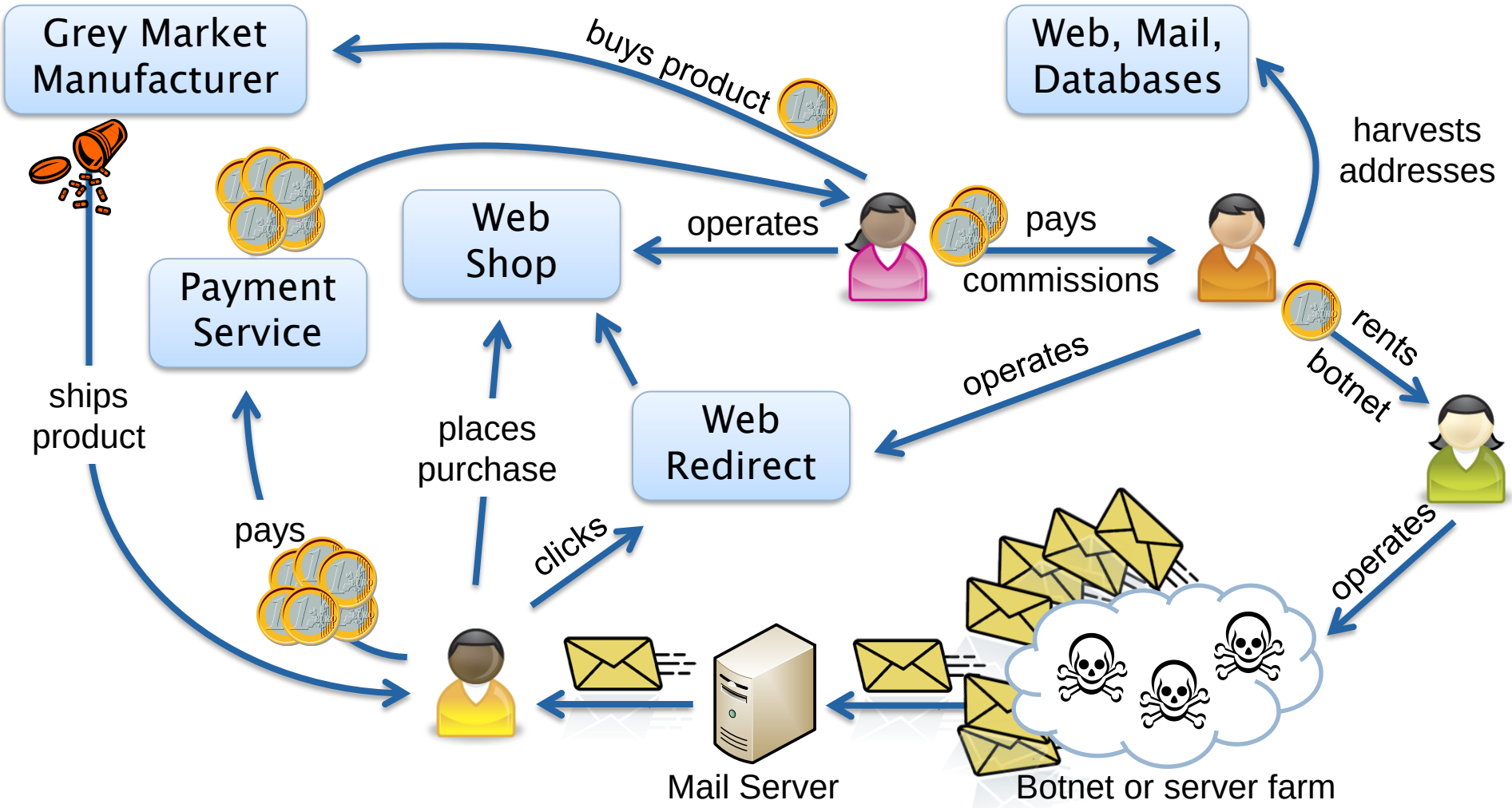
Ecosystem of Commercial Spam



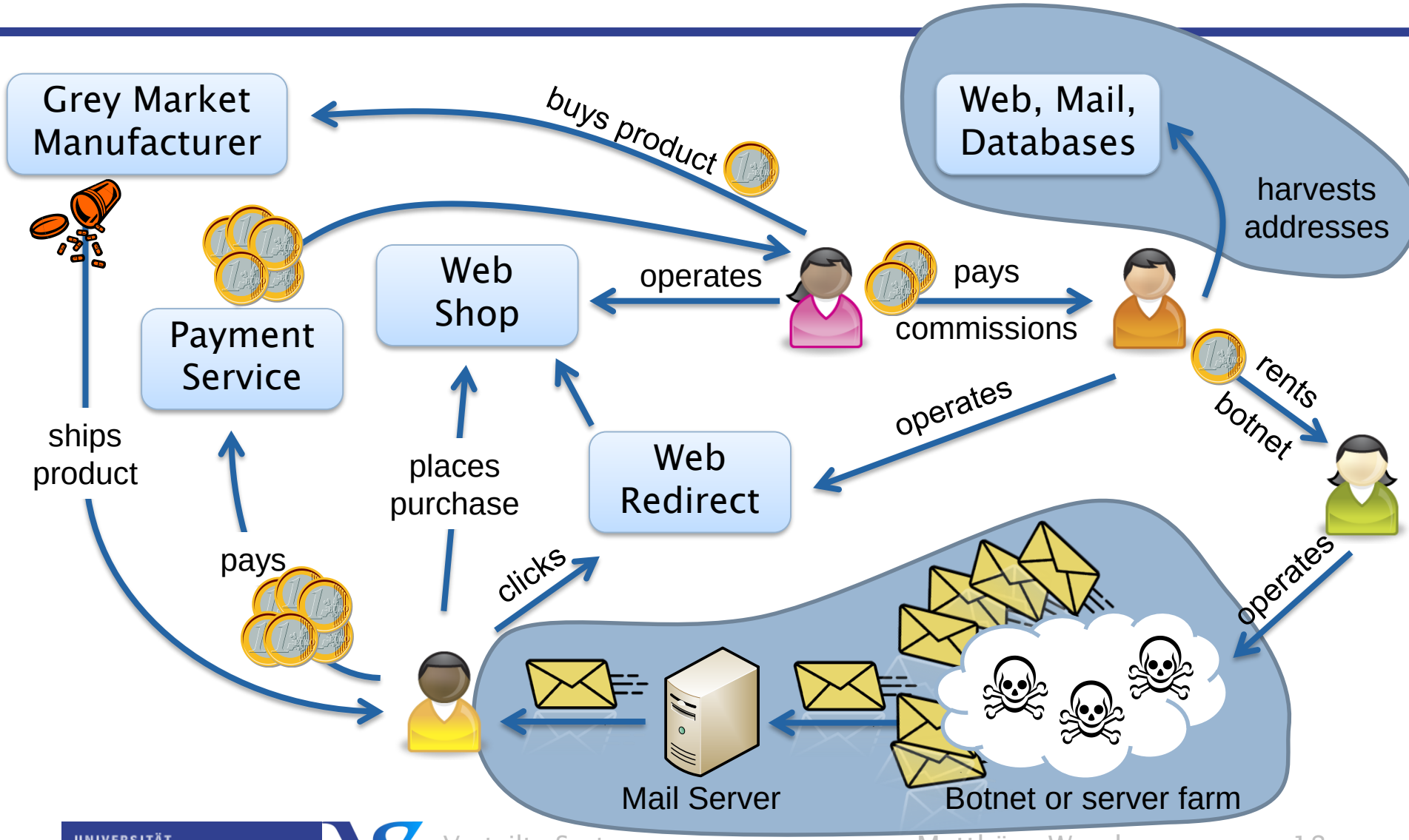
Ecosystem of Commercial Spam



Ecosystem of Commercial Spam

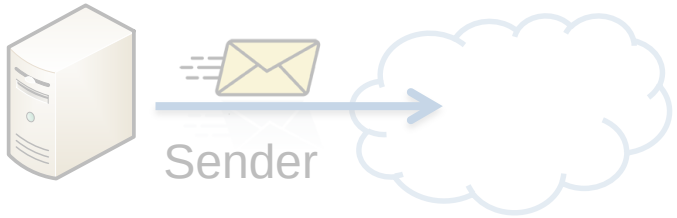


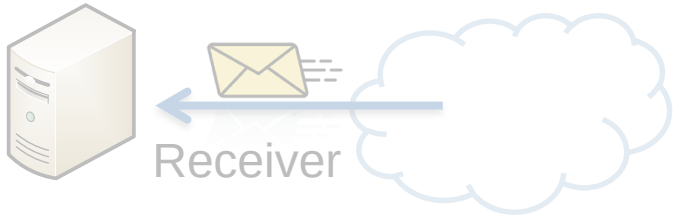
Ecosystem of Commercial Spam



Outline

- Prevent address harvesting
  - Address munging, disposable addresses
- Prevent spam leaving your network
  - Closed relays, port blocking
- Sender helps identifying legitimate email
  - Hashcash, SPF, DKIM
- Slow down sender
  - Challenge-Response, Greylisting, Teergrube
- Receiver-side only detection and filtering
  - Rule-based, statistical, centralized databases

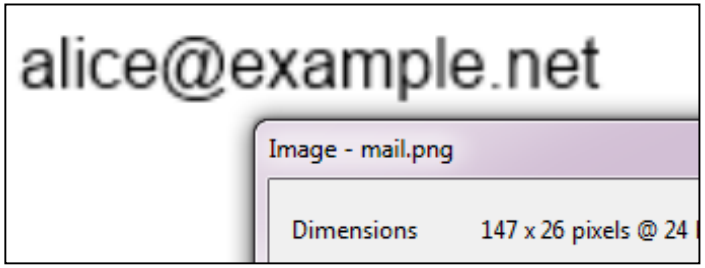




Address Munging

- Spammer crawls the web to retrieve email addresses
  - Also Usenet, WHOIS databases, random domain names, …
- Munge email address to dodge crawlers
  - Try not to annoy users
- Publish email address as image
- Generate address with JavaScript

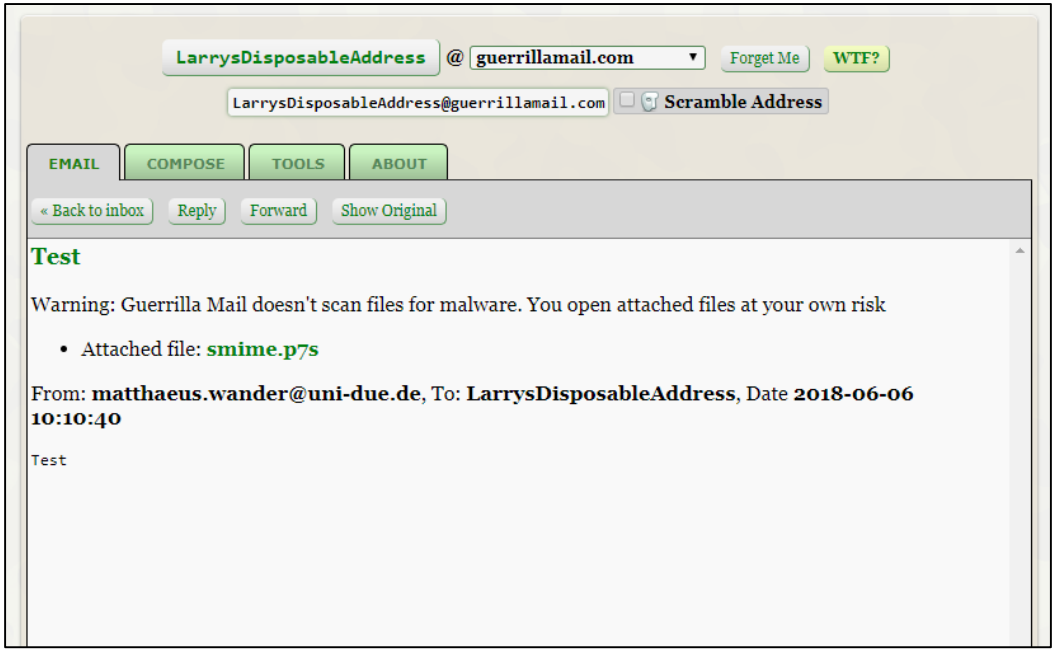
```
alice (at) example (dot) net
alice@exampleNOSPAM.net
```




```
<script type="text/javascript">
 var n = 'alice';
 var at = '@';
 var d = 'example.net';
 document.write(n + at + d);
 </script>
```

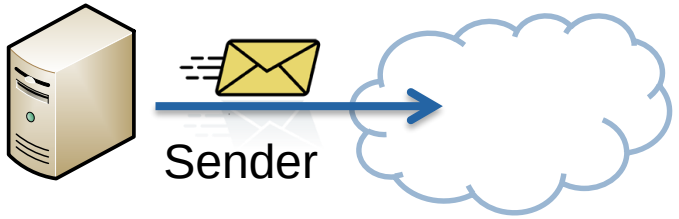
Disposable Address

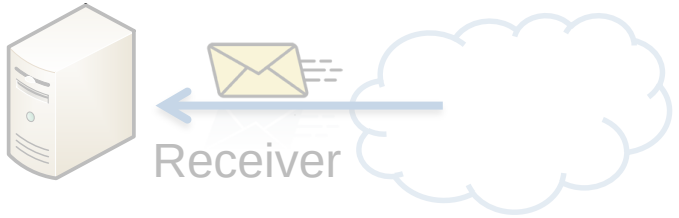
- Create temporary throw-away address
  - Temporary forwards
  - Public retrieval webinterface



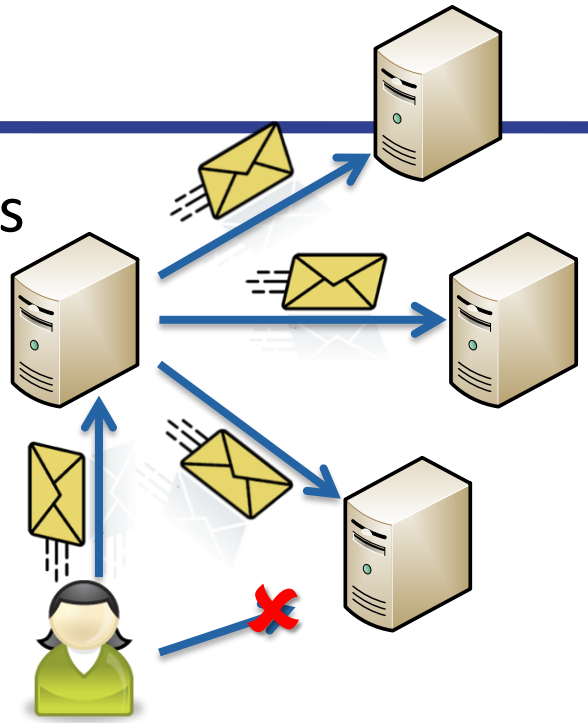
Outline

- Prevent address harvesting
  - Address munging, disposable addresses
- Prevent spam leaving your network
  - Closed relays, port blocking
- Sender helps identifying legitimate email
  - Hashcash, SPF, DKIM
- Slow down sender
  - Challenge-Response, Greylisting, Teergrube
- Receiver-side only detection and filtering
  - Rule-based, statistical, centralized databases





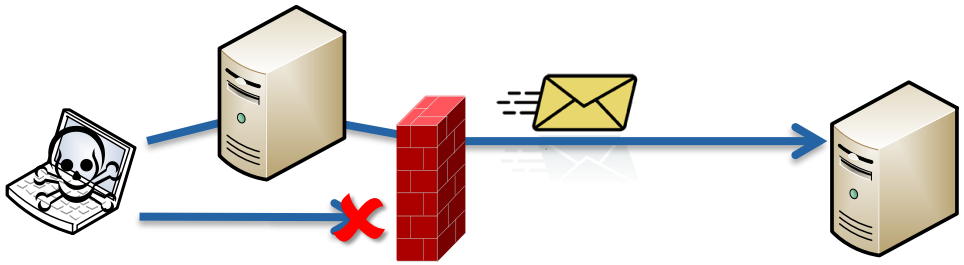
Spam Relays



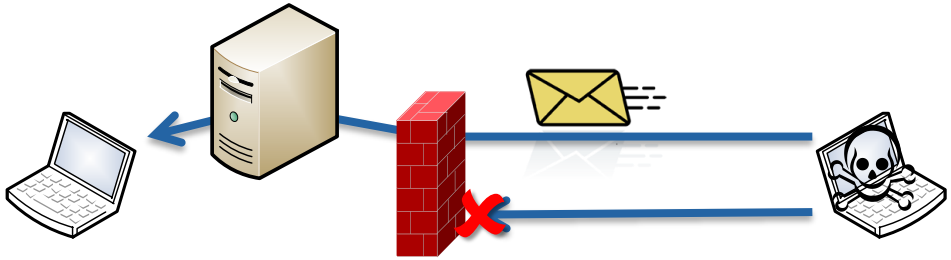
- Mail Transfer Agents (MTA) forward mails
- Open relays forward mails for anyone
- Spammer saves resources
  - Large blind carbon copy list of receivers
- Spammer bypasses blacklists
  - **All** open relays must be blacklisted
- Today‘s mailservers are closed relays in default config
  - Require authentication
- Open proxies also prone to abuse (if misconfigured)
  - SOCKS, HTTPS, TOR (port 25 blocked by default)
- Spam often originates from botnets (infected computers)

Port Blocking

- MTA listens always on TCP/25
- Block outgoing connections from client computers
  - Force clients to use designated MTA for outgoing mails

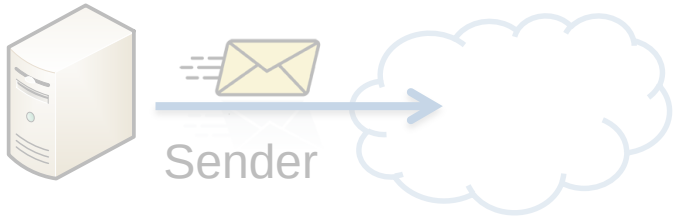


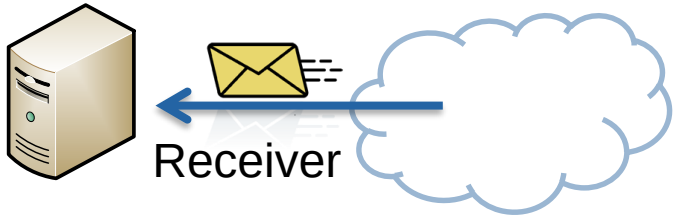
- Block incoming connections to client computers
  - Force remote MTA to use domain MTA for incoming mail



Outline

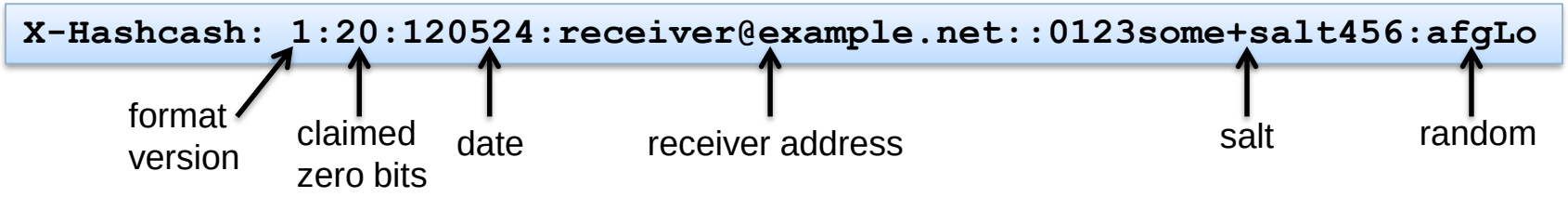
- Prevent address harvesting
  - Address munging, disposable addresses
- Prevent spam leaving your network
  - Closed relays, port blocking
- Sender helps identifying legitimate email
  - Hashcash, SPF, DKIM
- Slow down sender
  - Challenge-Response, Greylisting, Teergrube
- Receiver-side only detection and filtering
  - Rule-based, statistical, centralized databases





Hashcash

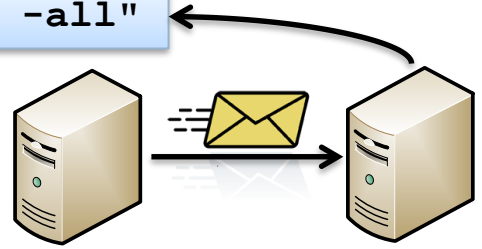
- Idea: associate sending email with cost like a post stamp
  - Resource: computing power
- Find hash = SHA1(date, receiver address, random) where hash ≤ 0x00000FFFFFFFFFFFFFFF
  - Modify random value until you find a matching hash value
- The result serves as proof-of-work
  - Finding 20 binary zeros requires on average 2^{20} hash operations
  - Receiver can verify result with 1 hash operation



Sender Policy Framework (SPF)

- Publish IP addresses of authorized MTAs
  - As TXT record in Domain Name System

```
example.net IN TXT "v=spf1 ip4:192.0.2.0/24 -all"
```



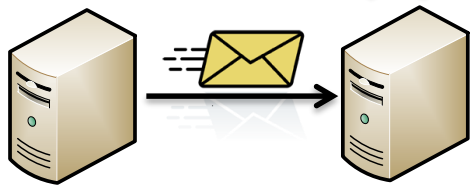
- Receiver gets email from *@example.net
  - Looks up SPF/TXT record of example.net
  - IP address of sending MTA ≙ SPF definition?
- Unauthorized SMTP sender indicates spoofing
  - Beware: SMTP relaying/forwarding is restricted
- MX record for incoming mail, SPF for outgoing mail
  - A related predecessor approach was called „Reverse MX“

DomainKeys Identified Mail (DKIM)

```
DKIM-Signature: v=1; a=rsa-sha256;
   d=example.net; s=brisbane;
   c=relaxed/simple; q=dns/txt;
   l=1234; t=1117574938; x=111[…];
   h=from:to:[…]; bh=MTIzNDz[…];
   b=dzdVyOfAKCdLXdJOc9G2q8L[…];
```

- MTA signs outgoing mail
  - Puts signature in email header
- Public key published in Domain name System
  - Again TXT record, but under particular name

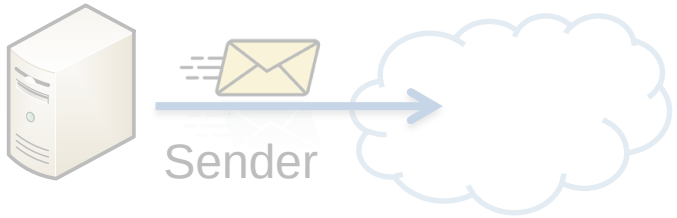
```
brisbane._domainkey.example.net. IN TXT "v=DKIM1; p=MIGfMA0GCSq[…]"
```

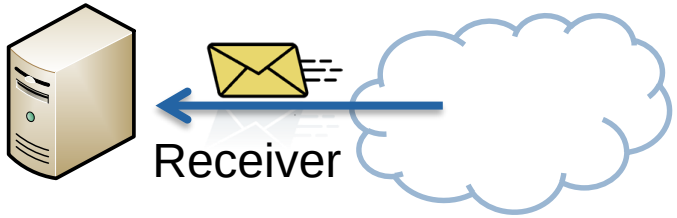


- Receiver gets email from *@example.net
  - Looks up DKIM public key
  - Verifies signature
- Compared to SPF:
  - Slighty more complex (involves public-key cryptography)
  - Survives SMTP relaying/forwarding

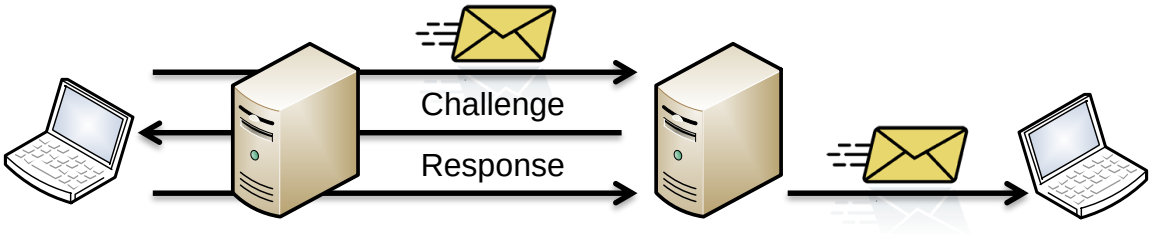
Outline

- Prevent address harvesting
  - Address munging, disposable addresses
- Prevent spam leaving your network
  - Closed relays, port blocking
- Sender helps identifying legitimate email
  - Hashcash, SPF, DKIM
- Slow down sender
  - Challenge-Response, Greylisting, Teergrube
- Receiver-side only detection and filtering
  - Rule-based, statistical, centralized databases





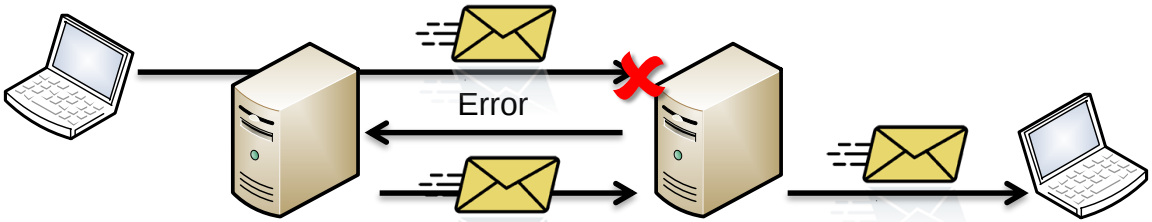
Human Interaction Challenge-Response



- MTA receives email from unknown sender
- MTA delays email and sends challenge
  - E.g. click URL or calculate 5+7
- User solves challenge and sends response
  - Requires manual interaction of sender
  - Spammer is expected to ignore challenge
- Mail is forwarded when response was correct
  - No manual steps for receiver
  - Sender is added to permanent whitelist

Greylisting

- SMTP was built for robustness
- In case of errors, mail is queued for later delivery
  - User warning after 4 hours of delivery errors
  - Give up delivery after 5 days



- MTA receives email from unknown sender
- Save sender in greylist and reply with temporary error
- If sender retries after ≥ 15 minutes, accept email
  - Spammer is not expected to retry later

Teergrube / Tarpit

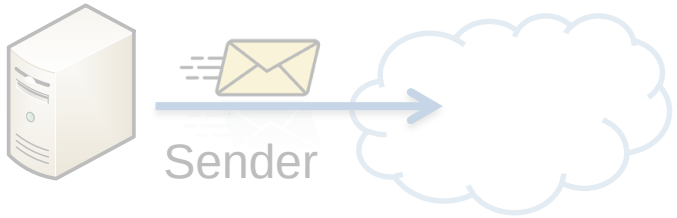
- Spammers send many mails in short time
- Idea: use your resources to slow down spammer
- Delay SMTP responses to keep TCP connection open
  - Send choked multiline response to avoid timeout trigger

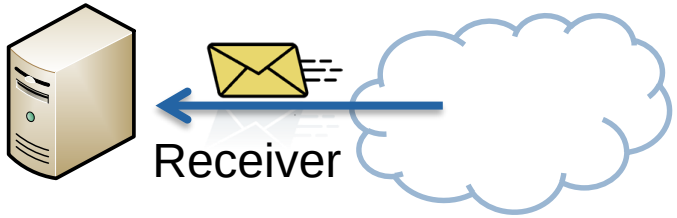
```
451-Well...
451-Give me a moment
451-Just a second
451-Or a minute
451-Or two
451 Error, closing connection
```

- Make sure you don‘t tarpit legitimate MTAs
  - Wastes resources and annoys administrators

Outline

- Prevent address harvesting
  - Address munging, disposable addresses
- Prevent spam leaving your network
  - Closed relays, port blocking
- Sender helps identifying legitimate email
  - Hashcash, SPF, DKIM
- Slow down sender
  - Challenge-Response, Greylisting, Teergrube
- Receiver-side detection and filtering
  - Rule-based, statistical, centralized databases





Rule-based Filtering

- Parse incoming mail for known spam patterns
  - Define rules with regular expressions
- Key words and phrases

```
/\bviagra .{0,25}(?:express|online|overnight)/i
```

  - „fast viagra delivery overnight“
- Disguise attempts
  - „V.A.L.I.U.M“

```
/<inter W1><post P2>(?!valium)<V><A><L><I><U><M>/i
```

  - Unnecessary URL encoding

```
/^https?:\/\/\S*%(?:3\d|[46][1-9a-f]|[57][\da])/i
```

- Errors and patterns unusual for legitimate MUA
  - Malformed headers

```
User-Agent =~ /Mozilla\/5\.0\d\d/
```

  - HTML message without plaintext copy

Bayesian Filter

- Calculate spam probability based on prior statistics
  - Statistics: word occurence in known spam/ham messages
  - Building statistics requires training
- Bayes‘ theorem
  - S: message is spam
  - W: word w occurs
- Example: 100 mails

Word	Spam	Ham
viagra	50	2
…	…	…
Total	80	20

Probability that word w occured when message was spam

Probability that message was spam

$$P(S \mid W) := \frac{P(W \mid S) \cdot P(S)}{P(W)}$$

Probability that message is spam if word w occurs

Probability that word w occured

$$P(S \mid W) = \frac{\frac{50}{80} \cdot \frac{80}{100}}{\frac{50}{80} \cdot \frac{80}{100} + \frac{2}{20} \cdot \frac{20}{100}} = \frac{\frac{50}{100}}{\frac{52}{100}} \approx 0{,}96$$

DNS Blacklists (DNSBL)

- Database of open relays and spammer IP addresses
  - May include ranges of dialin hosts
- Query database via DNS
  - Is 192.0.2.1 listed as spammer IP?

```
1.2.0.192.dnsbl.inps.de IN A 127.0.0.2
```

  - Yes = A record 127.0.0.2
  - No = No such name error (NXDOMAIN)
- Efficient query interface (DNS over UDP)
- Various different DNSBL providers
  - Trust required: false listing can cut off your email traffic
- DNS Whitelists also common for large email providers

Hash Value Databases

- Database of email spam hash values
  - e.g. Distributed Checksum Clearinghouse (DCC)
  - e.g. Vipul‘s Razor, Pyzor
- Calculate hash value over message body (not headers)
- Query database via UDP or pipelined TCP connection
- Hash buster: add random data to modify hash value
- Fuzzy hashing: preserves similarities in hash value
  - Results in match score
- Ephemeral hash: hash over different message part
  - Based on changing random number

Filtering Actions

```
X-Spam-Checker-Version: SpamAssassin 3.3.1 (2010-03-16) on whitespace.swznet.de
X-Spam-Flag: YES
X-Spam-Level: **********
X-Spam-Status: Yes, score=16.6 required=5.0 tests=BAYES_50,FSL_HELO_NON_FQDN_1,
    HELO_LOCALHOST,HTML_IMAGE_ONLY_20,HTML_MESSAGE,NIX_SPAM,RCVD_IN_BRBL_LASTEXT,
    RCVD_IN_PBL,RCVD_IN_PSBL,RCVD_IN_SORBS_DUL,RCVD_IN_SORBS_WEB,RCVD_IN_XBL,
    RDNS_NONE,SPF_FAIL,T_REMOTE_IMAGE autolearn=spam version=3.3.1
```

- Use combination of different measures
  - Calculate probability/score per email
- Take actions, depending on score
  - Deny email with error
  - Accept email but drop silently
  - Move to spam folder
  - Fall back to greylisting, challenge/response or teergrube
- Beware of false positives

Summary

- Prevent address harvesting
  - Address munging, disposable addresses
- Prevent spam leaving your network
  - Closed relays, port blocking
- Sender helps identifying legitimate email
  - Hashcash, SPF, DKIM
- Slow down sender
  - Challenge-Response, Greylisting, Teergrube
- Receiver-side only detection and filtering
  - Rule-based, statistical, centralized databases