

---

# Internet Services Security

## Internet Security [1] VU

Paolo Milani Comparetti

[pmilani@seclab.tuwien.ac.at](mailto:pmilani@seclab.tuwien.ac.at)

Clemens Kolbitsch

[ck@seclab.tuwien.ac.at](mailto:ck@seclab.tuwien.ac.at)

# Announcements

---

- Challenge 3 is over
  - 34 people have completed
  - First was BALINT "System Grapefruit " FLORIN (1.5 hours)
- Challenge 4 starts today
  - yet another web challenge!
  - hack Fakebook using XSS!
  - first challenge that requires a little bit of programming (in Javascript)
  - running until 12.05 (3 weeks!)
- Next week, Clemens Kolbitsch on Testing

# Internet Services Security

---

- You had 2 lectures on Web Application Security
- Only this 1 lecture on the rest of the internet
  - quick overview of a number of application layer protocols
  - a case study + demo (DNS cache poisoning)
- Focus is on (in)security of network protocol
  - not on specific vulnerabilities in a server implementation
  - the problem is in the protocol design!

# Internet Services

---

- Internet infrastructure
  - traditional (old) services
  - emerged to satisfy needs from the beginning of the Internet
  - provides basic services that other applications rely upon
  - often no or little security in mind
  - can cause indirect security problems, or prepare and enable attacks against target application
- Particular services
  - remote access (telnet, rservices)
  - name resolution (DNS)
  - file transfer (FTP)
  - mail transfer (SMTP)
  -

# Remote Access

---

- telnet, rlogin
  - horrible security
    - plaintext passwords
    - connection hijacking (hunt)
  - fortunately, it is virtually not used anymore
- ssh
  - secure replacement
  - ssh version 1
    - insecure because of possibility to insert data into remote stream
  - ssh version 2 is current, and should be used

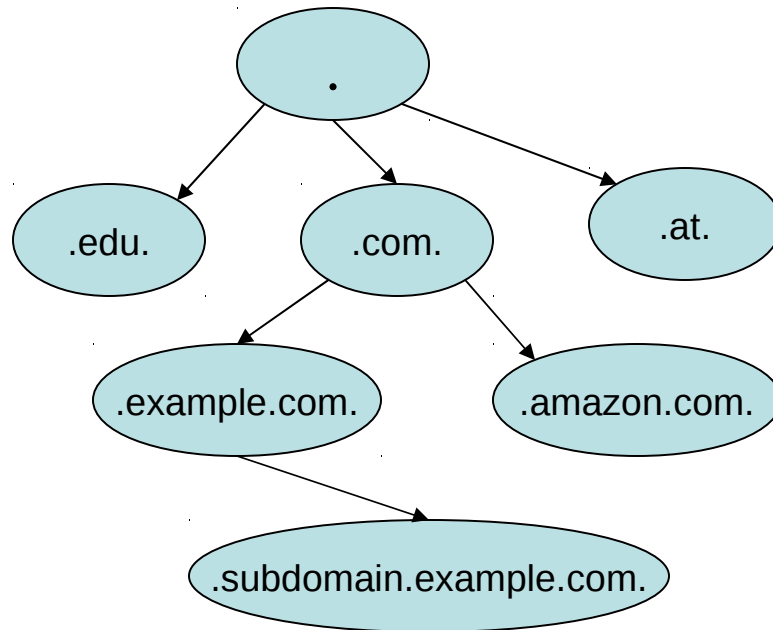
# DNS: Domain Name Server

---

- initially specified in RFC 1034/1035
- distributed database that maps **domain names** into **IP addresses** and vice versa
- name space is hierarchically divided in domains
- each domain is managed by a name server
- uses mostly UDP
- sometimes TCP for long queries and TCP for zone transfers between name servers

# DNS

---



# Name Server

---

- Name servers are responsible for mapping names of a domain
  - example
    - subdomain.domain.com is managed by dns.subdomain.domain.com
    - domain.com is managed by master.domain.com
- Root name servers
  - 13 machines distributed around the world
  - associated with the top level of the hierarchy
  - dispatch queries to the appropriate domains
- Server types
  - primary (authoritative for the domain, loads data from disk)
  - secondary (backup servers, get data through zone transfers)
  - caching-only (relies on other servers but caches results)
  - forwarding (simply forwards query to other servers)

# Name Server

---

- A server that cannot answer a query forwards the query up in the hierarchy
- Then, the search follows the correct branch in the hierarchy down to the authoritative server
- The results are usually maintained in a local cache
- Reverse lookup
  - mapping from IP addresses to names
  - also called pointer queries
  - use dedicated branch in name space starting with ARPA.IN-ADDR
  - example
    - if 128.131.172.79 is resolved, this is mapped into 79.172.131.128.in-addr.arpa

# DNS Clients

---

- At least one name server has to be specified
  - e.g., Linux uses `/etc/resolv.conf`
- Queries can be
  - recursive
    - require a name server to find the answer to the query itself
  - iterative
    - instead of the resolved name another server's address is returned, which can be asked
- Lookup can be performed with
  - `nslookup`, `host`, `dig`

# DNS Data

---

- same message format for requests and replies (binary)
- contains questions, answers, authoritative information
- DNS data is structured in Resource Records, which store the information.
- Different types of RR exist:
  - A defines an IP address for domain name
  - HINFO host information (CPU, OS)
  - NS authoritative name server for domain
  - MX mail server for domain
  - TXT human-readable information

# DNS Security Issues

---

- Daemon vulnerabilities
  - BIND named has a bad security history
- DNS often provides rich information
  - IP addresses
  - HINFO records
  - WKS (Well Known Servers)
  - can be gathered via exhaustive queries or via zone transfers
  - IP scanning is not necessary in many cases
- Simple DNS spoofing
- DNS **cache poisoning**

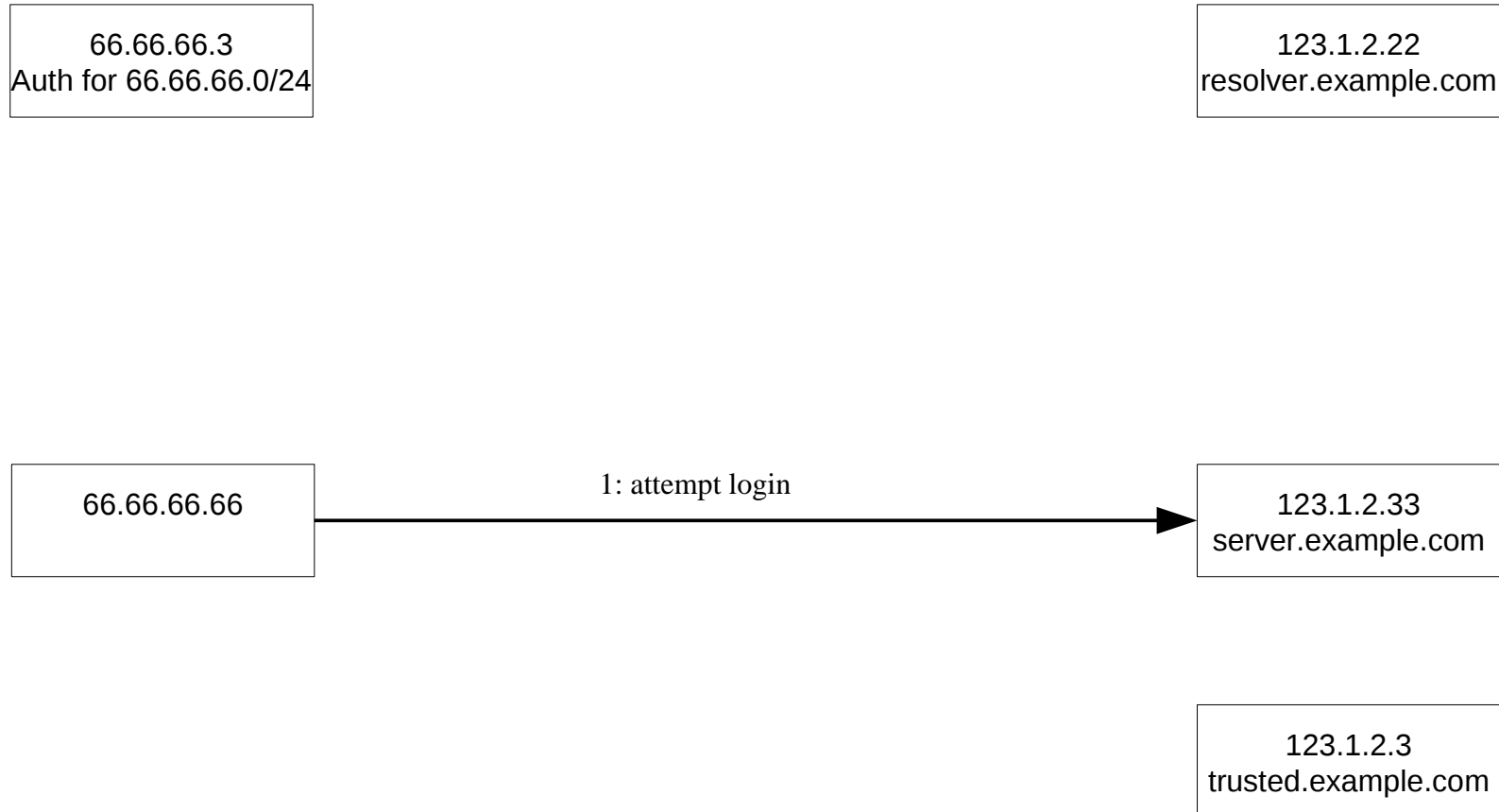
# Simple DNS Spoofing

---

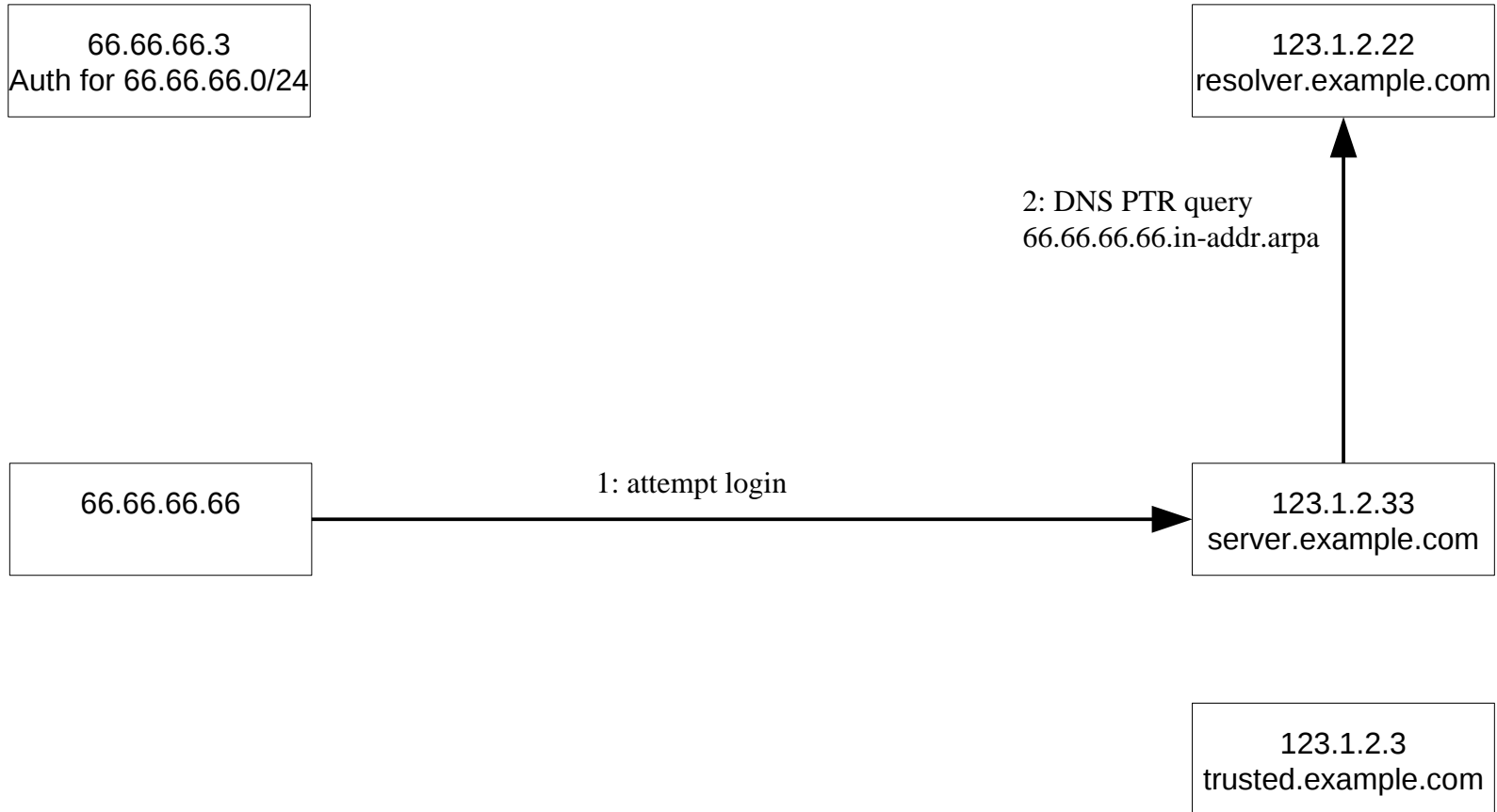
- Used when authentication is performed based on DNS names with reverse lookup
  - e.g., `trusted.example.com` may login using `rlogin` without specifying a username/password
  - or, only `trusted.example.com` may login at all
- Concept
  - domain name is obtained through reverse DNS query
  - a DNS query is forwarded to the authoritative DNS server for the IP address that logs in (under control of the attacker)
  - this DNS server replies with the (faked) trusted name

# Simple DNS Spoofing

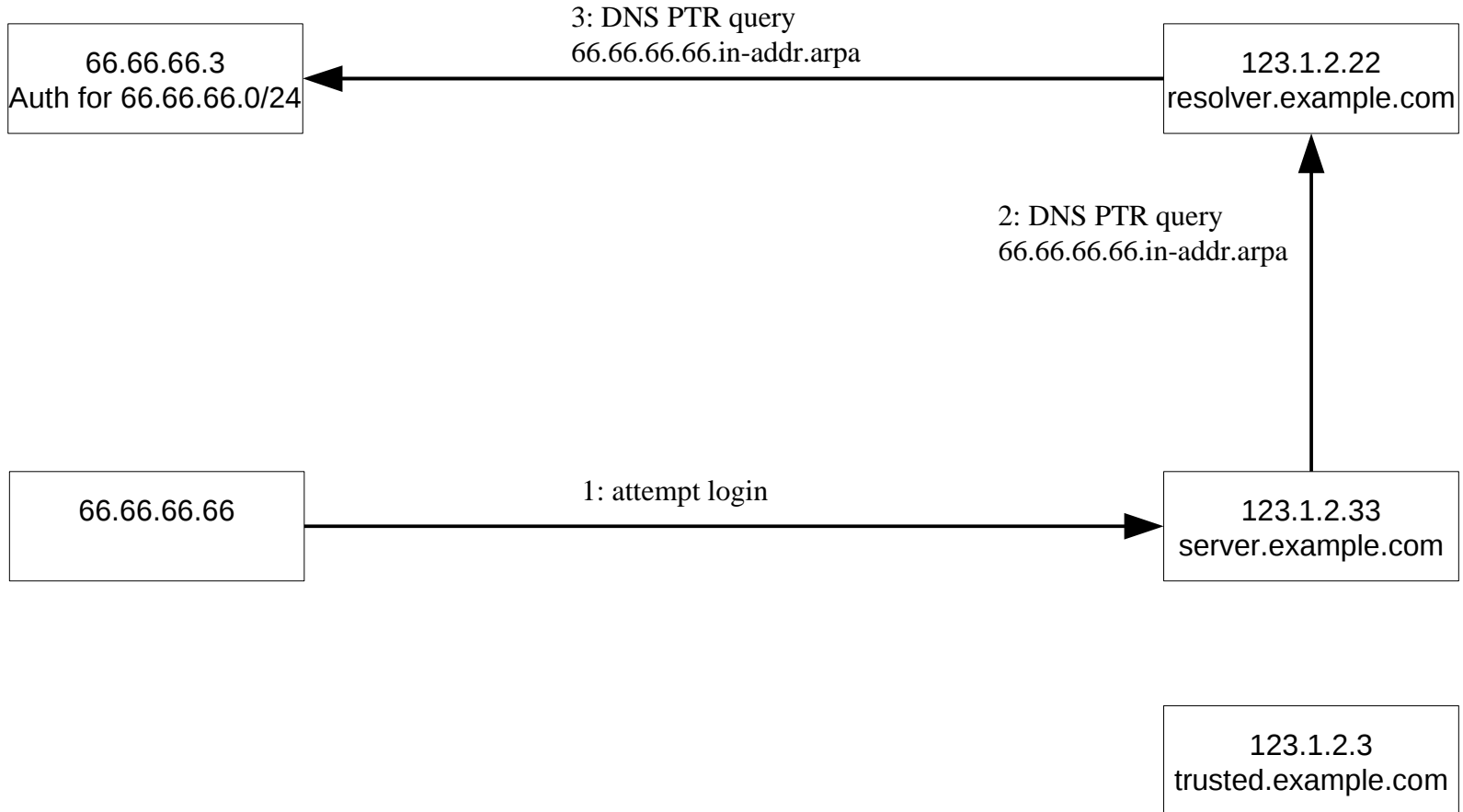
---



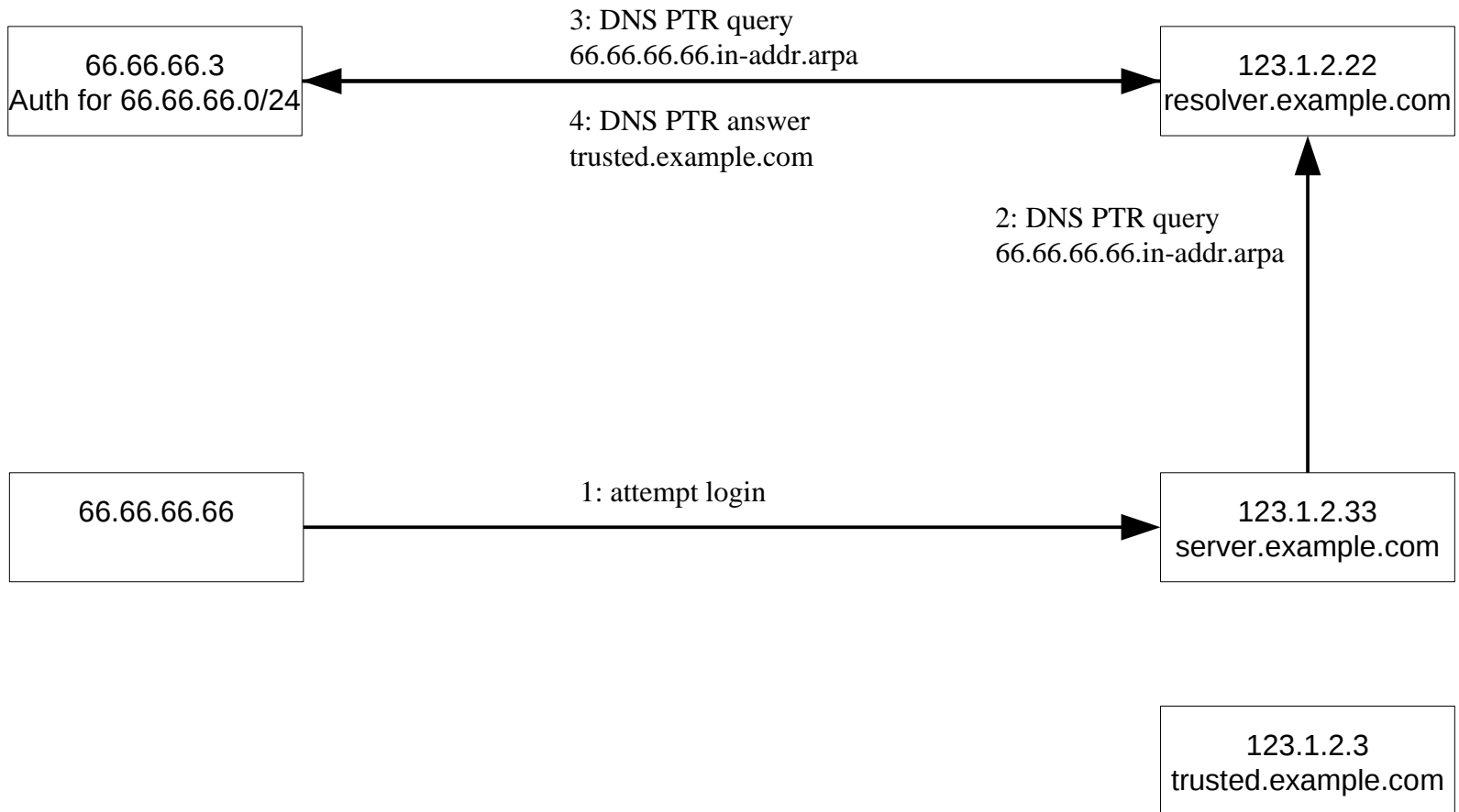
# Simple DNS Spoofing



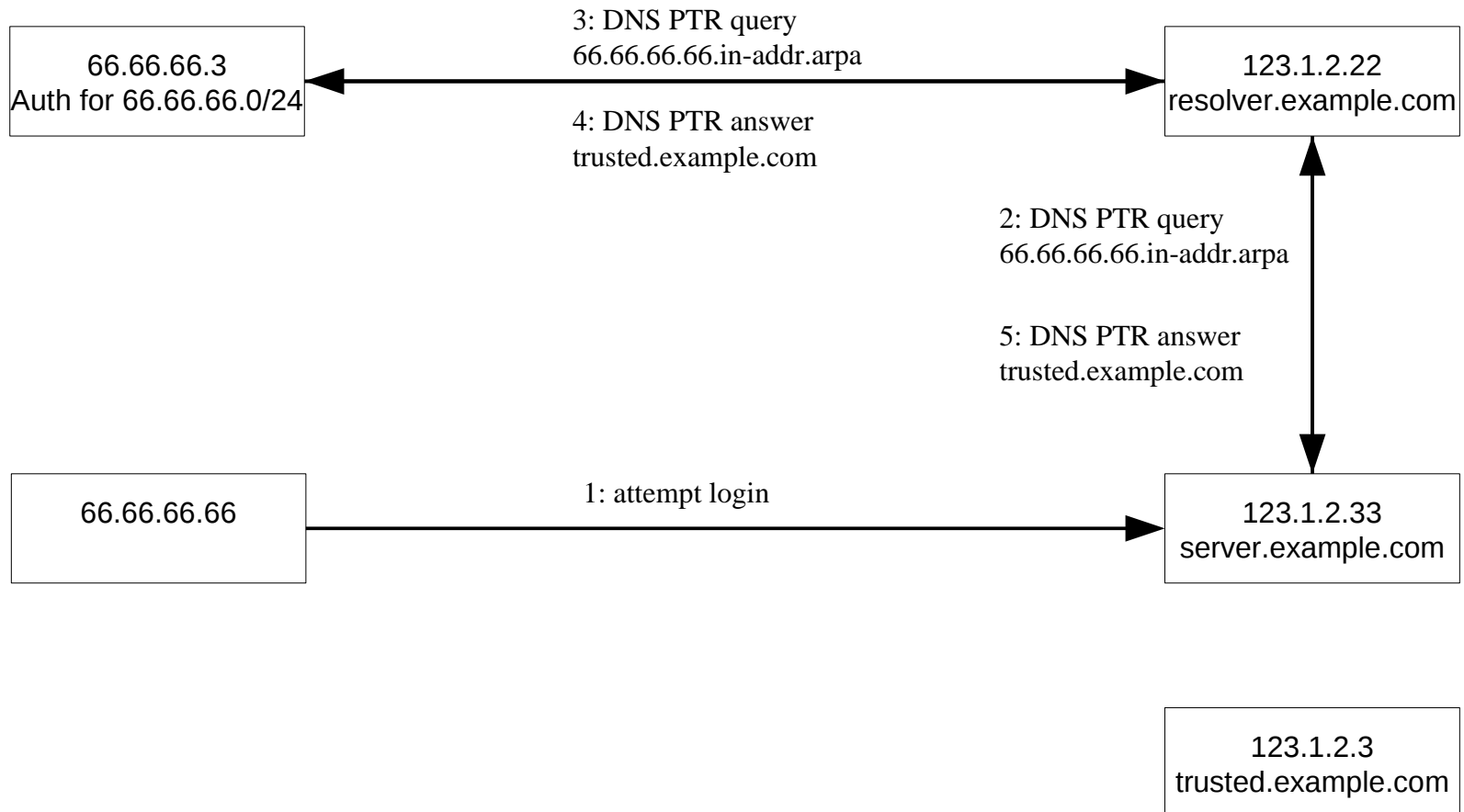
# Simple DNS Spoofing



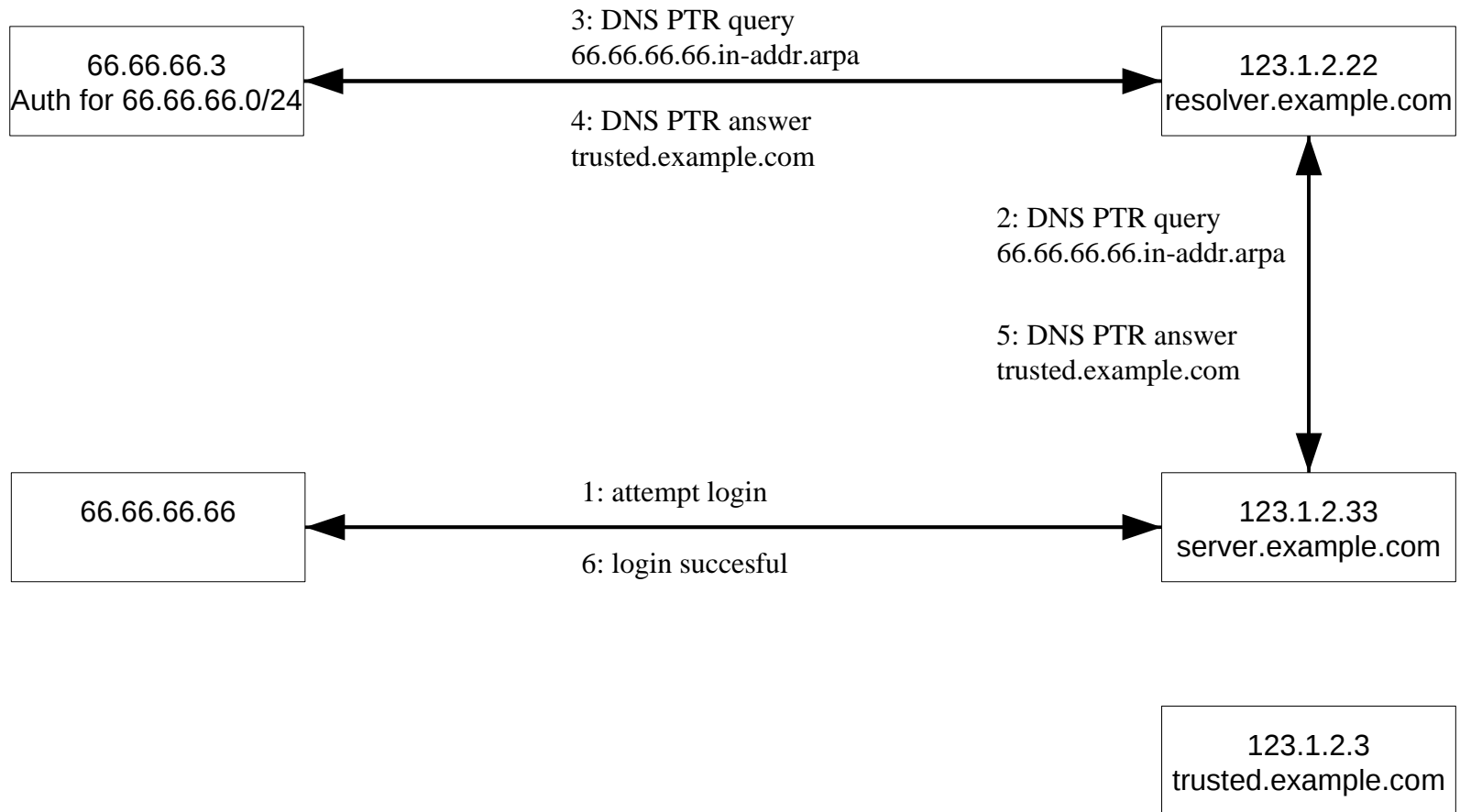
# Simple DNS Spoofing



# Simple DNS Spoofing



# Simple DNS Spoofing



# Simple DNS Spoofing

---

- Countermeasure
  - use double reverse lookup
  - reverse lookup 66.66.66.66 => trusted.example.com
  - now do forward lookup for trusted.example.com => 123.1.2.3
  - refuse login!

# DNS Cache Poisoning

---

- DNS requests/replies normally sent over UDP
- Reply from server is **not authenticated**
- Attacker can spoof replies, answering incorrect data
- Respond faster than legitimate server
- It is possible to perform DNS Hijacking by
  - racing with the server with respect to a client
  - racing with a server with respect to another server

# Spoofed DNS reply

---

- To be accepted, the spoofed replies must match a query!
- Use correct (spoofed) source IP address of the real server
- Use correct destination UDP port (the source port from which the query was sent)
- Answer correct query
- Correct value of DNS nonce field
  - 16 bit, randomly selected request id

# DNS Cache Poisoning

---

- Attack a caching-only server
- Send a request for host.example.com
  - Server will send request to authoritative NS for example.com
- Immediately send many spoofed replies
  - Source IP is one of the NS for example.com (~4 options)
  - Guess destination UDP port (16 bit)
  - Guess DNS nonce (16 bit)
- Number of replies needed on average:
  - $2^{16} * 2^{16} * 4 / 2 = 2^{33} \approx 8$  billion
- Need multi-terabit/s pipe to do it in 1 second (before real reply)

# Improving the Chances

---

- Src port known
  - many servers always send queries from same UDP port
    - bind up to ~9.4.2 chooses port at startup
  - attacker can find it out by setting up authoritative server for his own domain, and querying for that domain
  - now only need about  $2^{16} * 4 \approx 250000$  attempts
- Birthday attack
  - some servers allow multiple outstanding requests for same domain
  - send 100 queries+100 answers $\approx$ 10000 chances of guessing
- But still, If you lose the race, can't retry until cache expires (~1 week)

# Restarting the Race

---

- Trick published by Dan Kaminsky in 2008
  - big news coverage
- concerted patching effort
  - randomize SRC port of queries
  - avoid multiple outstanding queries for same domain
- Send requests for random, non-existent subdomains of target
  - spoof reply for random domain
  - additional authoritative NS section in reply, giving a new address for the authoritative NS for the domain

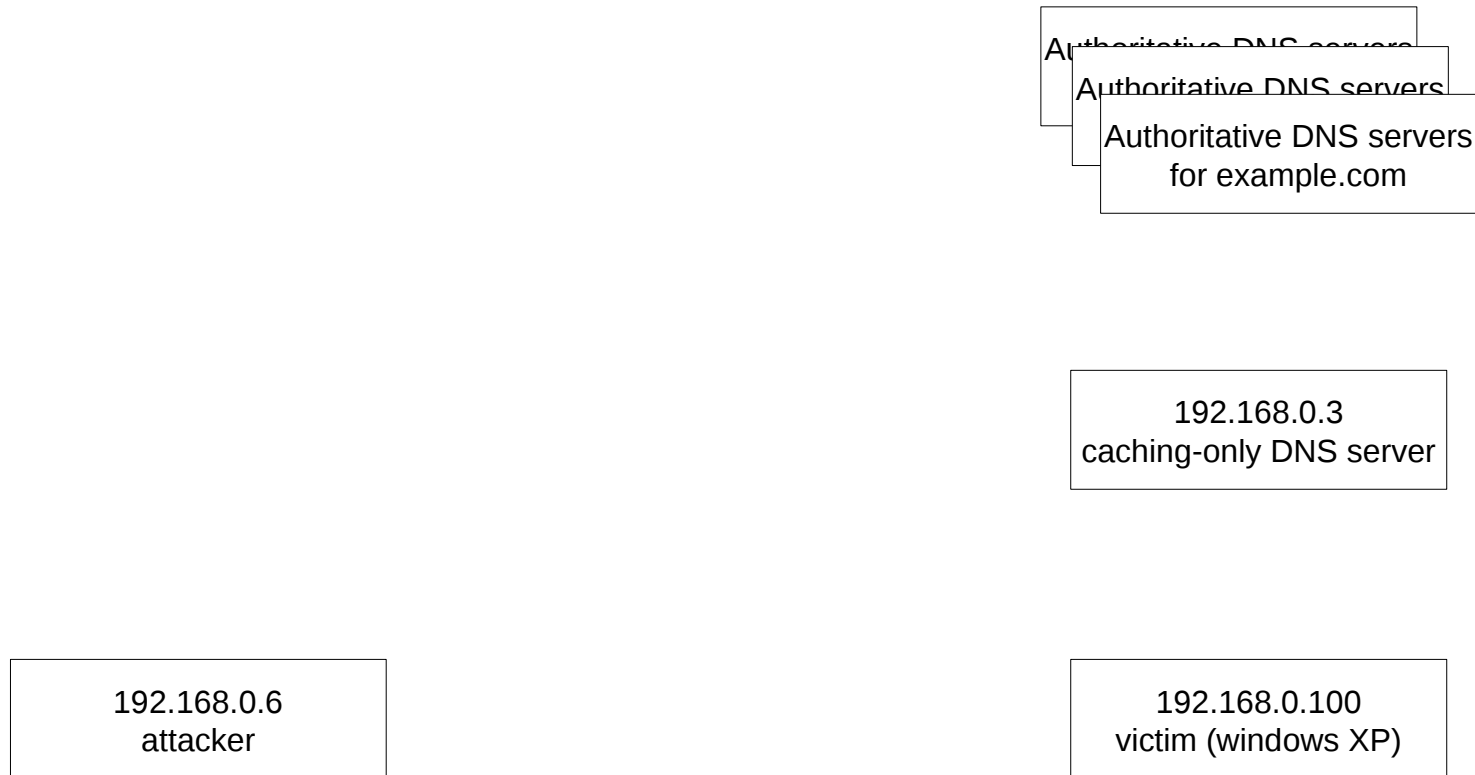
# Restarting the Race

---

- To poison `www.example.com`:
- Query for `fdkajfdksdfj.example.com`
  - send spoofed replies for `fdkajfdksdfj.example.com`
  - reply also holds "authoritative nameservers" section:
    - NS for `example.com` is `www.example.com`
  - also holds "additional records" section:
    - `www.example.com` is `66.66.66.66`
- if you lose the race, try again with `djdkdkdkdk.example.com`

# DNS Cache Poisoning Demo

---



# DNS Cache Poisoning Demo

---

Authoritative DNS servers  
Authoritative DNS servers  
Authoritative DNS servers  
for example.com

Let's start the Demo!

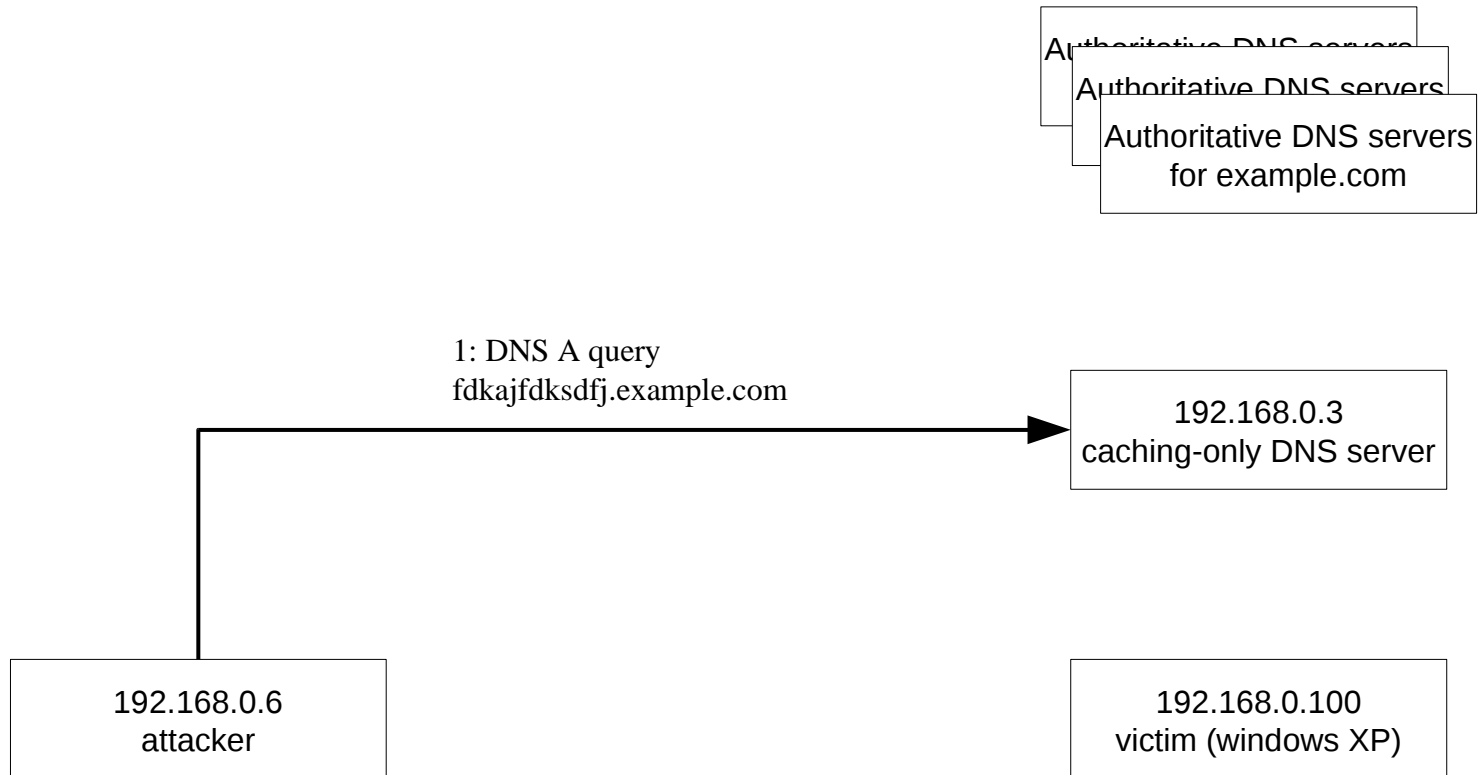
192.168.0.3  
caching-only DNS server

192.168.0.6  
attacker

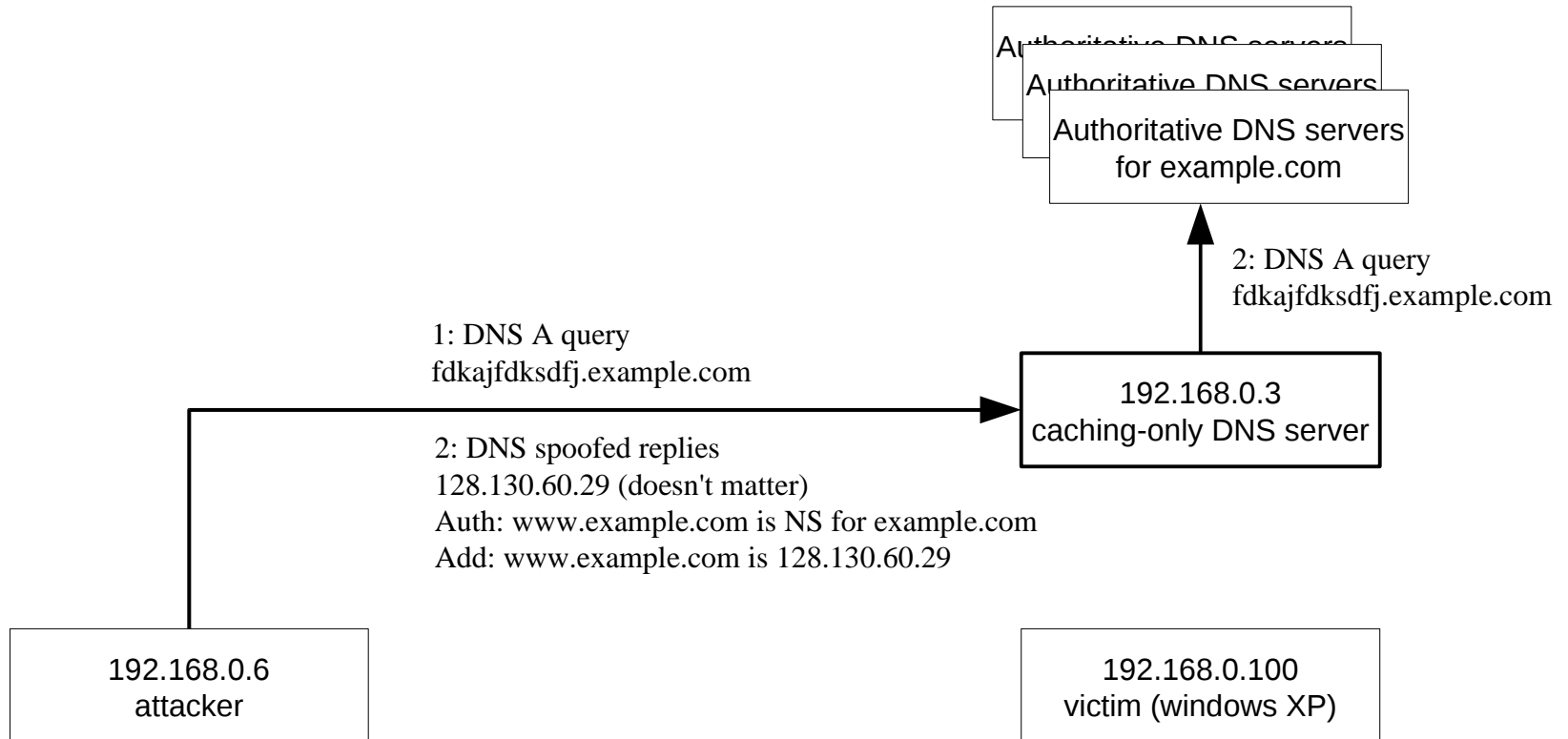
192.168.0.100  
victim (windows XP)

# DNS Cache Poisoning Demo

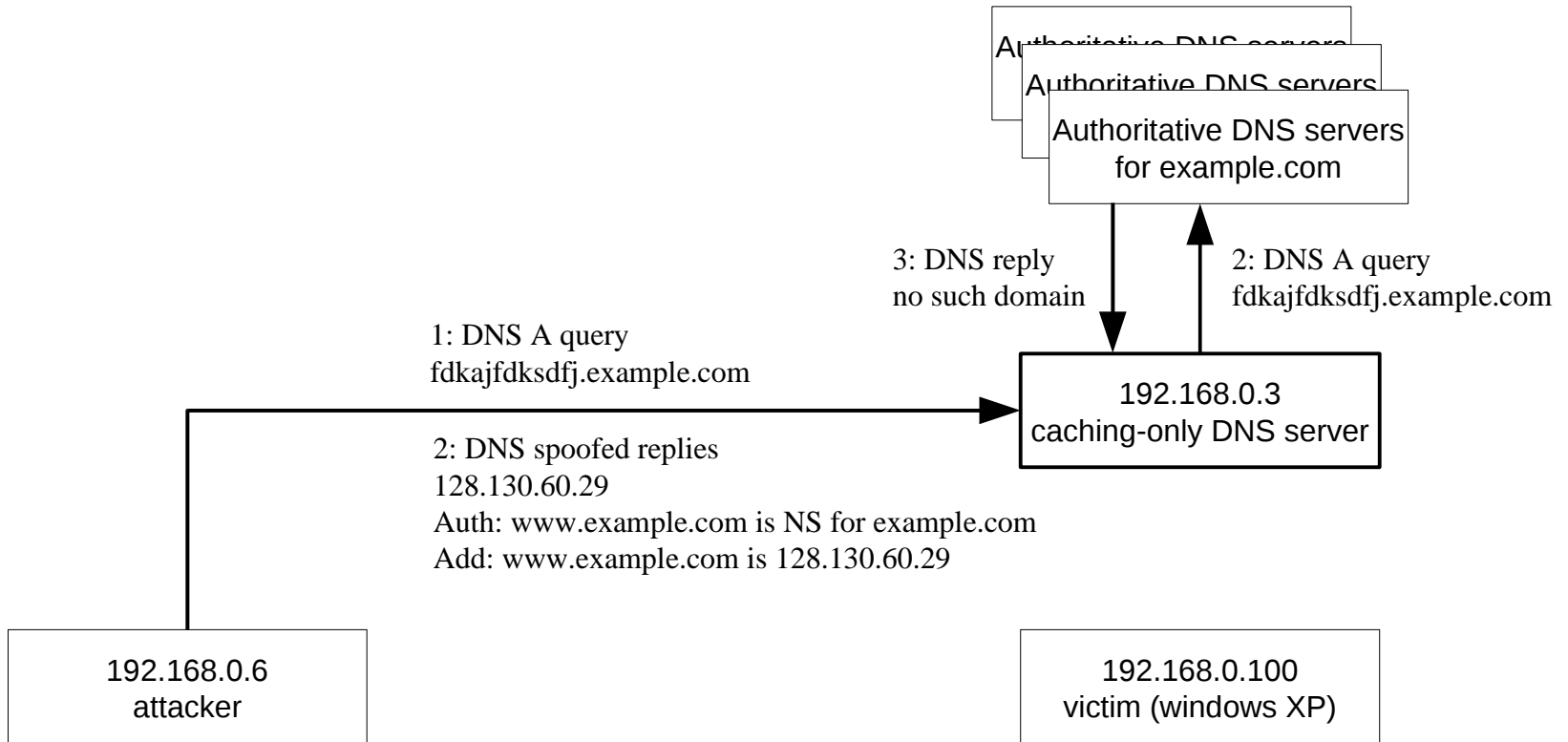
---



# DNS Cache Poisoning Demo



# DNS Cache Poisoning Demo



# DNS Cache Poisoning: Effects

---

- Redirect traffic
- Denial of Service
- MITM attack with no physical access
- Redirect email (MX record...)
- Exploit auto-update
  - java updater uses no crypto: just need to poison java.sun.com
    - metasploit evilgrade module
    - video demo: <http://www.infobyte.com.ar/demo/evilgrade.htm>

# DNS Cache Poisoning: Countermeasures

---

- Check the DNS server(s) you use use random src ports
  - sniff outgoing queries traffic (often not possible)
  - tool at [www.doxpara.com](http://www.doxpara.com) (DOWN now)
  - run a NS for your own domain, make a recursive query and sniff incoming packets
- Block queries to your recursive resolvers from outside your network
- DNSSEC:
  - authoritative replies are cryptographically signed
  - not yet deployed on DNS root zone
  - deployed on a few TLDs

# FTP: File Transfer Protocol

---

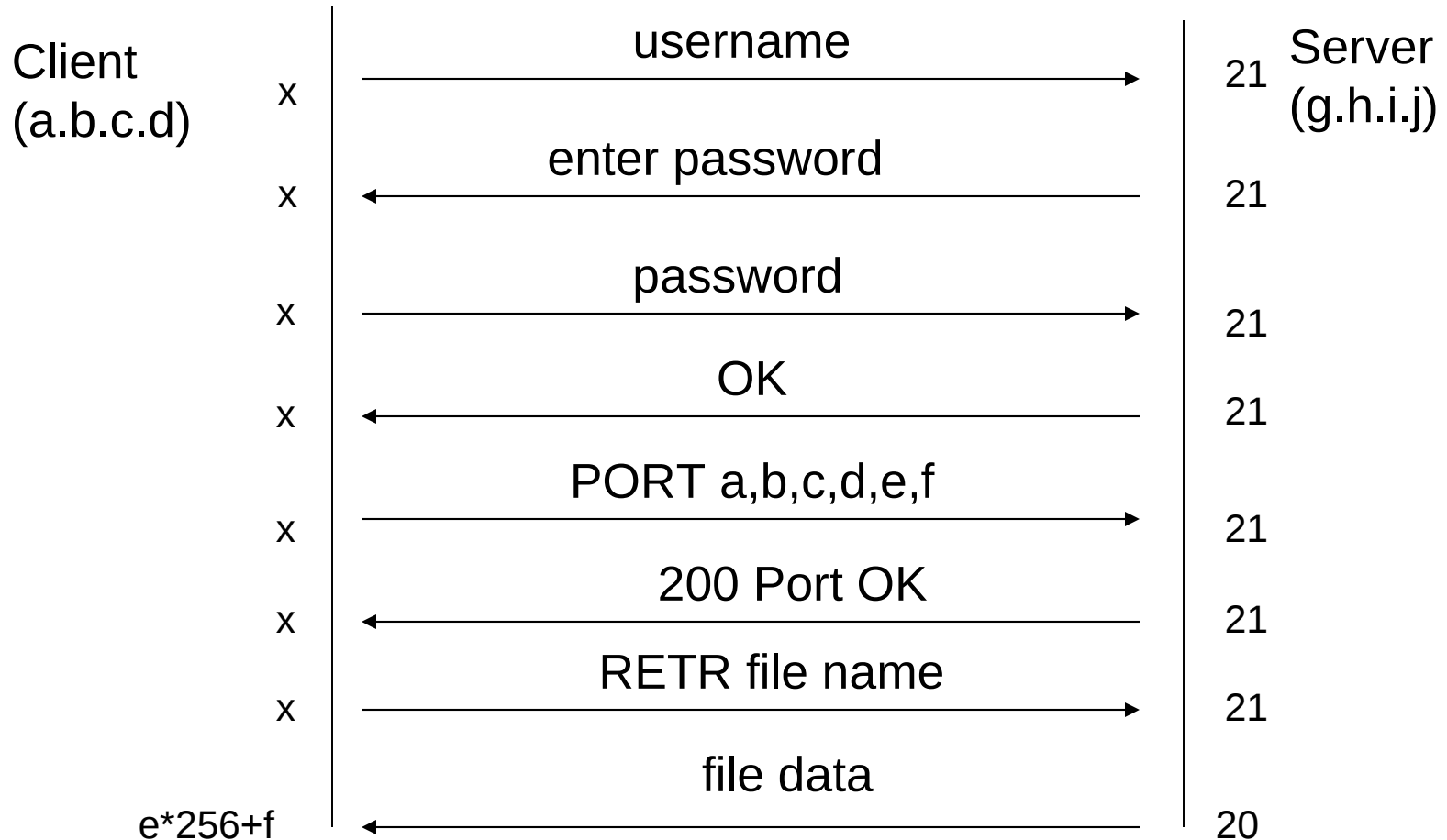
- initially specified in RFC 542
- provides file transfer service
- based on TCP
- client / server architecture
  - client (ftp) sends a connection request to the server (ftpd)
  - server is listening on port 21
  - client provides username and password to authenticate
  - client uses the GET and PUT commands to transfer files

# FTP

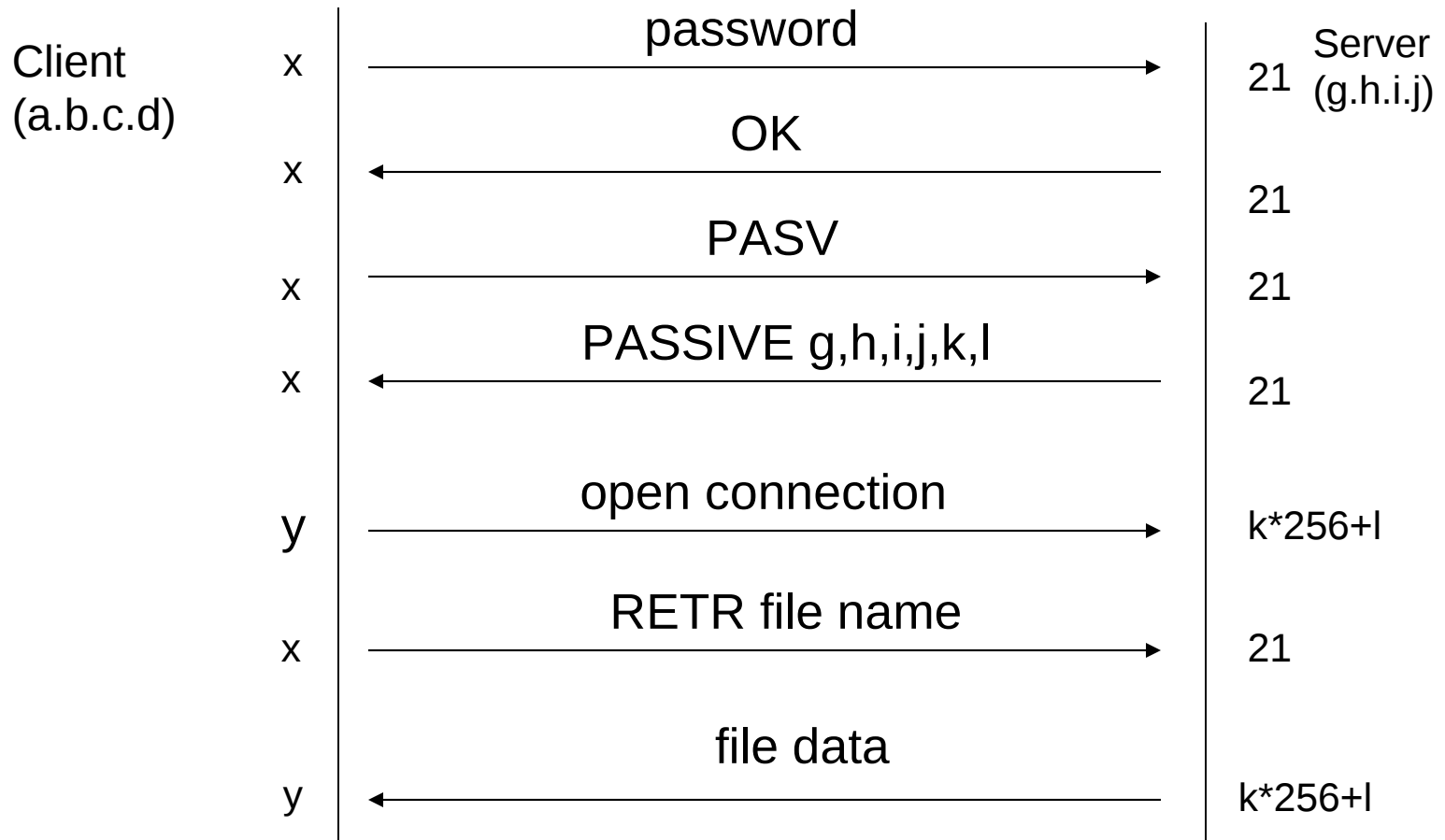
---

- 2 TCP connections are used
  - control stream for commands
  - data stream for the actual data that is transmitted
- Client tells the server to connect to one of its local ports using the PORT command
- Server opens a connection from port 20 to the port specified by the client
- Transfer is executed and the connection is closed

# FTP Protocol



# Passive FTP



# FTP Security

---

- Server implementation vulnerabilities
- Configuration errors
  - allow "anonymous" user to write files
  - write to user home directory
    - Can be abused to write files into home directories that are normally used for authentication (e.g. `.ssh/id_dsa`)
    - If an anonymous user is allowed to put such a file in the home directory he can get access to the computer, using private key authentication

# PASV Connection Theft

---

- Attacker can connect to port that was opened by server before the legitimate client does
- Since the command connection is still established, client commands lead to file transfers between attacker and server

# FTP Bounce

---

- The `PORT` command is used by the client to tell the server the address and port to be used when opening a data connection
- According to the RFC 959 the address does not have to be the same as the one the client has
  - idea is to allow transfers between two hosts without having to go through the client
- Therefore it is possible to instruct a server to open a connection to a third host

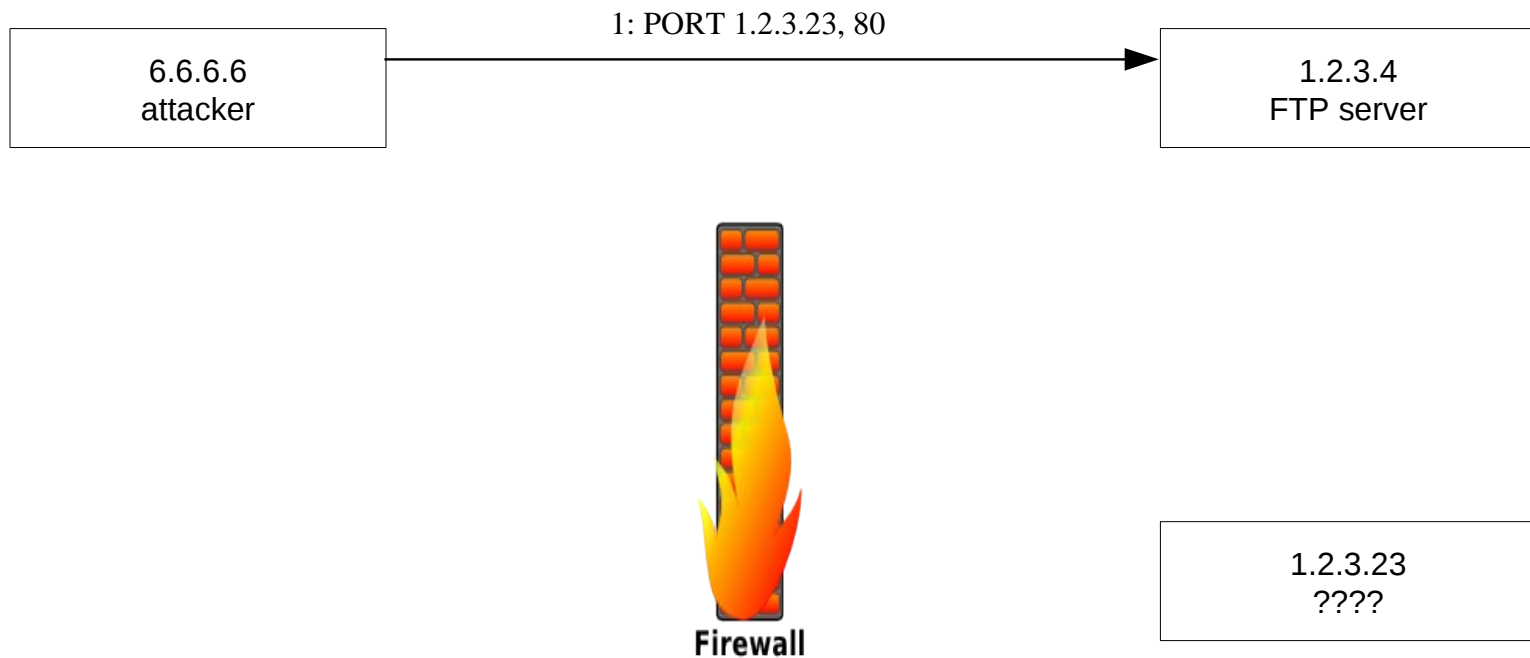
# FTP Bounce

---

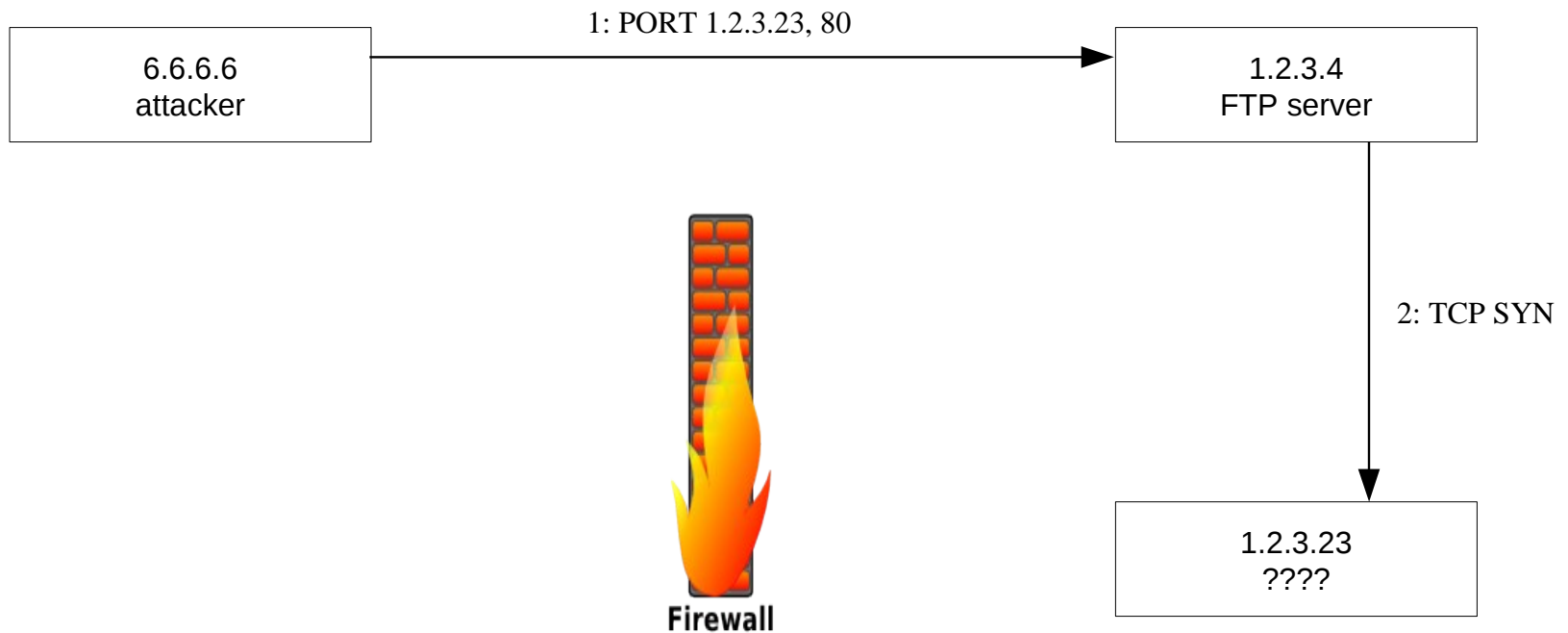
- Can be used to perform a TCP portscan
- The host running ftpd appears to be the source of the scan
- It is possible to scan „behind“ a firewall
  - suppose that only port 21 and 20 are open at the firewall

# FTP Bounce: Port Scan

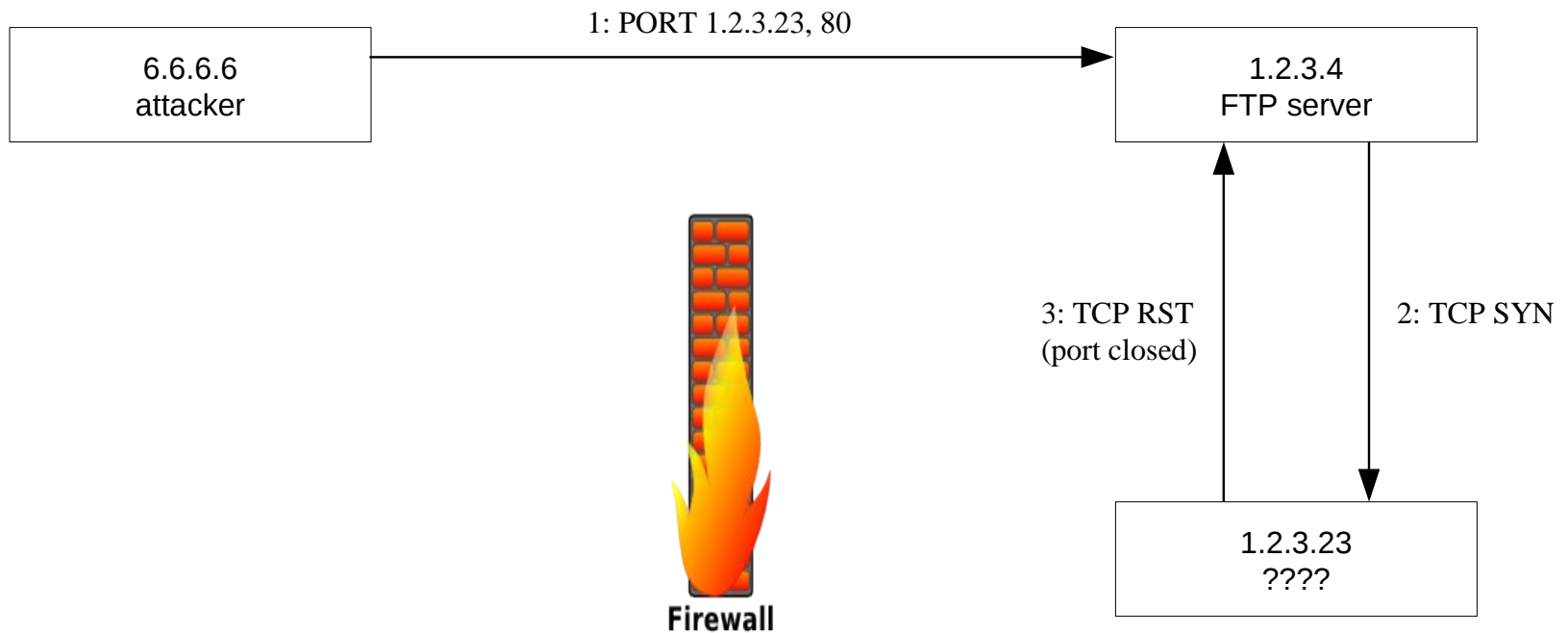
---



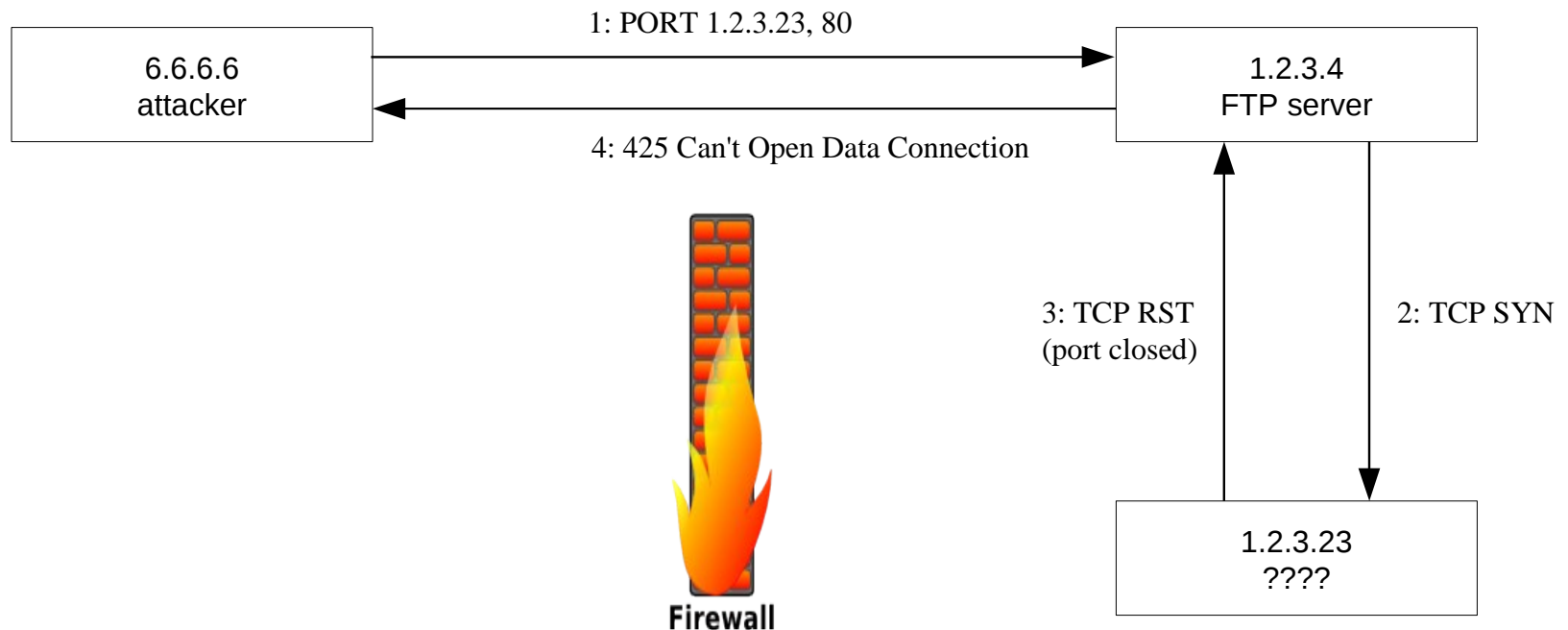
# FTP Bounce: Port Scan



# FTP Bounce: Port Scan

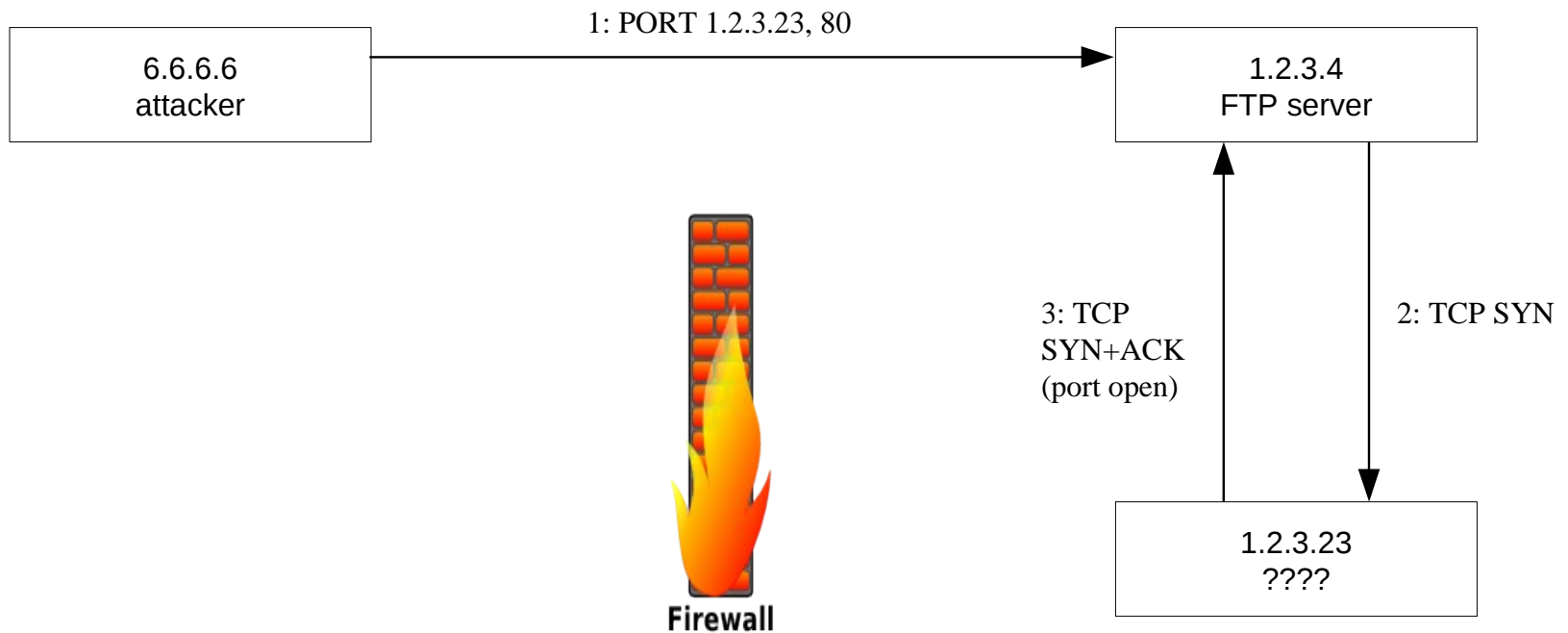


# FTP Bounce: Port Scan

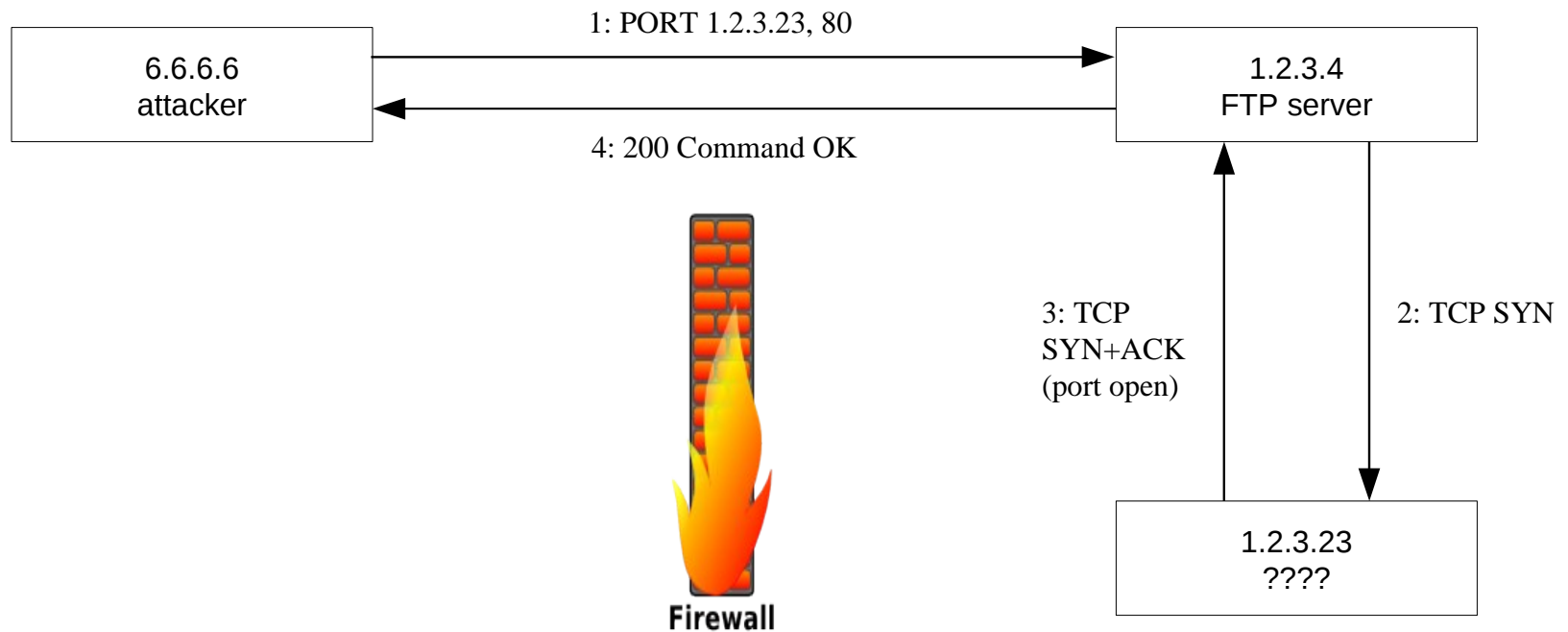


- 1.2.3.23 does not have an HTTP server running on port 80

# FTP Bounce: Port Scan



# FTP Bounce: Port Scan



- 1.2.3.23 has a server running on port 80!

# FTP Bounce

---

- Not only useful for port scans: can be used to send data to arbitrary ports
  - if an FTP writable directory exists, arbitrary data can be sent to a third host
  - can be used to bypass restrictions (IP based authentication)
  - connection laundry
- Step 1:
  - upload data to the ftp server (PUT mydata)
- Step 2:
  - PORT destination IP, destination port
- Step 3:
  - GET mydata

# SMTP: Simple Mail Transfer Protocol

---

- initially specified in RFC 821
- de facto standard for email transmission
- simple, text-based protocol
- MIME used to encode binary files (attachments)
- listens on port 25
- push protocol:
  - used to send email
  - used to exchange emails between servers
  - clients have to retrieve emails via other protocols such as IMAP or POP

# SMTP

---

- Security Issues
  - mail servers have wide distribution base and are publicly accessible
    - software vulnerabilities
    - configuration errors
  - *sendmail*
    - one of the first SMTP implementations (MTAs)
    - long history of vulnerabilities
    - complicated configuration (M4 macro language)
    - e.g., buffer overflow in Sendmail 8.12.9 and before (2003)
  - *postfix, qmail*
    - secure replacements

# SMTP Session

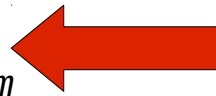
---

```
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
```

# SMTP Session

---

```
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
```



No Authentication

# SMTP

---

- No Authentication
  - everyone can connect to a SMTP server and transmit a message
  - server cannot check sender identity (besides IP address/domain name)
- Fundamental reason: open, distributed system
  - you can receive email from anyone on the internet
  - there is no central authority
  - this is why email was so successful!
  - ...also the root cause of SPAM

# Open Mail Relay

---

- The mail server for example.com should deliver:
  - messages from email accounts of example.com
  - messages to email accounts of example.com
- It should NOT relay messages:
  - from untrusted sources, to destinations outside example.com
  - Open Mail Relay: will be used to deliver SPAM

# SMTP Authentication

---

- IP address
  - check if user is inside the example.com network
- Extensions
  - SMTP-AUTH
    - access control list with explicit login
    - clients must be aware of SMTP-AUTH
  - POP-before-SMTP
    - logins are simulated by POP request (which require a login)
    - when a client performs a POP request, its IP address is authenticated with the SMTP server for some time (e.g., 30 minutes)

# Address Spoofing

---

- Authentication by IP:
  - Anyone in example.com network can send email from ceo@example.com
- Sender can forge source address
  - pretend to be relaying email from mybank.com

# Address Spoofing (countermeasures)

---

- SPF: Sender Policy Framework
  - leverage DNS infrastructure
  - owner of mydomain.com specifies which IPs are authorized to send emails from \*@mydomain.com
  - uses TXT records in DNS server (no changes required to DNS implementation)
  - SMTP server can check sender IP/domain name against authorized senders

# SPF Example

---

```
$host -t TXT gmx.net
```

```
gmx.net_descriptive text "v=spf1 ip4:213.165.64.0/23 ip4:74.208.5.64/26  
-all"
```

- An SPF compliant SMTP server receiving mail from \*@gmx.net:
  - will check if client IP is in the **list**. If not, it will reject mail claiming to be from gmx.net

# Spam

---

- Unsolicited email message
- Gather destination email addresses
  - brute force guessing
  - harvesting (web pages, mailing lists, news groups, ...)
  - malware (steal user's address book)
  - verified address are more valuable
- Delivering spam messages
  - own machine (not very smart)
  - other machines
    - open mail relays
    - open proxies
    - web forms
    - zombie nets (compromised machines)

# Spam

---

- Countermeasures
  - spam filtering tools (e.g., SpamAssassin)
  - blacklists: identify origins of spam messages and quickly distribute this information
  - graylisting: temporarily reject email from unknown senders
    - legitimate senders will retry
    - spammers often don't
  - infrastructure
    - SPF (sender policy framework)

# Spam

---

- Reasons for spam
  - legitimate businesses advertise products and services
  - attempts to get money from victims
    - actually quite old idea, was done with letters decades ago
    - victims sometimes even travel to remote places
  - offer of porn or other interesting material to lure people on sites where Trojan horses can be installed
- Statistics (a few years old)
  - Ikarus Scan Centers
    - 3.5 million mail messages per day
    - 60% of these messages are spam
    - 30% contain virus attachments
  - MessageLabs (used by EU)
    - 66% are spam

# Phishing

---

- More recent scam that
  - exploits weakness of SMTP protocol and
  - social engineering aspects
- Tricks people into providing sensitive information
  - create a situation that asks receiver to act on (urgent) problem
  - provide a link to site to solve problem
  - site prepared by attacker
    - appearance of site is spoofed
    - asks for personal information
- Interesting side note
  - scammers typically require people to launder money
  - additional spam mails that invite people to “earn money with their bank account”

# Phishing

From: Chase Online <banking@chaseonline.com>  
Subject: **Chase Online Account Notification**  
Date: March 28, 2006 8:09:58 AM CEST  
To: Christopher Kruegel



## Important Changes to Your Online Experience

Dear Chase Customer,

We're pleased to announce that later this month your online account information will be available through Chase Online<sup>SM</sup>. The site was recently ranked #1 by Gómez<sup>®</sup>, an independent Internet rating service. It is easy to navigate, with icons to guide you to many of our most frequently used services. See for yourself by taking a test-drive: To make the switch you will need to re-enroll your account by clicking the above link as soon as possible, as your business card accounts will no longer be available in our old database.

Please note the following changes to the payment posting policy on Chase Online<sup>SM</sup>: Payments made prior to 4PM EST will be credited the same day. Payments made after 4PM EST will be credited the next day at no charge. If it is after 4PM and you need your payment to be credited the same day, you can do so for a fee.

<http://chaseonline.chase.com/siteminderagent/forms/formpost.fcc>

We apologize for any inconvenience this may cause. Thanks for choosing Chase, and welcome to Chase Online.

Sincerely,

A handwritten signature in black ink that reads "Carter Franke".

Carter Franke  
Chase Card Services

# Phishing

From: Chase Online <banking@chaseonline.com>  
Subject: **Chase Online Account Notification**  
Date: March 28, 2006 8:09:58 AM CEST  
To: Christopher

The image shows a phishing email on the left and a screenshot of a fake Chase website on the right. The email is from 'Chase Online' and has a subject line 'Chase Online Account Notification'. The website screenshot shows a browser window with the URL 'http://www.chase.com/'. The website has a blue header with the Chase logo and navigation links. A large banner advertises a '4.50% APY CD Guaranteed return'. Below the banner are sections for 'Returning Users: Log On' with input fields for 'User ID' and 'Password', and a 'Log On' button. There are also sections for 'Personal Banking', 'Business', 'Personal Lending', and 'Insurance & Investing'. A 'Protect Your Identity' section is visible at the bottom left. The email text on the left is partially visible, showing a greeting 'Dear Chase' and a closing 'Sincerely, Carter'.

# Phishing

From: Chase Online <banking@chaseonline.com>  
Subject: Chase Online Account Notification  
Date: March 28, 2006 8:09:58 AM CEST  
To: Christopher

Chase Online Account Notification

Mail thinks this message is Junk Mail. [?] [Load Images] [Not Junk]

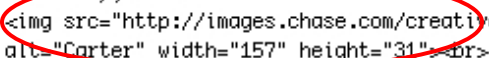
charge. If it is after 4PM and you need your payment to be credited the same day, you can do so for a fee.  
<br><br>

[http://datsun.websitewelcome.com/~lourdes/.www.chase.com/colportal/prospect.php?\\_nfpb=change\\_form](http://datsun.websitewelcome.com/~lourdes/.www.chase.com/colportal/prospect.php?_nfpb=change_form) <a href="http://chaseonline.chase.com/siteminderagent/forms/formpost.fcc">

<br>  
</br>

We apologize for any inconvenience this may cause. Thanks for choosing Chase, and welcome to Chase Online.  
<br><br>

Sincerely, <br><br>

 alt="Carter" width="157" height="31" <br>  
Carter Franke <br>  
Chase Card Services</font><font size="2" color="#000000" face="Helvetica, Geneva, Arial, SunSans-Regular, sans-serif"><br>

# Phishing

**From:** Scarleting M. Waisting <ddribin@golfcoursecommunities.com>  
**Subject:** **[SPAM?] Verdienen mit Ihrem Bankkonto!**  
**Date:** March 22, 2006 10:59:14 PM CET  
**To:** Chris <chris@auto.tuwien.ac.at>

## PLATINWAY CORP

Ist das Ihr alter Traum ein hochbezahlter Manager einer erfolgreichen Firma zu werden? Wir helfen Ihnen diesen Traum in Erfüllung zu bringen. Sie bekommen einen stabilen Gewinn von der Arbeit im Internet.

Warum wir? Das ist sehr einfach! Haben sie schon die Nase voll mit ewigen Geldmangel? Brauchen Sie ein neues Haus oder ein neues Auto? Möchten Sie, daß Ihre Kinder gute Uni-Ausbildung bekommen?

Erinnern Sie sich an Worte von Pythagoras: „Nie beim Erreichten stehen bleiben!“

Alles, was Sie brauchen, ist Folgendes:

- ein Bürger Deutschlands zu sein
- ein Bankkonto zu haben
- ein Computer mit Internet-Verbindung zu besitzen
- etwas Freizeit zu haben
- und natürlich Ihren Wunsch.

Ihre Arbeit wird einfach, angenehm und vor allem EINTRÄGLICH sein! Unsere Gesellschaft kümmert sich um Ihren Wohlstand. Lassen Sie sich kostenlos registrieren um die neuen Möglichkeiten der Arbeit im Internet zu entdecken.

Machen Sie Ihr Leben besser mit Platinway Corp.

URL: <http://www.platinway.org/>  
E-Mail: [support@platinway.net](mailto:support@platinway.net)

© Copyright 2006 **PLATINWAY CORP.** All rights reserved.

# Phishing

---

- Camouflage techniques
  - use images, look&feel from original site
  - sender name and email addresses can be faked easily
  - attempt to avoid obvious spelling and grammar mistakes :-)
  
  - link to phishing site must be obfuscated
  - URL and port redirection  
`http://www.bank.com@evil.com:80/index.html`

# Phishing Defense

---

- User education
- Stronger authentication of sources
  - difficult without global PKI
  - ad-hoc mechanisms such as SiteKey or iTans
    - can be bypassed by active phishing attacks
- Techniques to detect sites that faithfully mimic others
  - SpoofGuard
    - browser plug-in
    - uses heuristics such as image similarity, domain name similarity, ...
  - active crawling of the web for suspicious sites

# Phishing Defense

---

- Techniques to ensure that password is not shared between sites
  - problem that users want to reuse passwords
- Password hashing
  - generate unique passwords for different sites
    - combine original password and URL
  - cannot protect sensitive information in general, because data changed
- AntiPhish
  - browser plug-in for Firefox and Internet Explorer
  - user explicitly tags of all sensitive information
  - sharing of information results in warnings

# Conclusions

---

- Traditional Internet applications
  - not built with security in mind
  - some could be easily replaced (telnet, rservices)
  - others cause significant problems
- Remote Access
- DNS
  - simple UDP-based request / reply structure
  - root server bottleneck (denial of service danger), domain hijacking
- FTP
  - different command and data channel, FTP bounce
- SMTP
  - sender authentication lacking, spam, phishing