Security in Web Applications

François Roland

Hermès Engineering consultant
Software architect at bpost

francois.roland@hermes-ecs.com
@FrRoland

Hermès Engineering

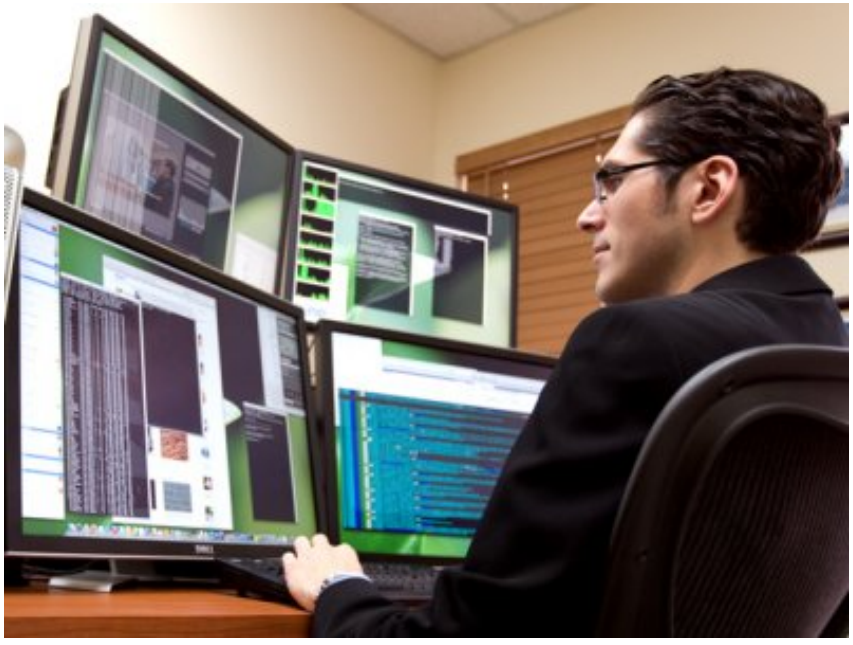
- IT consultancy services
- About 80 IT specialists
- Belgium and Luxembourg

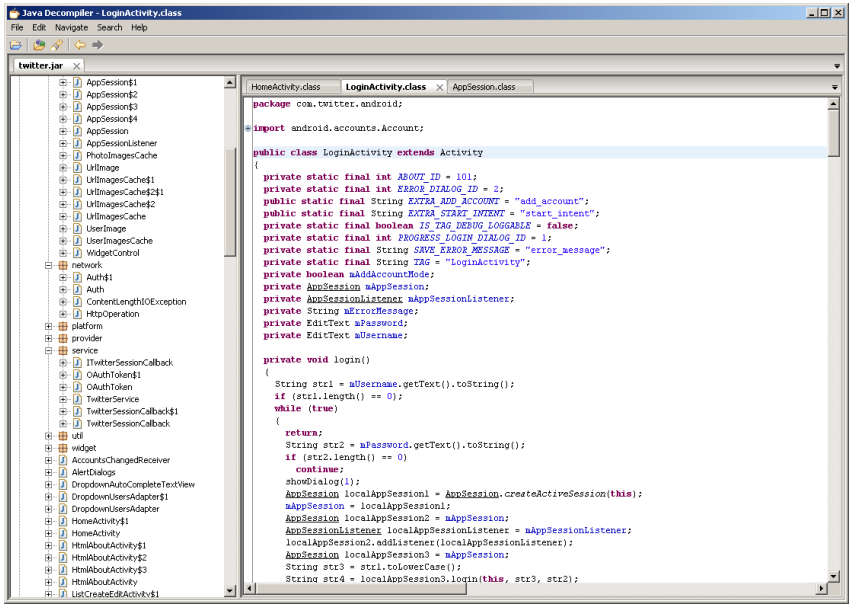
business intelligence

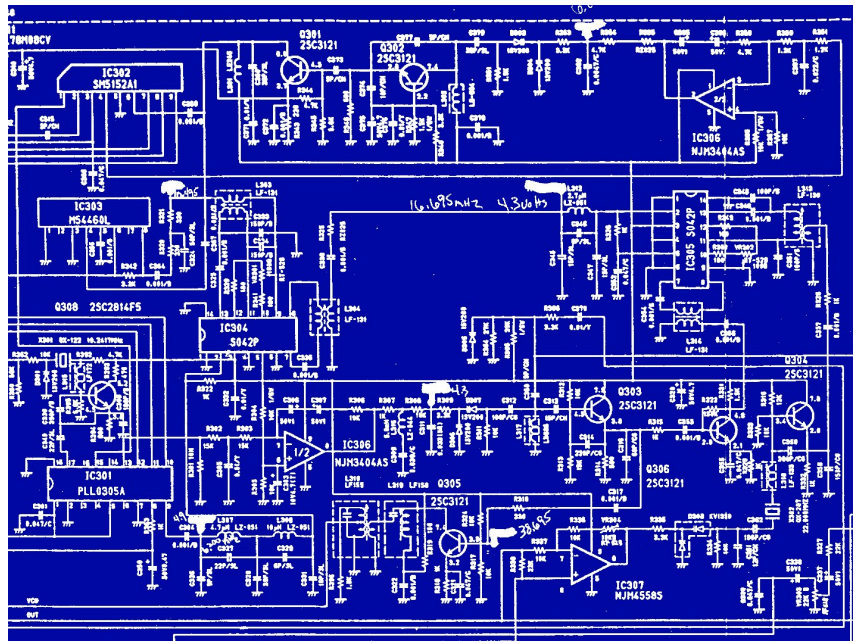
operational IT

systems

cryptography




Java Decompiler - LoginActivity.class
File Edit Navigate Search Help
twitter.jar
HomeActivity.class
LoginActivity.class
AppSession.class
AppSession$1
AppSession$2
AppSession$3
AppSession$4
AppSession
AppSessionListener
PhotoImagesCache
UrlImage
UrlImagesCache$1
UrlImagesCache$2$1
UrlImagesCache$2
UrlImagesCache
UserImage
UserImagesCache
WidgetControl
network
Auth$1
Auth
ContentLengthIOException
HttpOperation
platform
provider
service
ITwitterSessionCallback
OAuthToken$1
OAuthToken
TwitterService
TwitterSessionCallback$1
TwitterSessionCallback
util
widget
AccountsChangedReceiver
AlertDialogs
DropdownAutoCompleteTextView
DropdownUsersAdapter$1
DropdownUsersAdapter
HomeActivity$1
HomeActivity
HtmlAboutActivity$1
HtmlAboutActivity$2
HtmlAboutActivity$3
HtmlAboutActivity
ListCreateEditActivity$1
package com.twitter.android;
import android.accounts.Account;
public class LoginActivity extends Activity
{
private static final int ABOUT_ID = 101;
private static final int ERROR_DIALOG_ID = 2;
public static final String EXTRA_ADD_ACCOUNT = "add_account";
public static final String EXTRA_START_INTENT = "start_intent";
private static final boolean IS_TAG_DEBUG_LOGGABLE = false;
private static final int PROGRESS_LOGIN_DIALOG_ID = 1;
private static final String SAVE_ERROR_MESSAGE = "error_message";
private static final String TAG = "LoginActivity";
private boolean mAddAccountMode;
private AppSession mAppSession;
private AppSessionListener mAppSessionListener;
private String mErrorMessage;
private EditText mPassword;
private EditText mUsername;
private void login()
{
String str1 = mUsername.getText().toString();
if (str1.length() == 0);
while (true)
{
return;
String str2 = mPassword.getText().toString();
if (str2.length() == 0)
continue;
showDialog(1);
AppSession localAppSession1 = AppSession.createActiveSession(this);
mAppSession = localAppSession1;
AppSession localAppSession2 = mAppSession;
AppSessionListener localAppSessionListener = mAppSessionListener;
localAppSession2.addListener(localAppSessionListener);
AppSession localAppSession3 = mAppSession;
String str3 = str1.toLowerCase();
String str4 = localAppSession3.login(this, str3, str2);


IC302
SM5152A1
Q301
2SC3121
Q302
2SC3121
IC305
NJM3404AS
IC303
M54469L
IC304
S042P
IC305 S042P
Q308 2SC2814F5
Q303
2SC3121
Q304
2SC3121
IC306
NJM3404AS
IC301
PLL0305A
Q305
2SC3121
Q306
2SC3121
IC307
NJM4558S


easy
STAPLES


Vs


SECURE


TIME
The Project Management Triangle
QUALITY
COST/ RESOURCES

identify

investigate

diagnose

Agenda

- Psycho-sociological Aspect
- Authentication and Identity
- Attacks
- Prevention
- Detection and Analysis

The Team

Krzysztof Magusiak

Michaël Lacroix

Vincent Gribomont

Psycho-sociological Aspect

Internet Security History

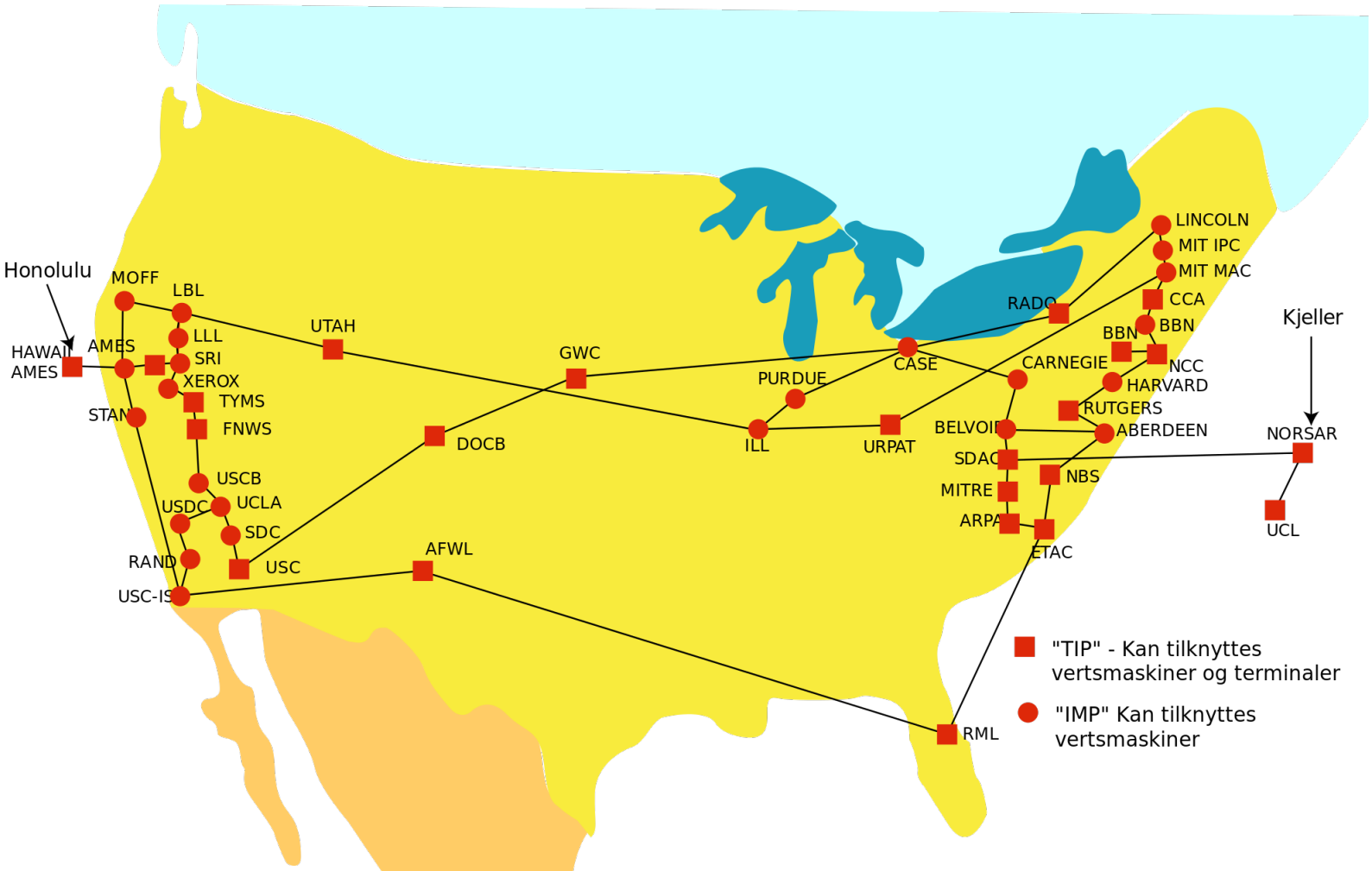
Army, Universities, ...

Internet Security History

Army, Universities, ...
Security?

Internet Security History

Army, Universities, ...
Security?



Nowadays...

It's **everywhere**!

$\not\Rightarrow$

PlayStation Network outage (2011)

77 million accounts affected.


- Security measures
- Guarantees
- Protection

Infosecurity Europe 2003 survey

90% of surveyed people

Infosecurity Europe 2003 survey

90% of surveyed people

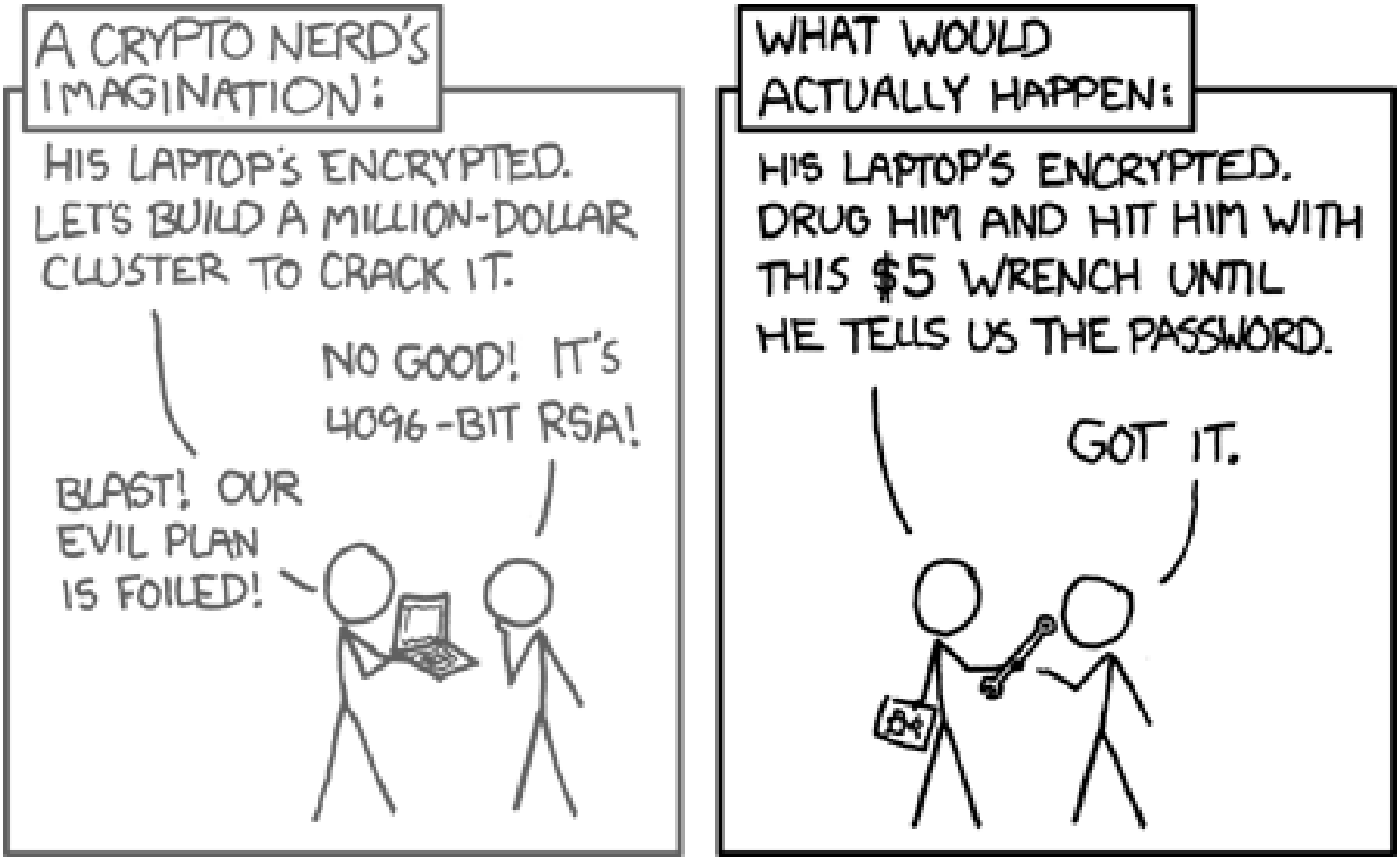
password

Infosecurity Europe 2003 survey

90% of surveyed people

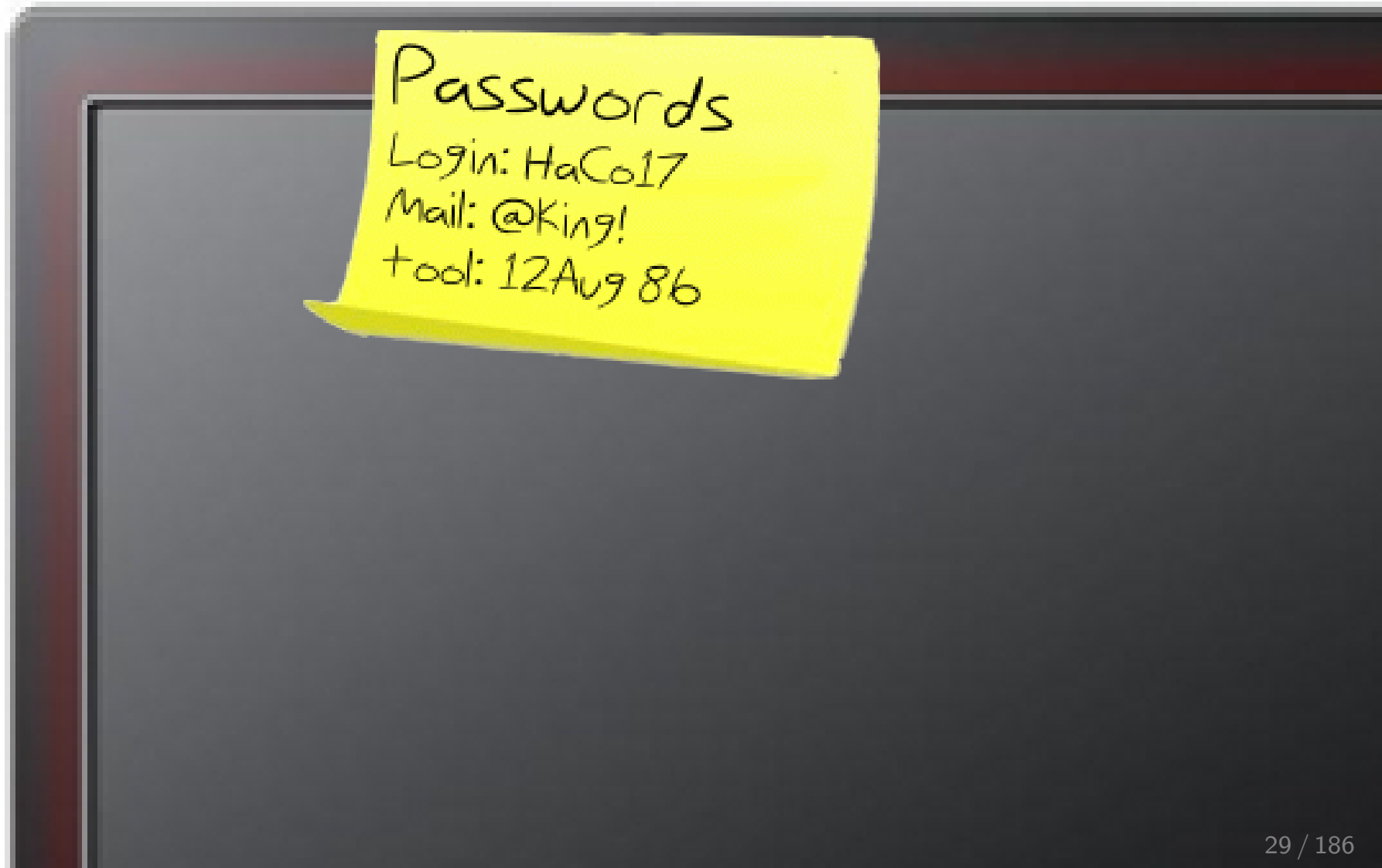
password Vs 


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.


Most Common
Password List

1. password
2. 123546
3. 12345678
4. abc123
5. qwerty
6. monkey
7. letmein
8. dragon
9. 111111
10. baseball
11. ...


Passwords
Login: HaCo17
Mail: @King!
+ool: 12Aug 86

- 1password
  `https://agilebits.com/onepassword`
- lastpass
  `https://lastpass.com`
- pwdhash
  `https://www.pwdhash.com`

Security  Usability

Security  Usability

secure password  simple password

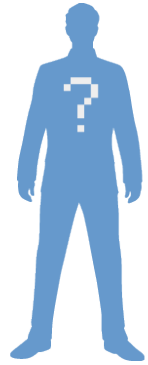
Security		**Usability**
secure password		simple password
authentication		no authentication

Security		**Usability**
secure password		simple password
authentication		no authentication
encoded data		plain text

Authentication and Identity

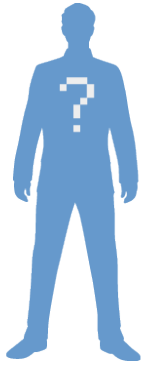
Identity

Who you are.



Identity

Who you are.

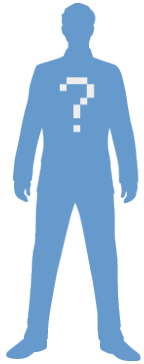


Authentication

Are you who you claim to be?

Identity

Who you are.



Authentication

Are you who you claim to be?

Authorization

What are you allowed to do?

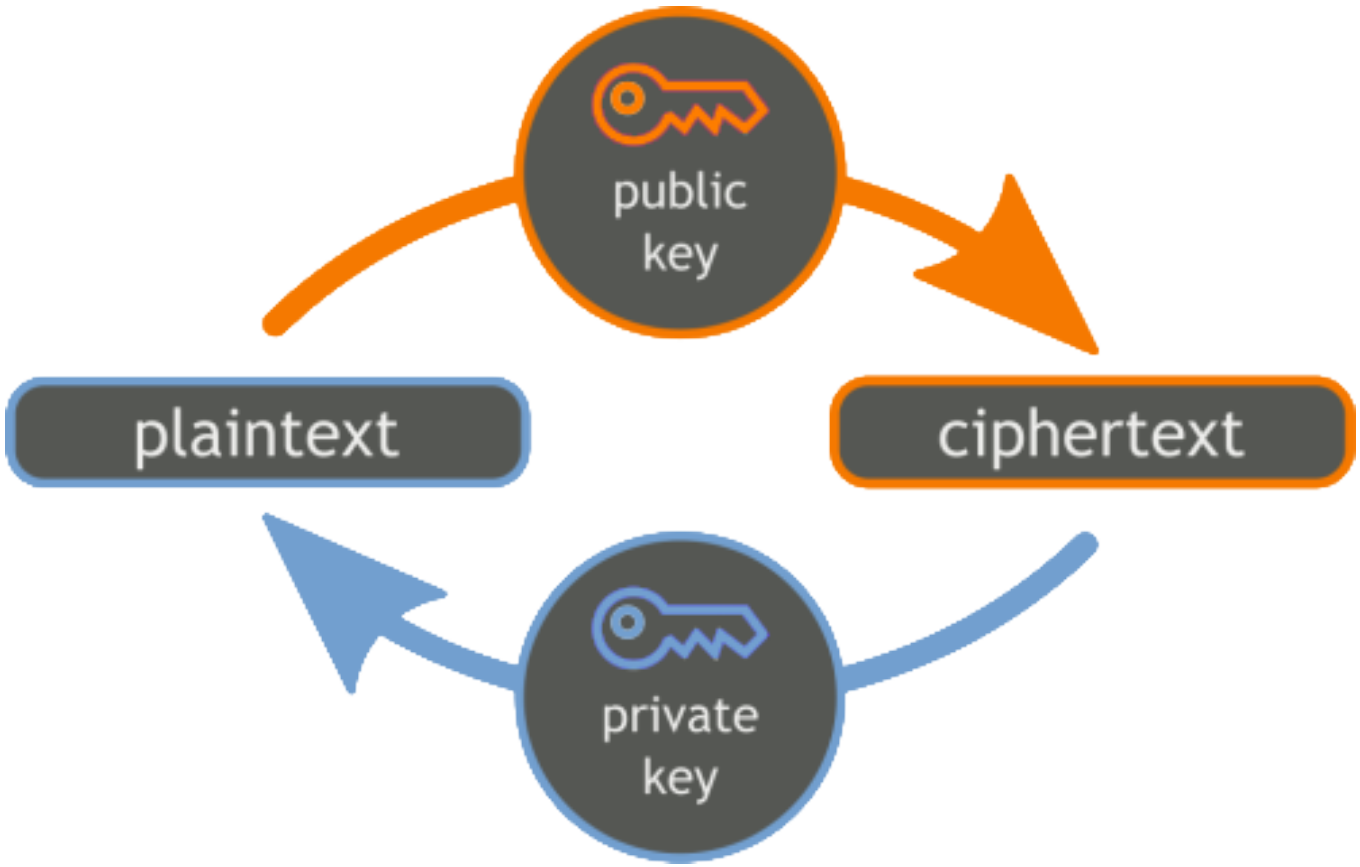
Identification and confidentiality

How does HTTP work?

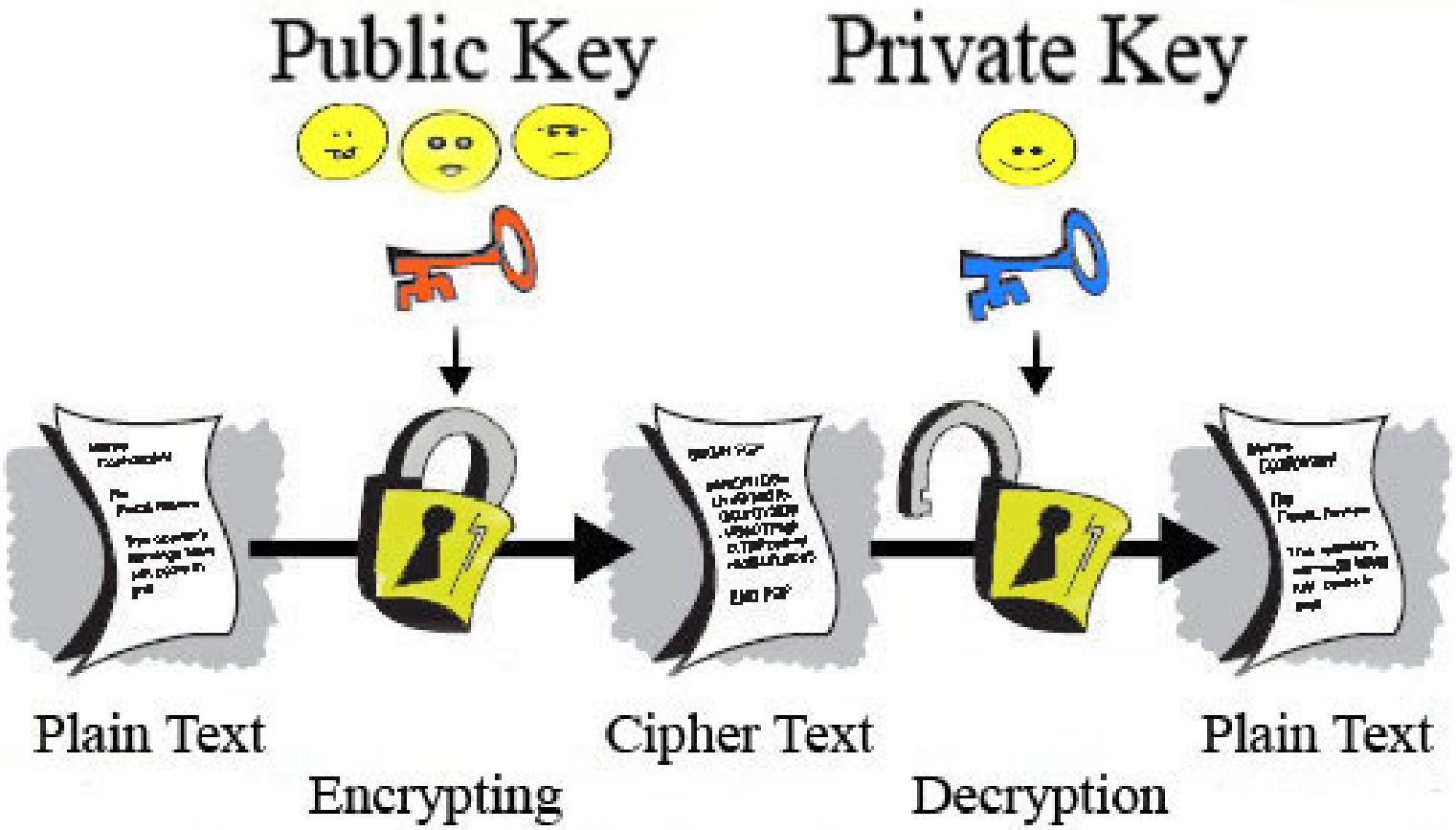
```
GET /wiki/Hypertext_Transfer_Protocol HTTP/1.1
Host: en.wikipedia.org
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:19.0)
    Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,
    application/xml;q=0.9,*/*;q=0.8
Cookie: mediaWiki.user.id=m5M[illegible]bI4Lrx;
Connection: keep-alive
...
```

Symmetric cryptography

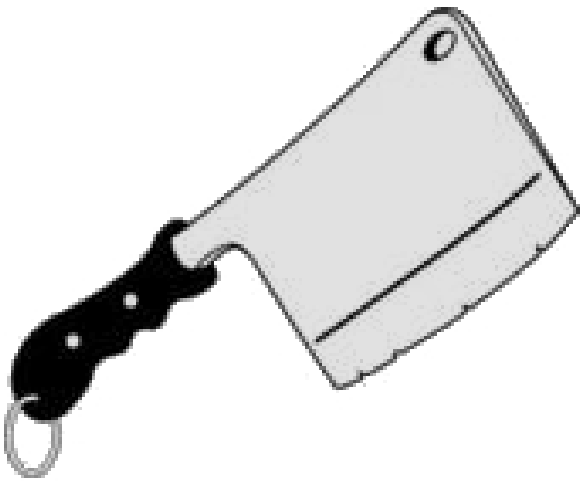
Asymmetric cryptography

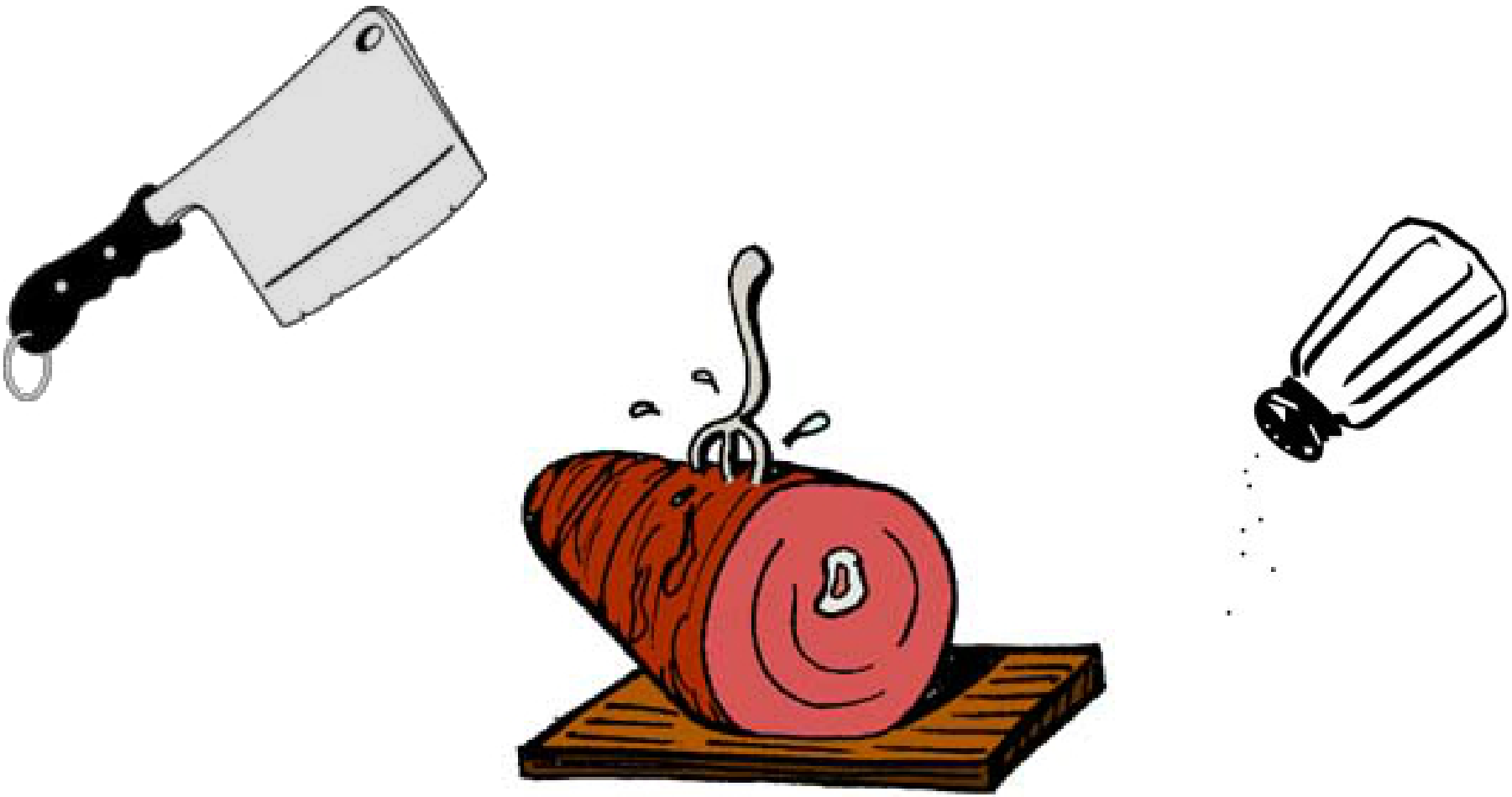


Public-private keys



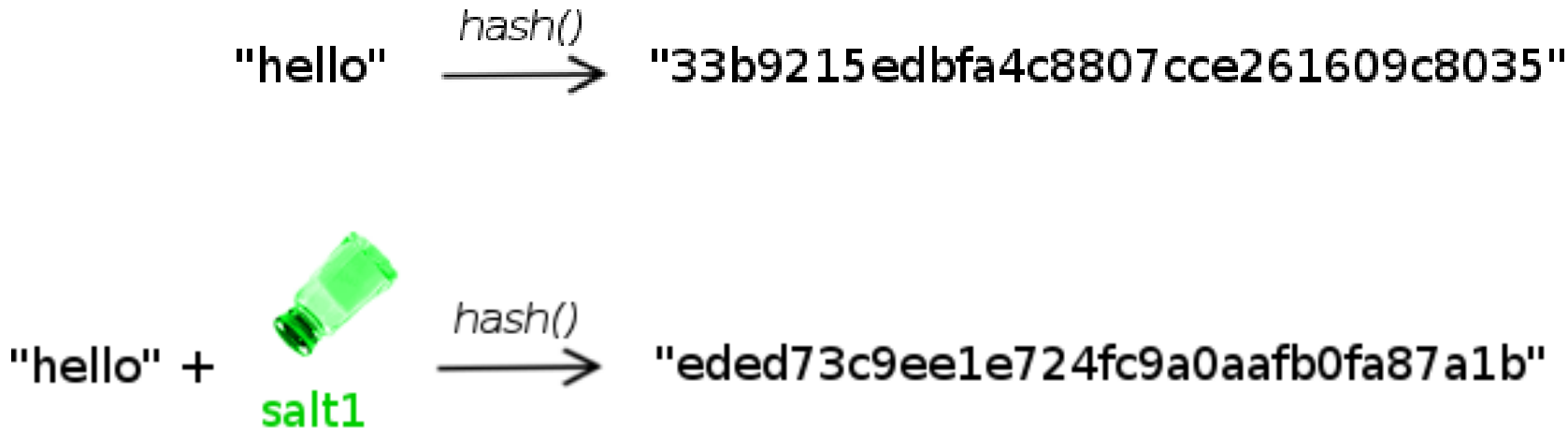
Symmetric		**Asymmetric**
Fast		Slow
Pre-shared secret		No pre-shared secret
Owner(s) of the secret		Owner of the private secret



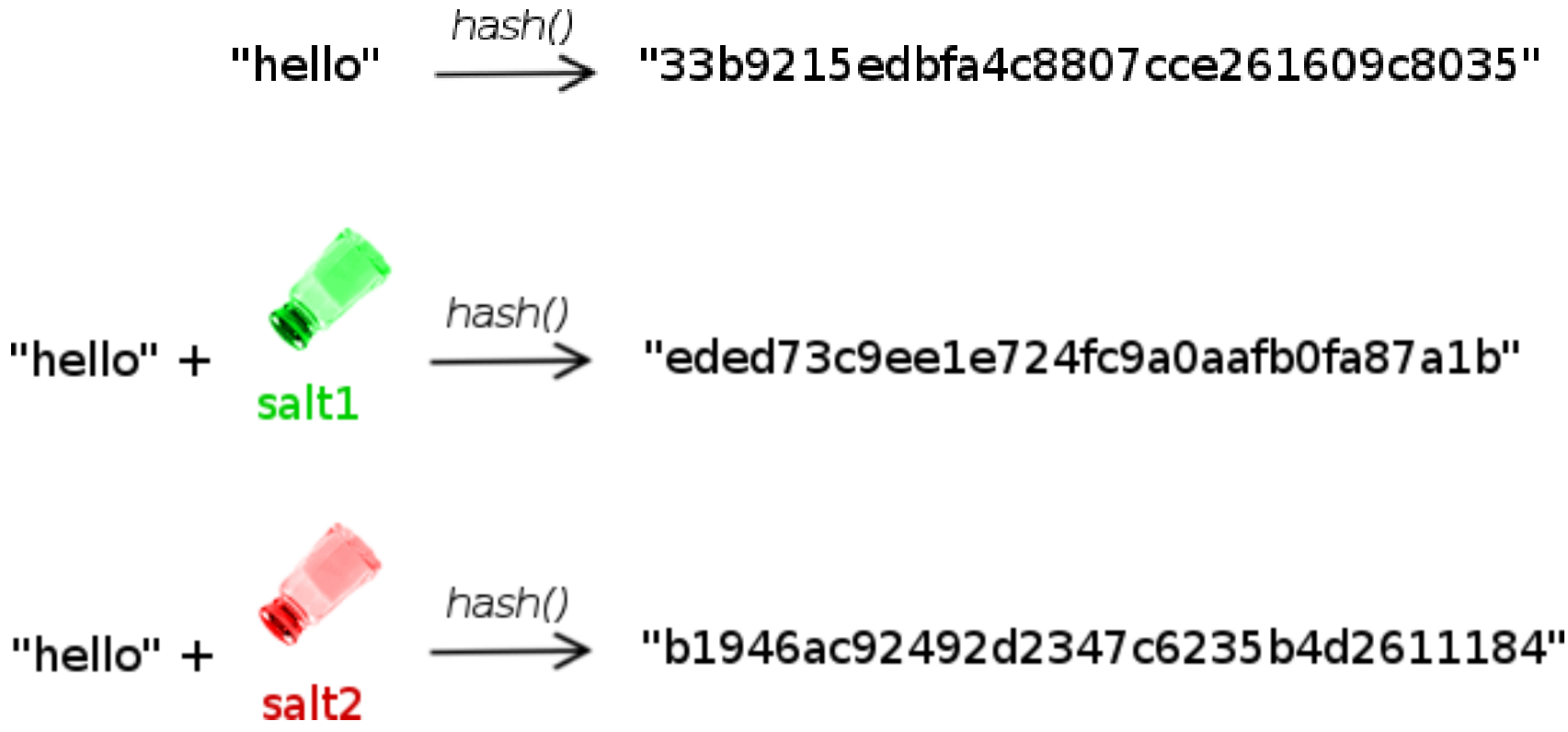


Hash + Salt

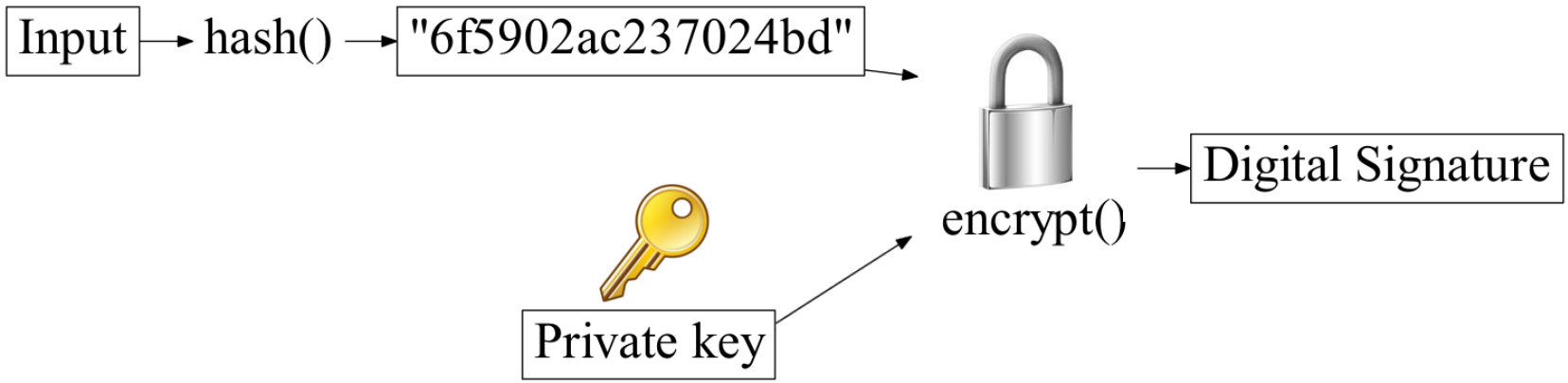
Hash + Salt


"hello"
hash()
"33b9215edbfa4c8807cce261609c8035"
"hello" +
salt1
hash()
"eded73c9ee1e724fc9a0aafb0fa87a1b"

Hash + Salt


"hello"
hash()
"33b9215edbfa4c8807cce261609c8035"
"hello" +
salt1
hash()
"eded73c9ee1e724fc9a0aafb0fa87a1b"
"hello" +
salt2
hash()
"b1946ac92492d2347c6235b4d2611184"

Signatures


Input
hash()
"6f5902ac237024bd"
encrypt()
Digital Signature
Private key

Authentication means

Authentication means

- Password

Authentication means

- Password
- Tokens

Authentication means

- Password
- Tokens
- One time passwords

Authentication means

- Password
- Tokens
- One time passwords
- Certificates

Authentication means

- Password
- Tokens
- One time passwords
- Certificates
- Signature challenge

Authentication means

- Password
- Tokens
- One time passwords
- Certificates
- Signature challenge
- Single Sign-On

- Password is the most common form of authentication

- Password is the most common form of authentication
- Password should be easily changeable

- Password is the most common form of authentication
- Password should be easily changeable
- People should have different passwords for different services

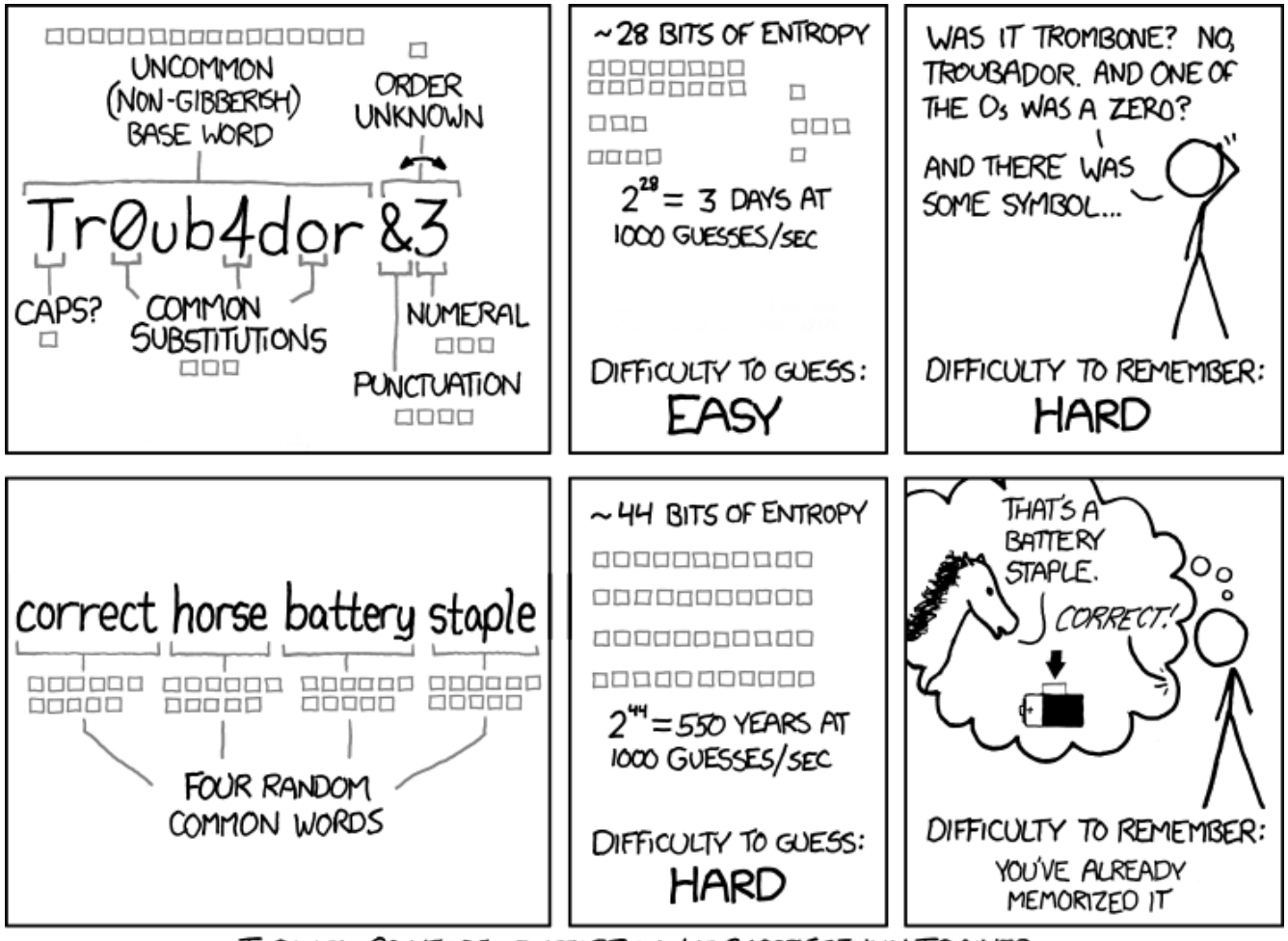
- Password is the most common form of authentication
- Password should be easily changeable
- People should have different passwords for different services
- Complexity must be sufficient

- Password is the most common form of authentication
- Password should be easily changeable
- People should have different passwords for different services
- Complexity must be sufficient
  - Sufficient length

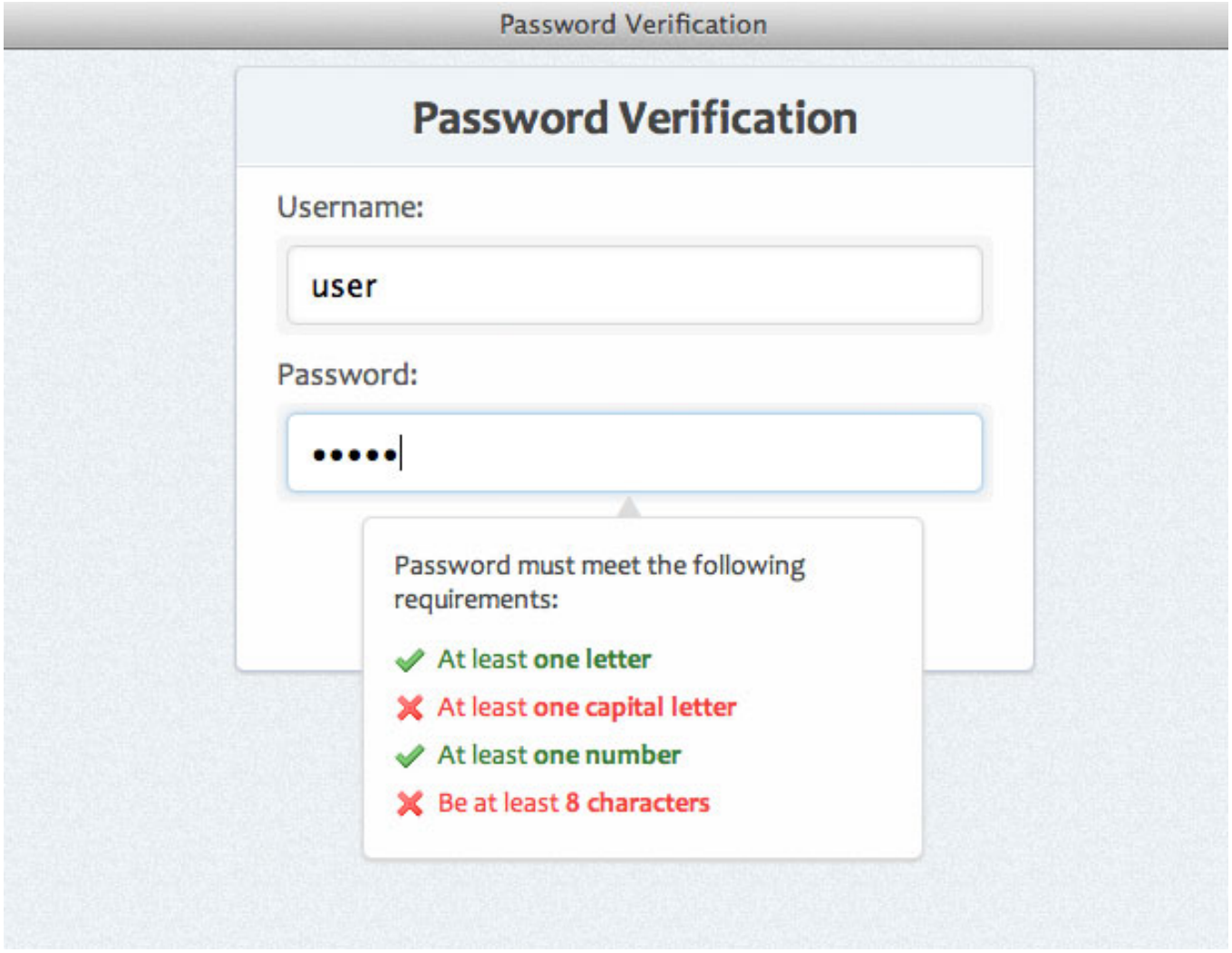
- Password is the most common form of authentication
- Password should be easily changeable
- People should have different passwords for different services
- Complexity must be sufficient
  - Sufficient length
  - Mix lower- and upper-case character

- Password is the most common form of authentication
- Password should be easily changeable
- People should have different passwords for different services
- Complexity must be sufficient
  - Sufficient length
  - Mix lower- and upper-case character
  - Use number and special digit

Password strength



Password complexity



Tokens


.be CITOYENS
www.belgium.be
Luc Van Den Bossche
1. NAVOZE
2. ZERUTA
3. SOLIPE
4. MOZARE
5. QULEKO
6. WAPERI
7. MOLEKA
8. TUDOFA
9. MOZARE
10. ZERUTA
11. SOLIPE
12. QULEKO
13. WAPERI
14. MOLEKA
15. NAVOZE
16. TUDOFA
17. ZERUTA
18. SOLIPE
19. QULEKO
20. WAPERI
21. MOLEKA
22. TUDOFA
23. NAVOZE
24. MOZARE

One time passwords

One time passwords - choice and distribution

Certificates

Standard SSL

- 2048 bits
- Auto validation

Pro SSL

- 2048 bits
- Paper validation
- Secure transactions up to $250,000

Business SSL

- 2048 bits
- Paper validation, by forms
- Secure transactions up to $250,000
- **Validated by COMODO**

Certificate Authority

- Private organism

Certificate Authority

- Private organism
- Verify the identity of certificate applicants

Certificate Authority

- Private organism
- Verify the identity of certificate applicants
- Sign, send and maintain certificates and revocation list

Certificate Authority

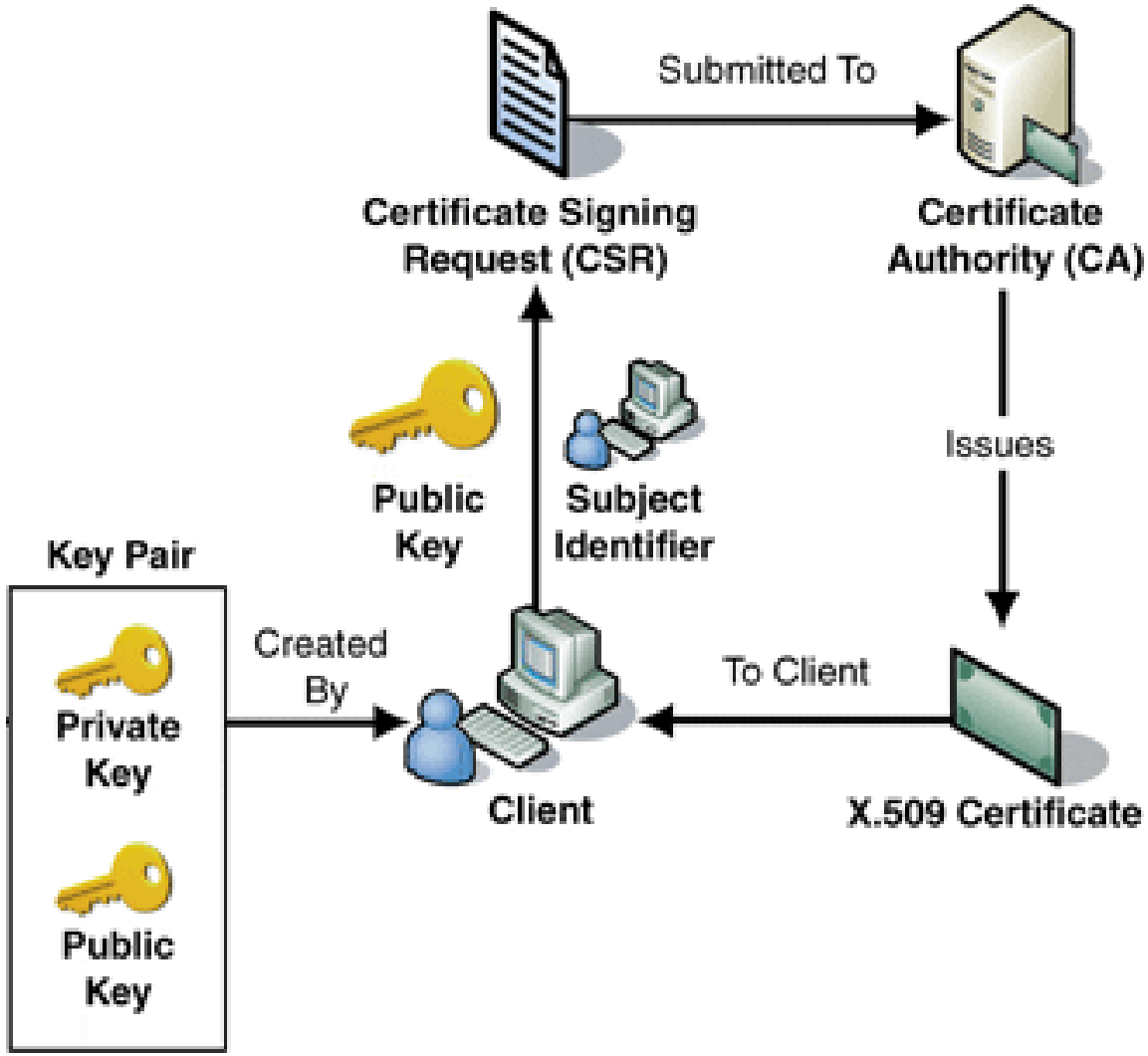
- Private organism
- Verify the identity of certificate applicants
- Sign, send and maintain certificates and revocation list
- Self-signature

Certificates - What are they made of?

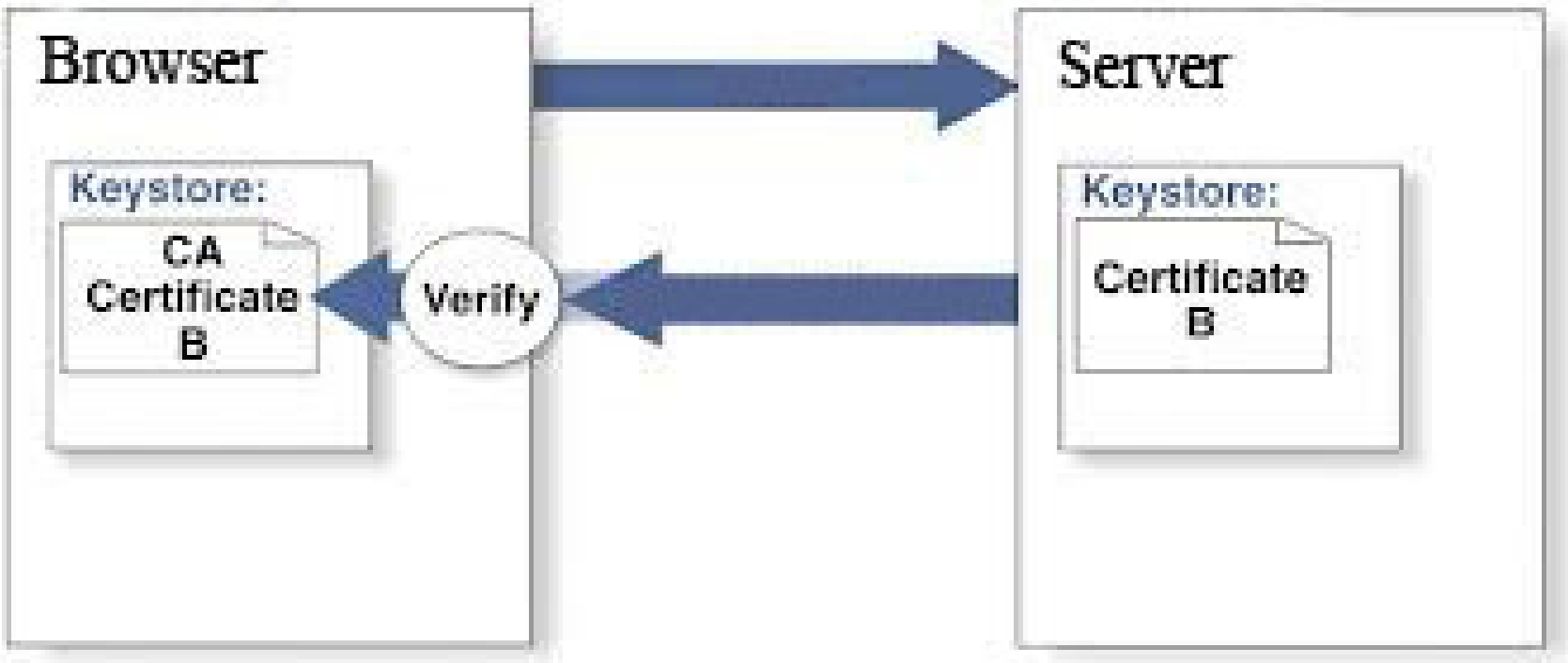
```
Certificate:
  Data:
    Version: 1 (0x0)
    Serial Number: 7829 (0x1e95)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=ZA, ST=Western Cape,...
            OU=Certification Services Division,
            CN=ThawteServerCA/emailAddress=...
    Validity
      Not Before: Jul  9 16:04:02 1998 GMT
      Not After : Jul  9 16:04:02 1999 GMT
```

```
  Subject: C=US, ST=Maryland,...
           OU=FreeSoft, CN=www.freesoft.org
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:be ...:41:8f
      Exponent: 65537 (0x10001)
Signature Algorithm: md5WithRSAEncryption
  93:5f...:68:9f
```

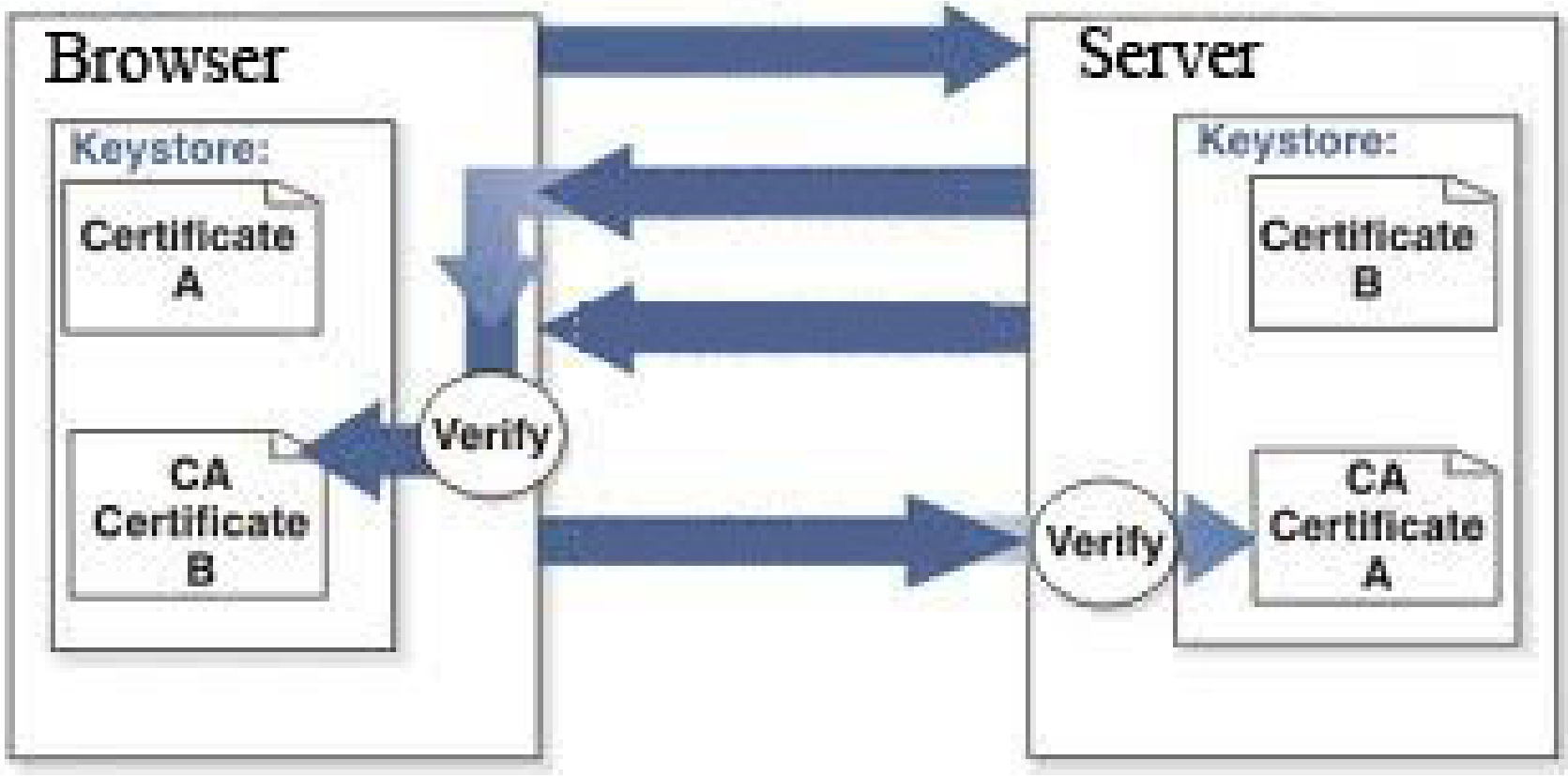
Obtaining a certificate



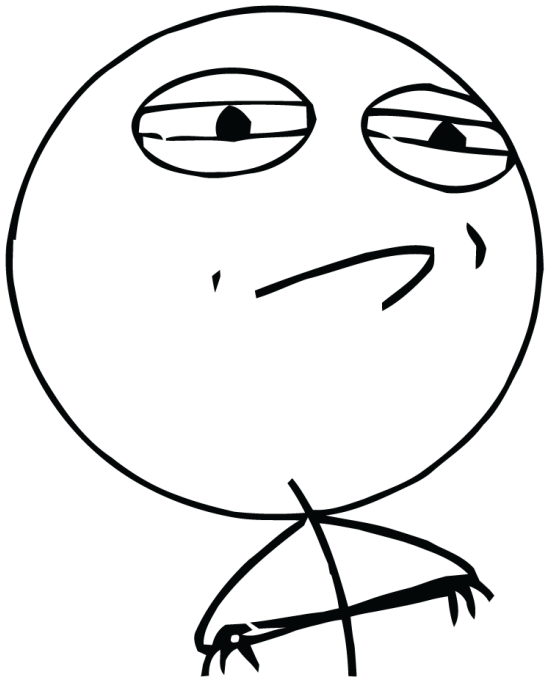
One-way


Browser
Server
Keystore:
CA
Certificate
B
Verify
Keystore:
Certificate
B

Two-way


Browser
Keystore:
Certificate A
CA Certificate B
Verify
Server
Keystore:
Certificate B
CA Certificate A
Verify

Signature challenge



Number used

O

N

C

E

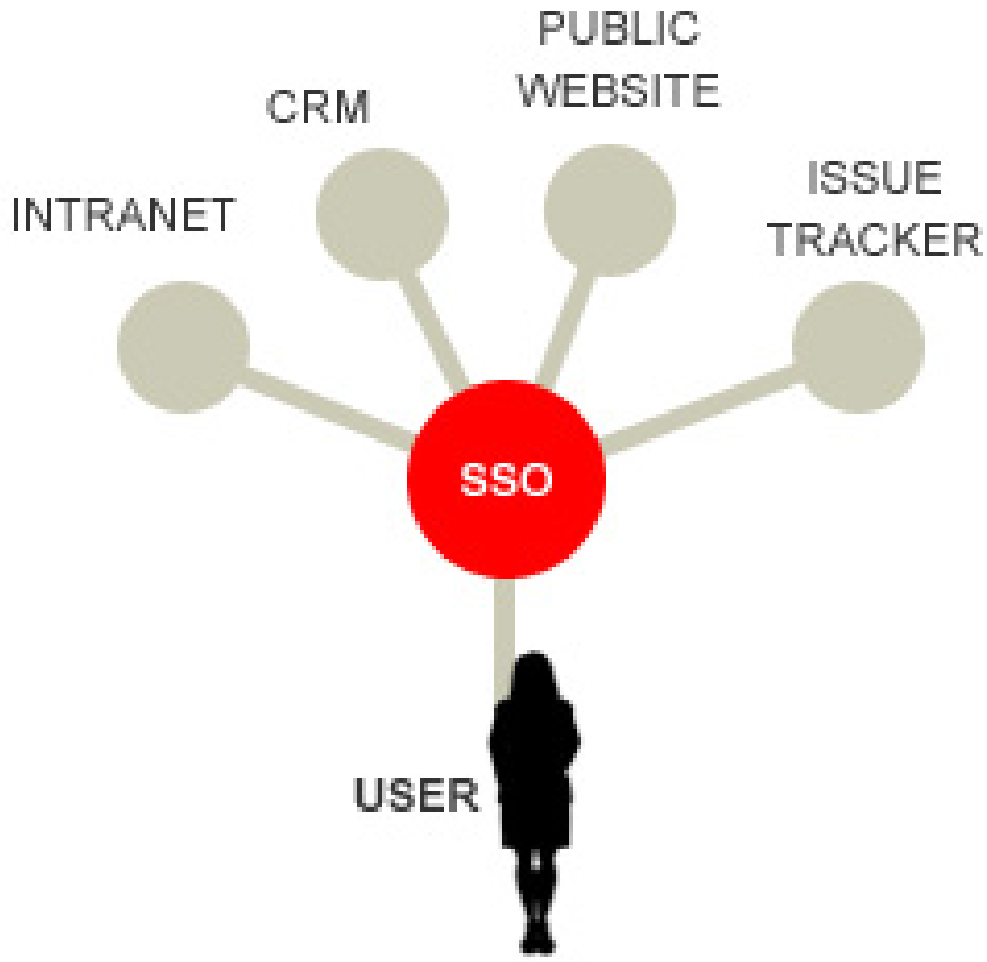
Multi-factor authentication

IS

OWNS

KNOWS

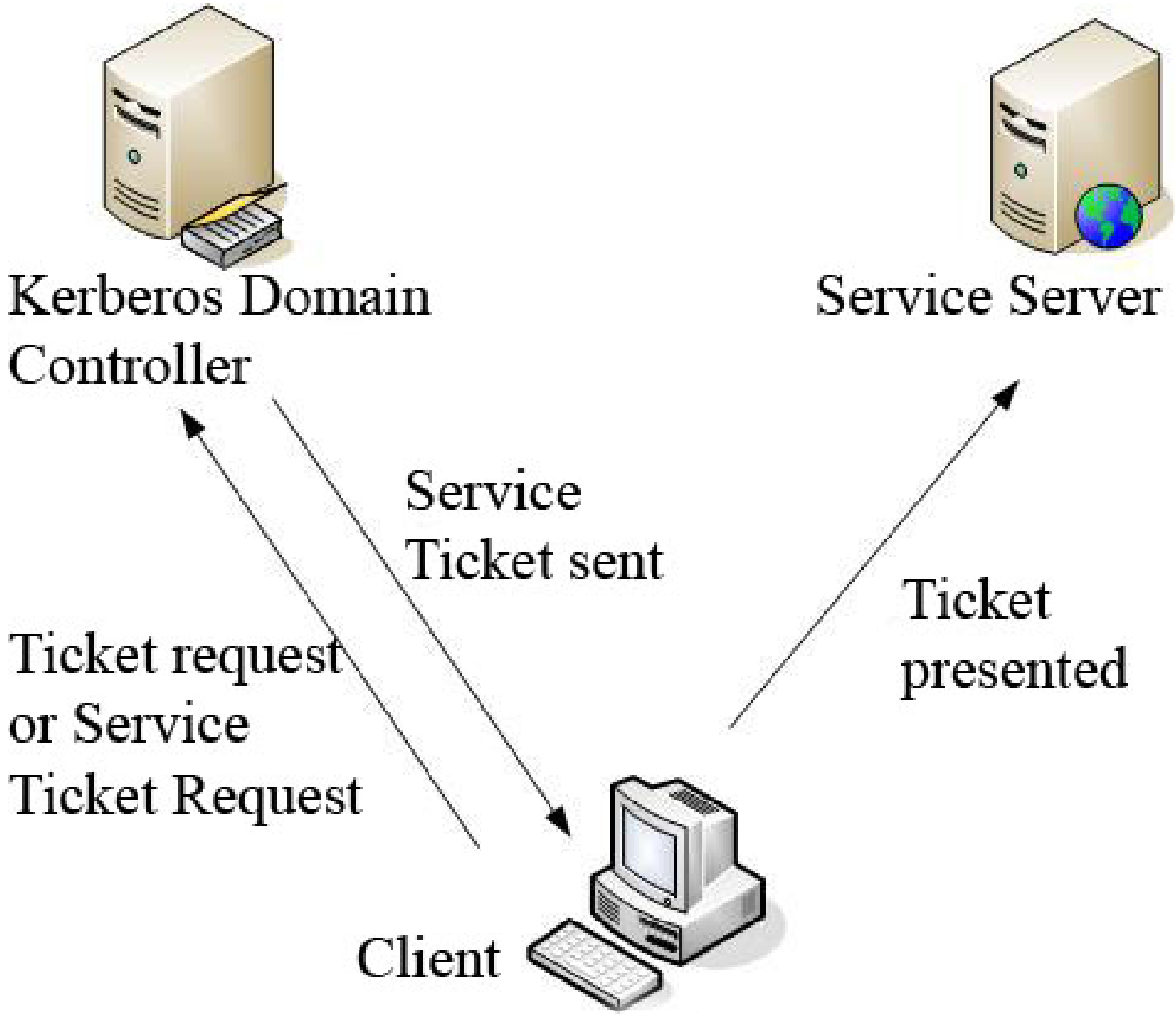
Single sign-on - principle



SSO providers

Kerberos

Kerberos - How does it work?



What if an user loss its authentication means?

Compromised authentication means

- Deactivate a *compromised* authentication means

Compromised authentication means

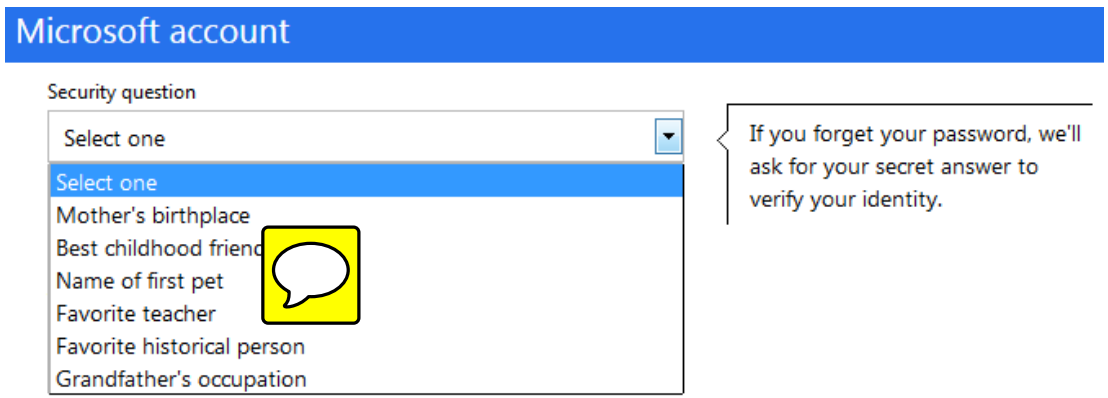
- Deactivate a *compromised* authentication means
- Authenticate the user by an *uncompromised* means

Compromised authentication means

- Deactivate a *compromised* authentication means
- Authenticate the user by an *uncompromised* means
- Give him a *new* primary authentication means

Lost password

- Security questions

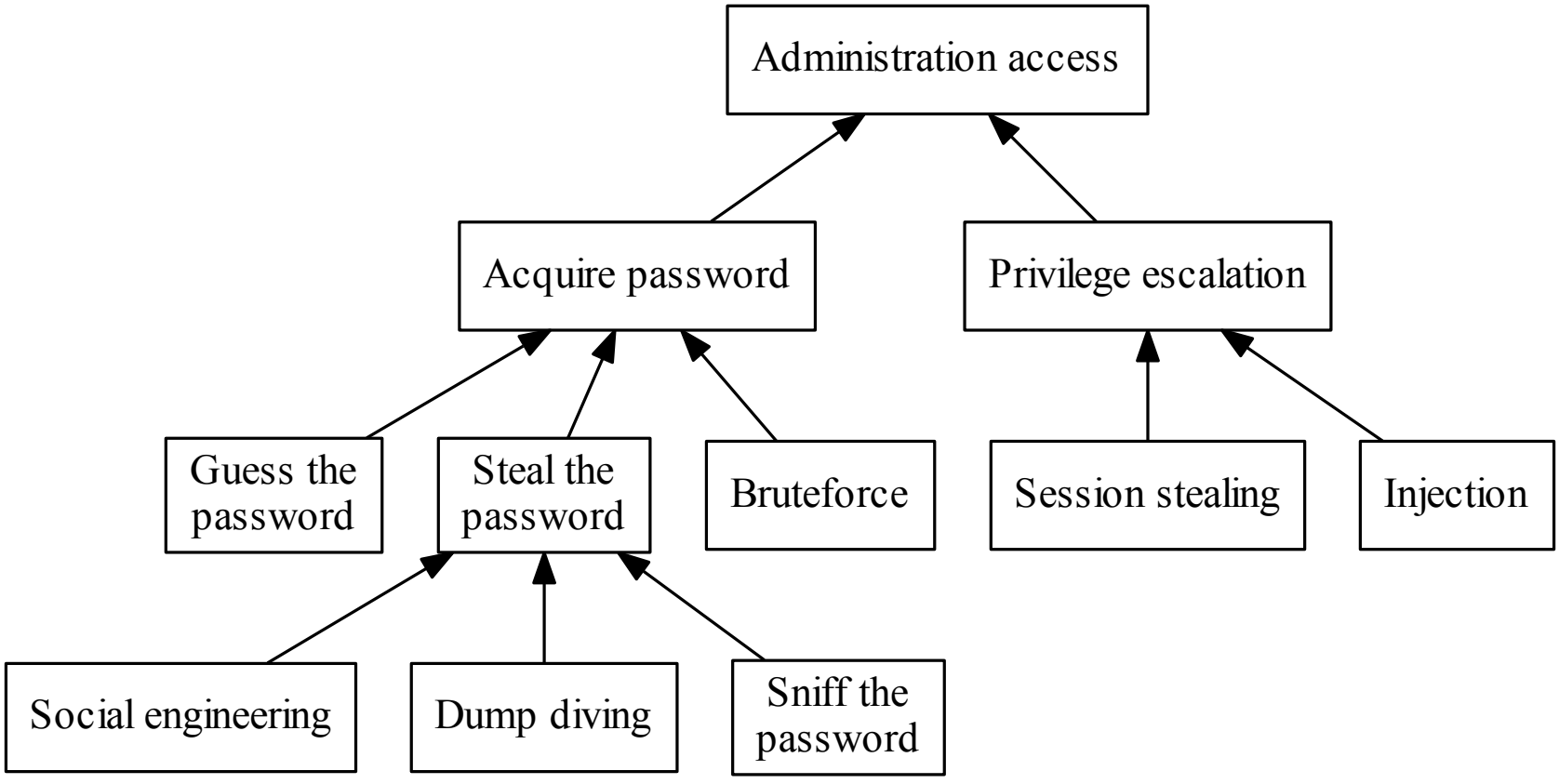


Lost password

- Security questions

- Another channel (e-mail, post, sms)

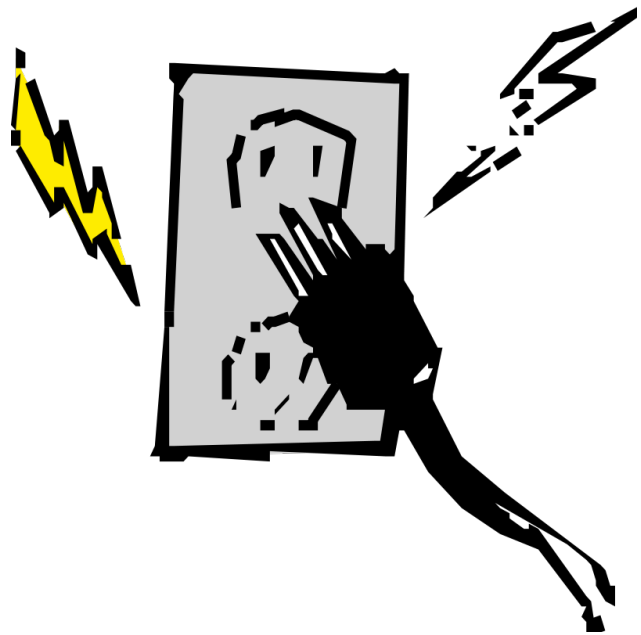
Attacks


Administration access
Acquire password
Privilege escalation
Guess the password
Steal the password
Bruteforce
Session stealing
Injection
Social engineering
Dump diving
Sniff the password

Threat agents

Threat agents

Threat agents



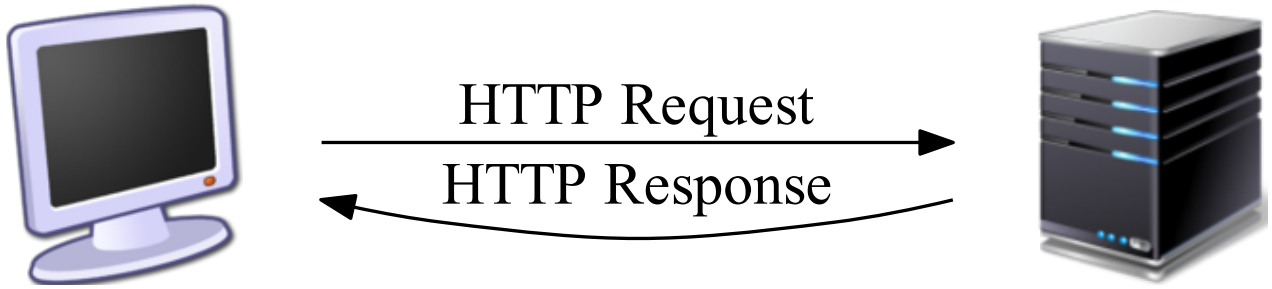
Threat agents

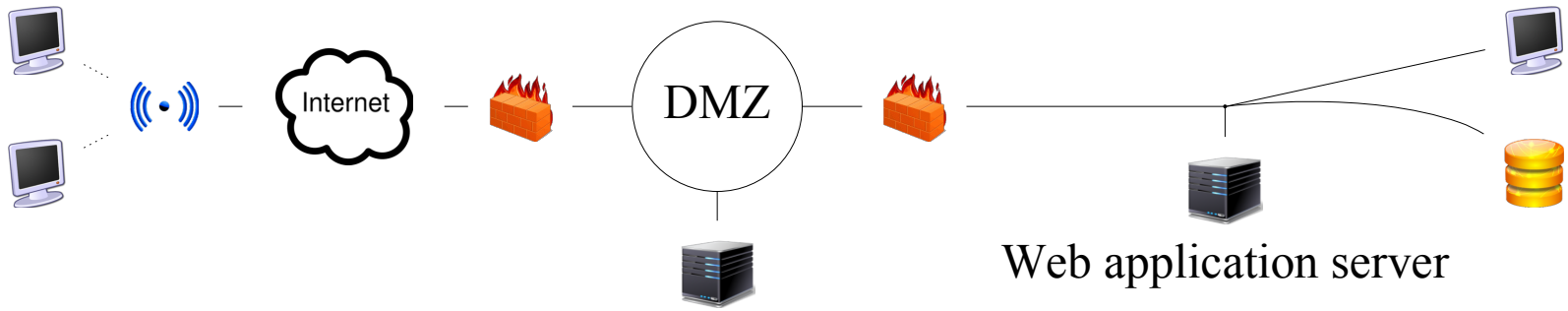
What is a *risk*?

The probable frequency and probable magnitude of future loss.
– *The Open Group*

$$\text{risk} = \text{likelihood} * \text{impact}$$

How is the web structured?


HTTP Request
HTTP Response


Internet
DMZ
Web server
Web application server

Open Web Application Security Project

https://www.owasp.org/

Top 10

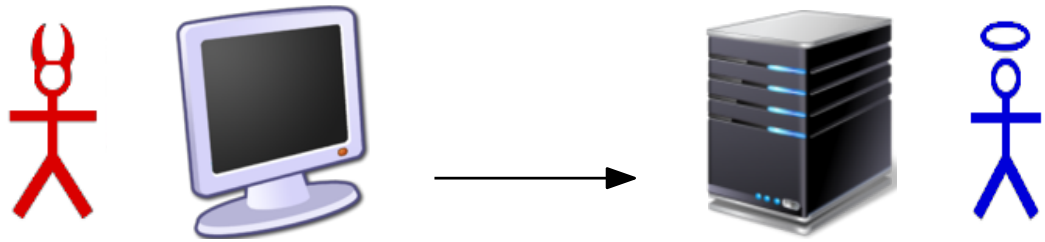
1. Injection
2. Cross-site Scripting (XSS)
3. Authentication and Session Management
4. Insecure Direct Object References
5. Cross-site Request Forgery (CSRF)
6. Security Misconfiguration
7. Insecure Cryptographic Storage
8. Failure to Restrict URL Access
9. Insufficient Transport Layer Protection
10. Unvalidated Forwards and Redirects

Demo website

Technologies Java website

- Spring
- Hibernate - h2database

1. Injection



THE password

```
String query = "select * from users"
  + " where user_name = '" + name + "'"
  + " and password = '" + password + "';
```

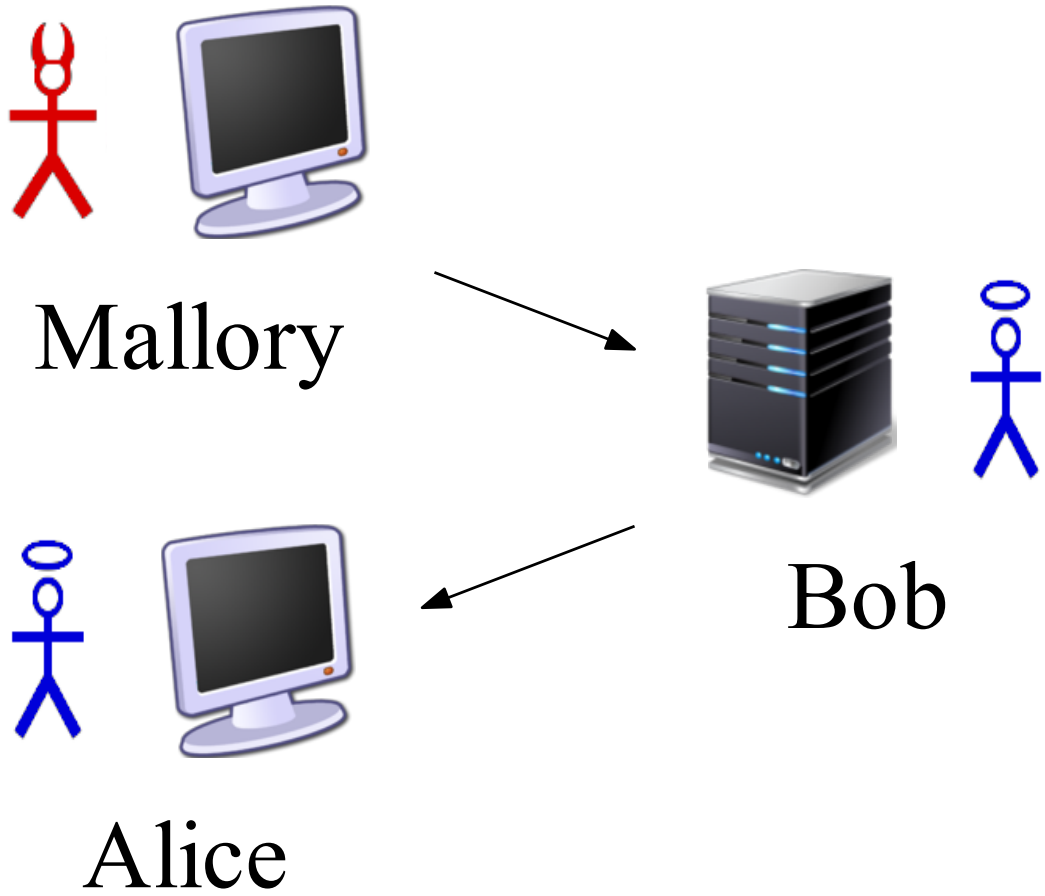
```
String query = "select * from users"
  + " where user_name = '" + name + "'"
  + " and password = '" + password + "';
```

- Parameterized interface

```
String query = "select * from users"
  + " where user_name = ?"
  + " and password = ?";
PreparedStatement st = con.prepareStatement(query);
st.setString(1, name);
st.setString(2, password);
ResultSet rs = st.executeQuery();
```

- Escaping routines
- White list validation

2. Cross-site scripting (XSS)



```
<div>
  ${hotel.description}
</div>
```

```
<div>
  ${hotel.description}
</div>
```

```
<div>
  <c:out value="${hotel.description}" />
</div>
```

3. Authentication and Session Management

Session Stealing

`http://host.com/page;jsessionid=dfd4fa35df2...`

cross-site scripting (XSS)

Session Stealing

`http://host.com/page;jsessionid=dfd4fa35df2...`

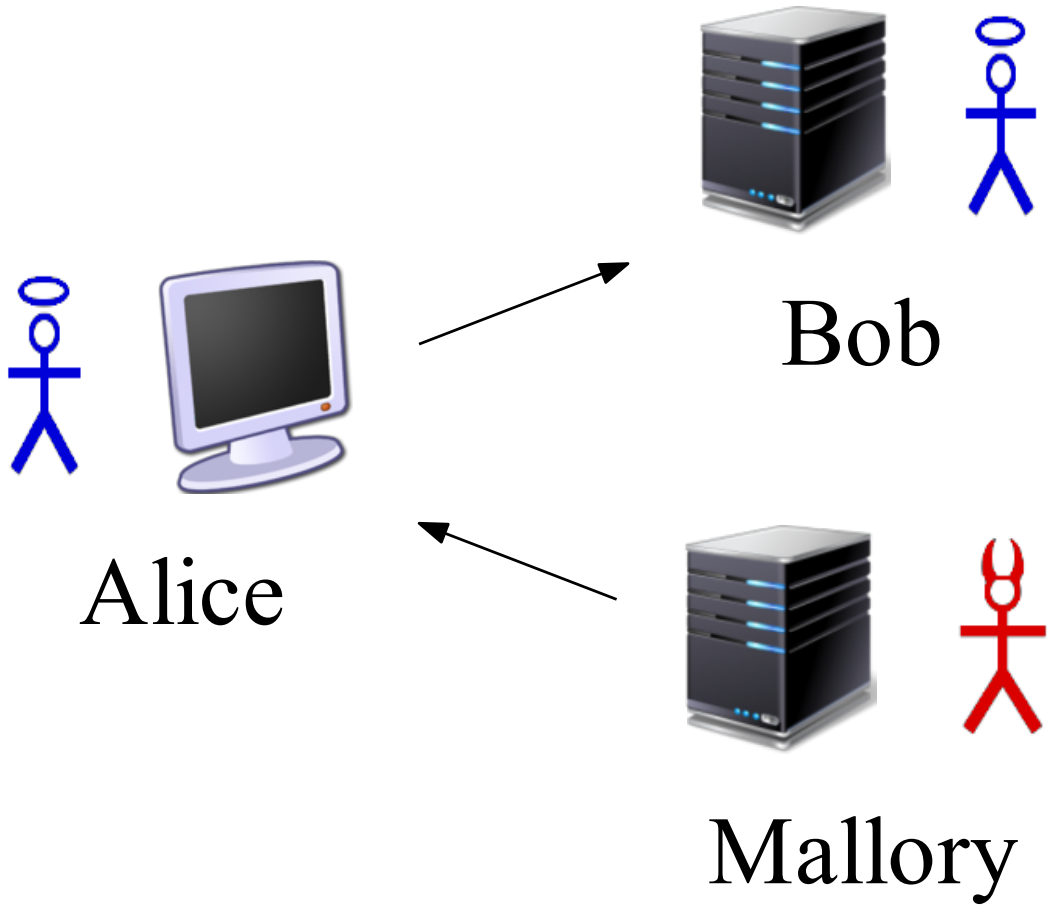
cross-site scripting (XSS)

Good authentication mechanism

`http://myhotel.com/user/152`

Check permissions

5. Cross-site Request Forgery (CSRF)

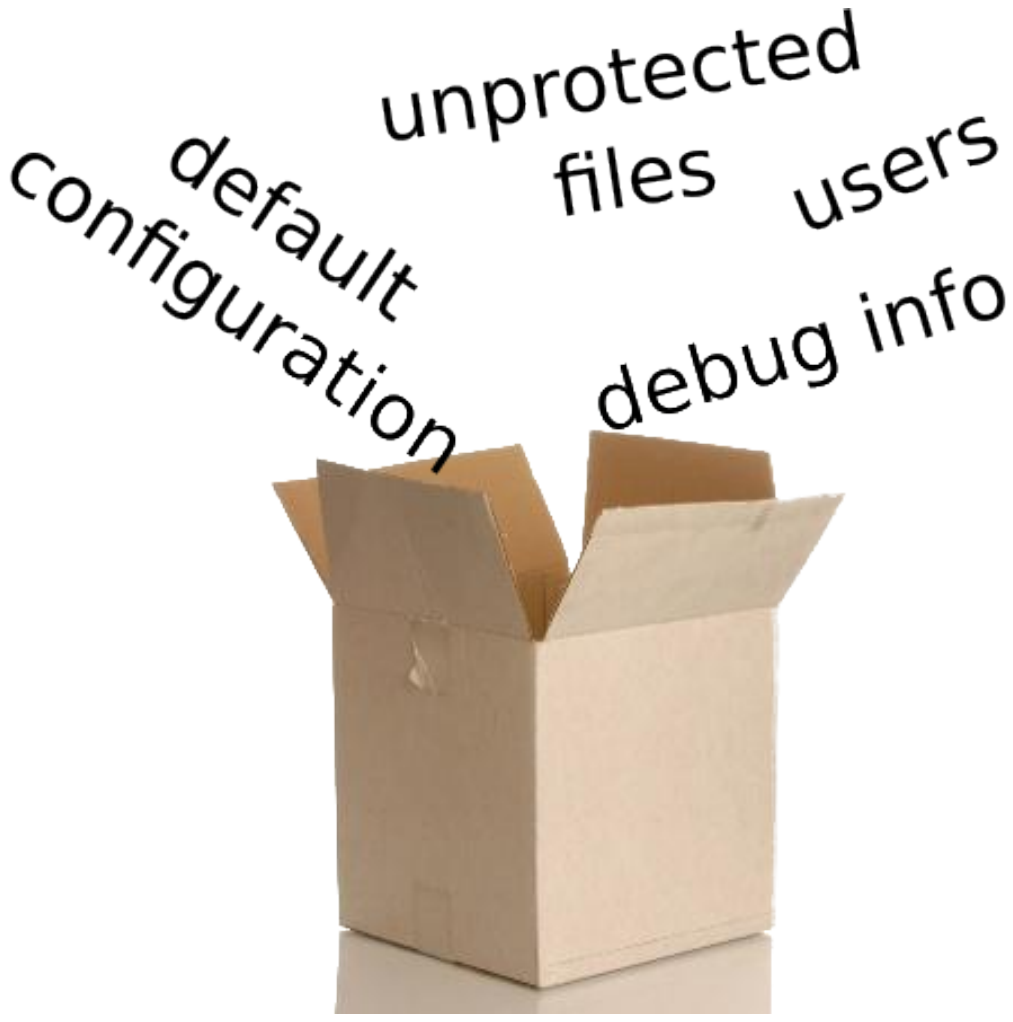


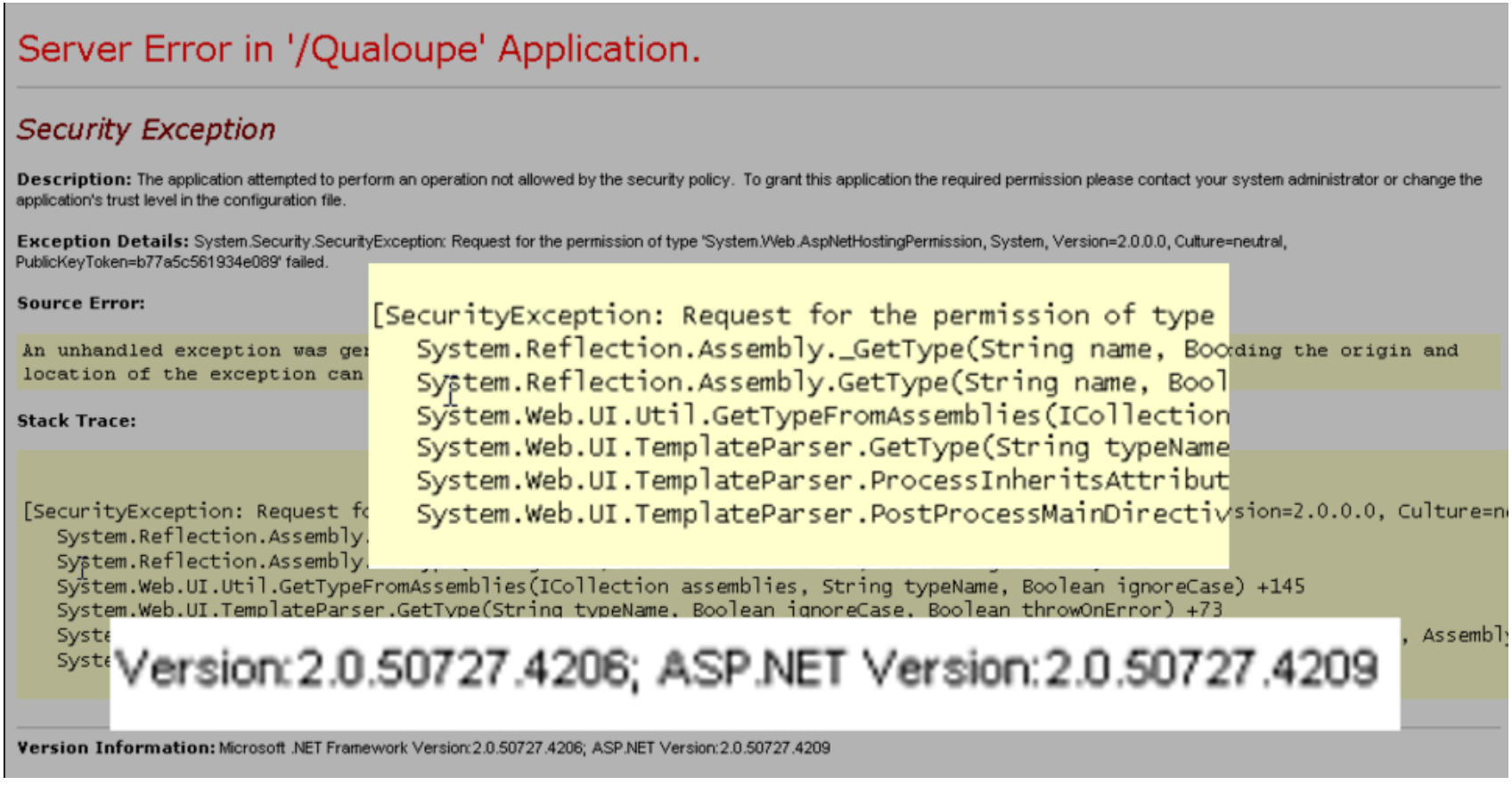
```
<img src="http://host.com/forum/thread1/
addComment?text=hello%20world" alt="" />
```

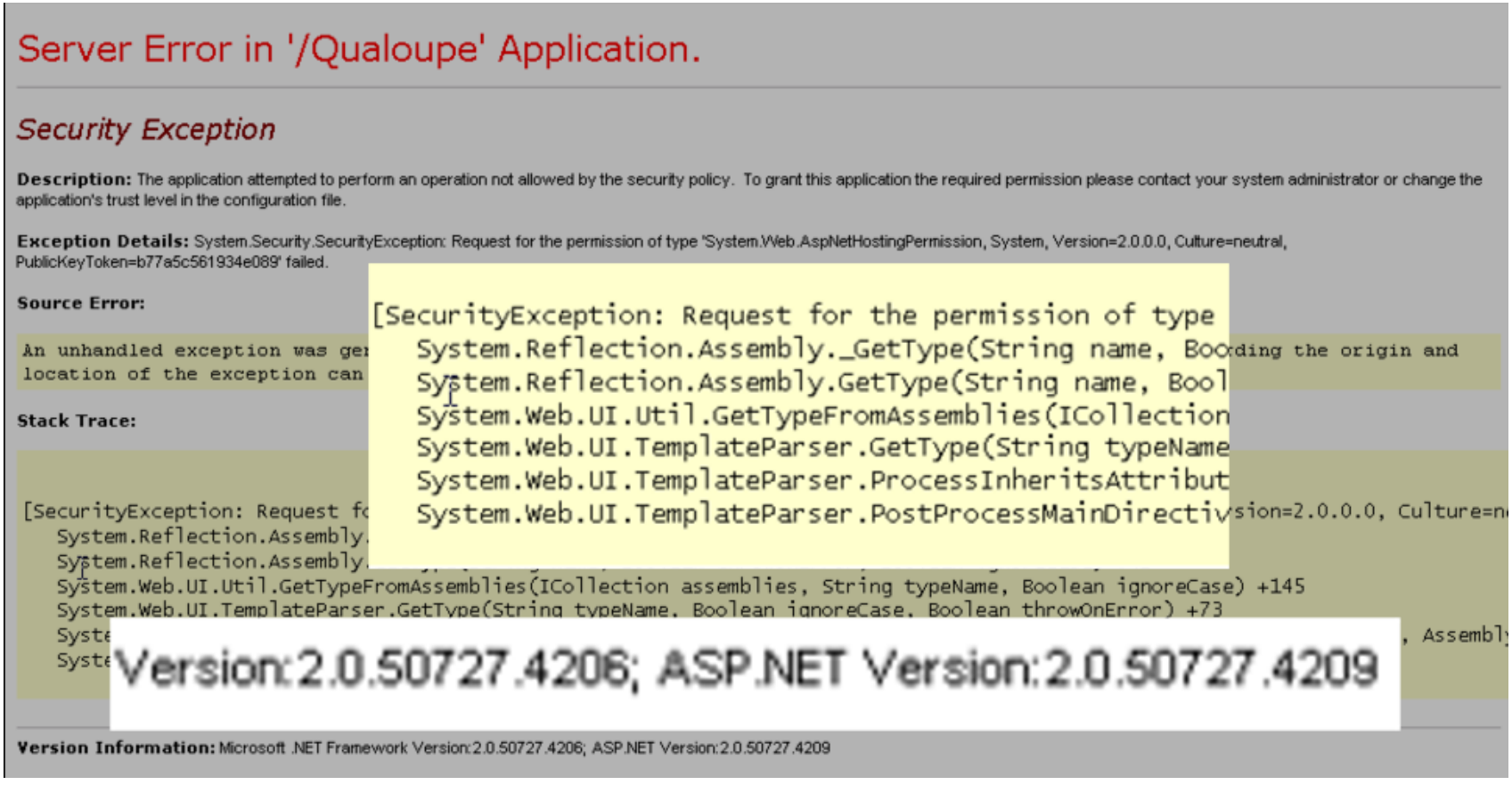
```
<img src="http://host.com/forum/thread1/
addComment?text=hello%20world" alt="" />
```

```
<form method="post" action="addComment">
  <input type="hidden" name="token"
      value="sd5646sdfse8wd" />
  ...
</form>
```

6. Security Misconfiguration


unprotected
files
users
default
configuration
debug info

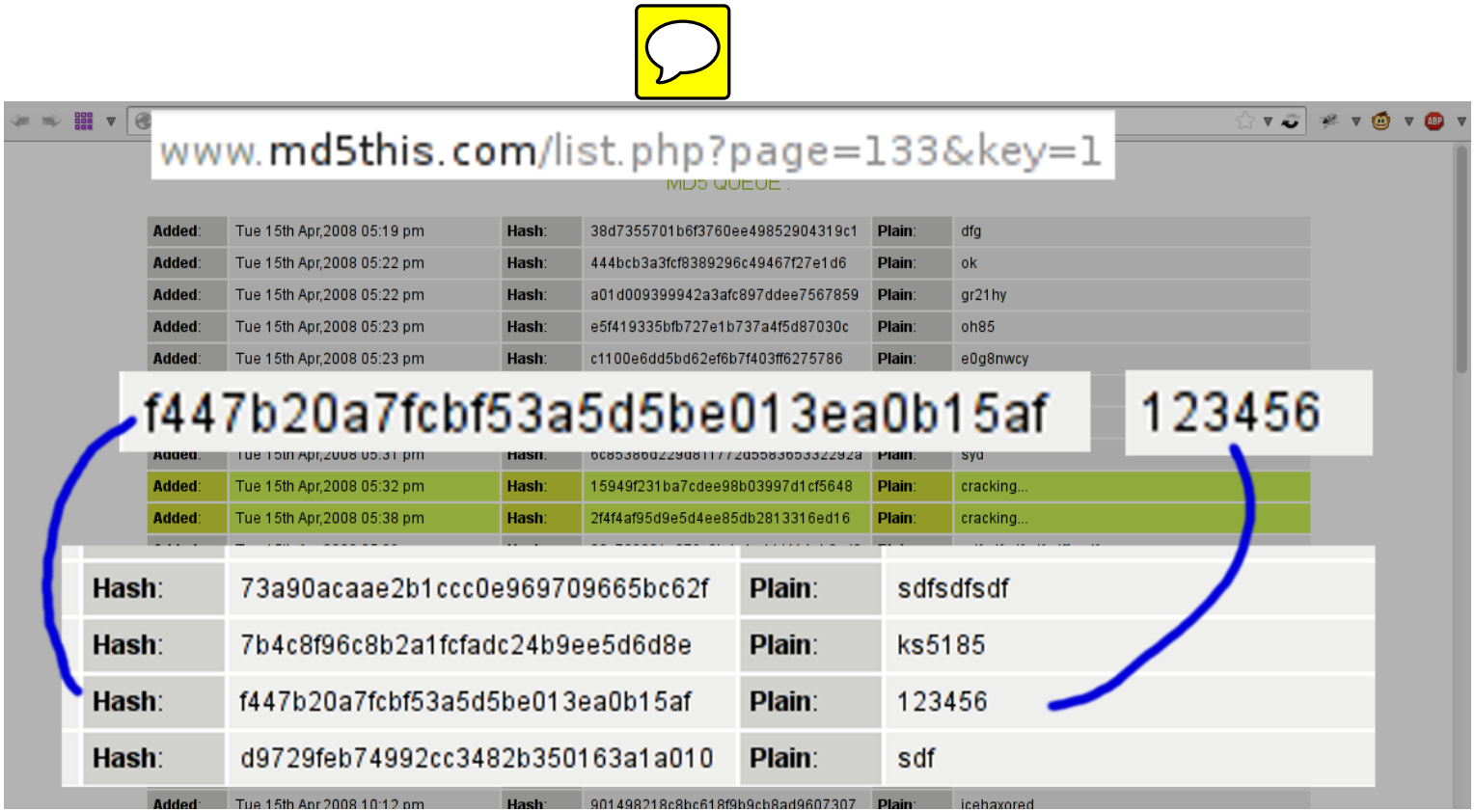

Server Error in '/Qualoupe' Application.
Security Exception
Description: The application attempted to perform an operation not allowed by the security policy. To grant this application the required permission please contact your system administrator or change the application's trust level in the configuration file.
Exception Details: System.Security.SecurityException: Request for the permission of type 'System.Web.AspNetHostingPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089' failed.
Source Error:
An unhandled exception was ge
location of the exception can
ding the origin and
Stack Trace:
[SecurityException: Request fo
System.Reflection.Assembly.
System.Reflection.Assembly.
System.Web.UI.Util.GetTypeFromAssemblies(ICollection assemblies, String typeName, Boolean ignoreCase) +145
System.Web.UI.TemplateParser.GetType(String typeName, Boolean ignoreCase, Boolean throwOnError) +73
Syste
Syste
[SecurityException: Request for the permission of type
System.Reflection.Assembly._GetType(String name, Boo
System.Reflection.Assembly.GetType(String name, Bool
System.Web.UI.Util.GetTypeFromAssemblies(ICollection
System.Web.UI.TemplateParser.GetType(String typeName
System.Web.UI.TemplateParser.ProcessInheritsAttribut
System.Web.UI.TemplateParser.PostProcessMainDirectiv
sion=2.0.0.0, Culture=n
, Assembly
Version:2.0.50727.4206; ASP.NET Version:2.0.50727.4209
Version Information: Microsoft .NET Framework Version:2.0.50727.4206; ASP.NET Version:2.0.50727.4209



... more information for the attacker

7. Insecure Cryptographic Storage

ID	NAME	PASSWORD	...
1	admin	2ezf5zf2d	...
2	jacky	toto	...
3	k	123456	...


www.md5this.com/list.php?page=133&key=1
f447b20a7fcbf53a5d5be013ea0b15af
123456
Hash: 73a90acaae2b1ccc0e969709665bc62f Plain: sdfsdfsdf
Hash: 7b4c8f96c8b2a1fcfadc24b9ee5d6d8e Plain: ks5185
Hash: f447b20a7fcbf53a5d5be013ea0b15af Plain: 123456
Hash: d9729feb74992cc3482b350163a1a010 Plain: sdf


BACKUP

8. Failure to Restrict URL Access

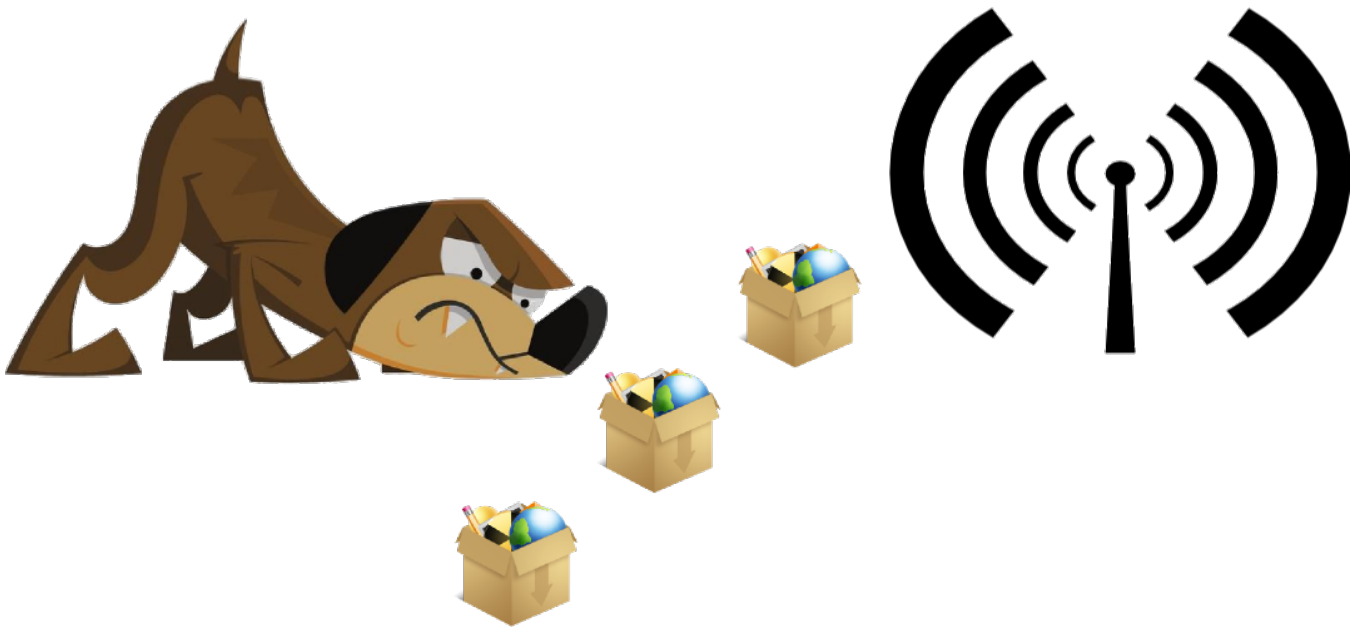
`http://myhotel.com/admin`

8. Failure to Restrict URL Access

`http://myhotel.com/admin`

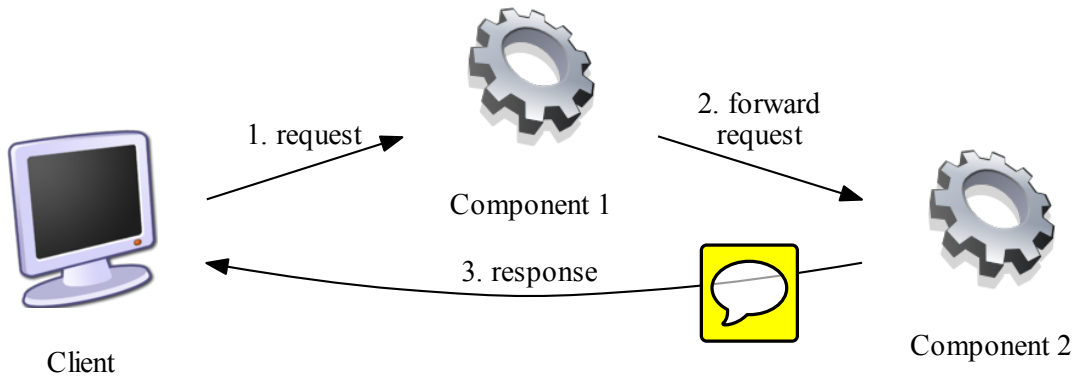
Authorization

9. Insufficient Transport Layer Protection



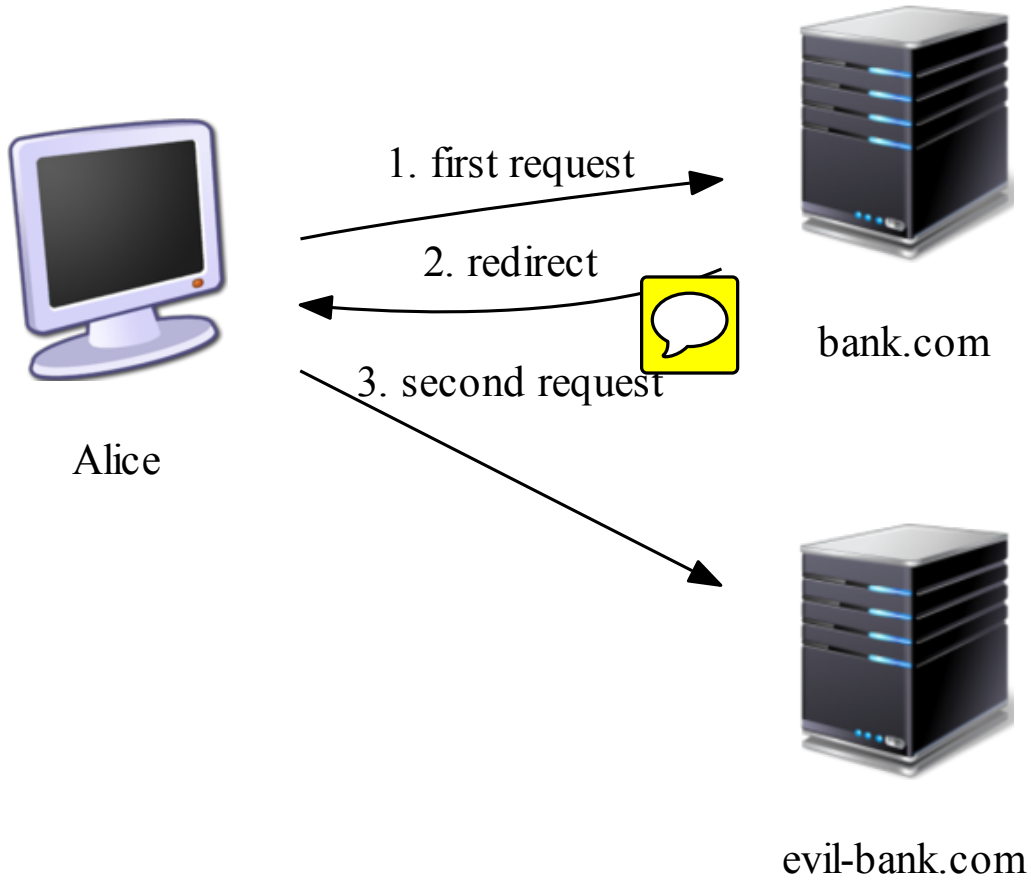

https://

10. Unvalidated **Forwards** and Redirects




```
http://host.com/forward.jsp?
fwd=admin.jsp
```

10. Unvalidated Forwards and **Redirects**



`https://bank.com/secured/../redirect?url=http://evil-bank.com/secured`

Others attacks

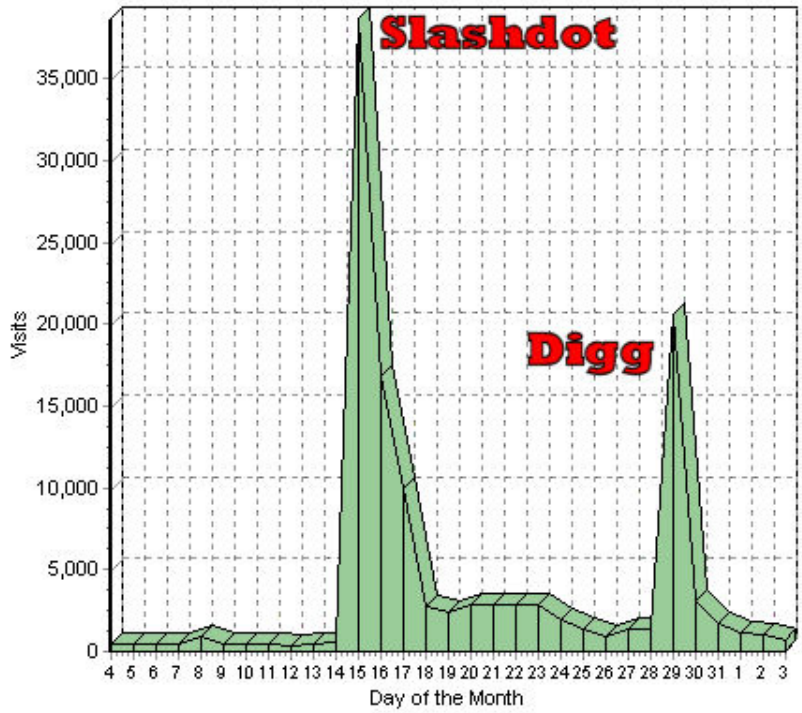
1. Denial of Service
2. Man in the Middle
3. Social Engineering
4. Phishing
5. Brute Force
6. And so on...

What about just crashing the system?

Denial of Service (DoS)

DoS attack types

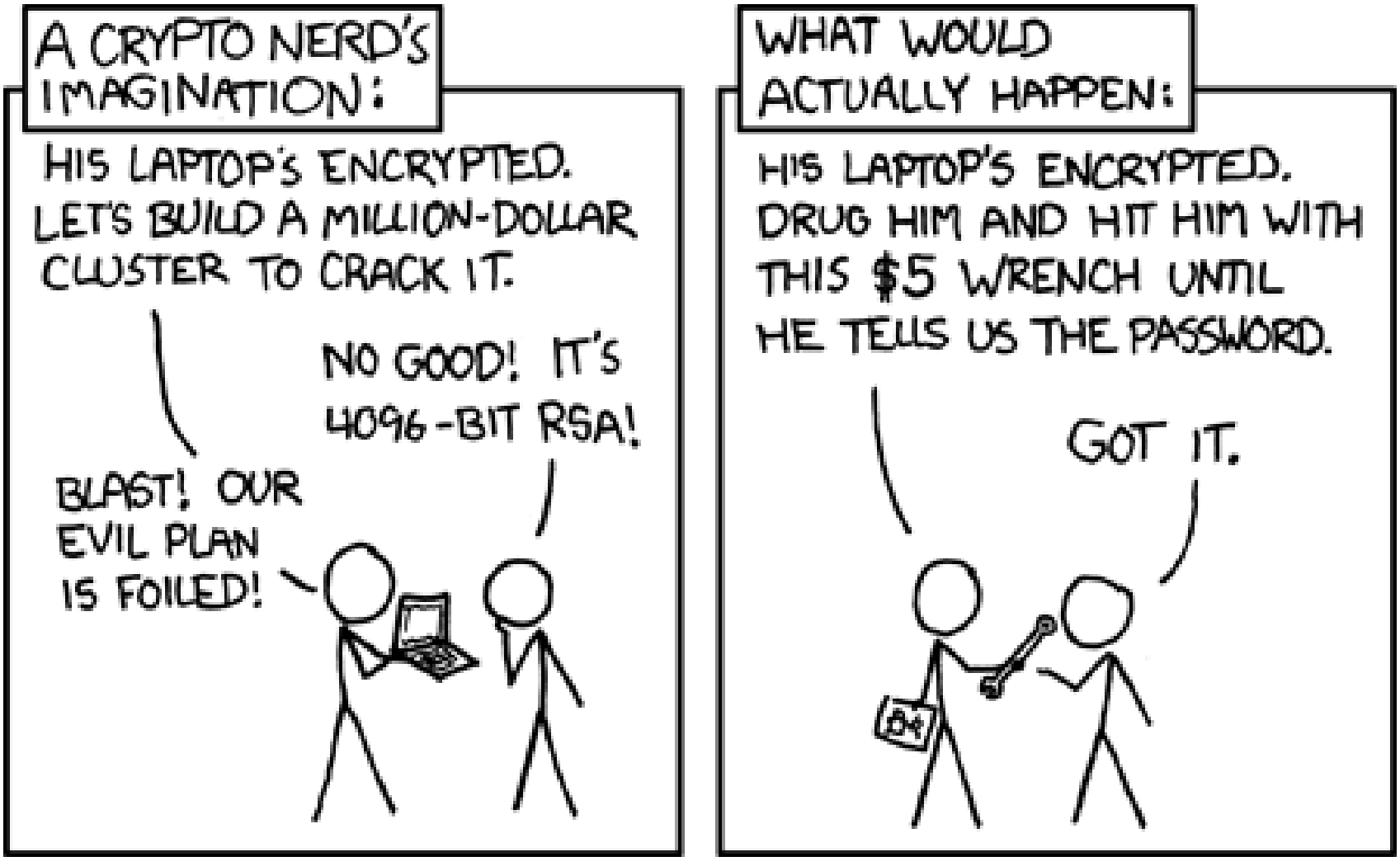
DoS attack types

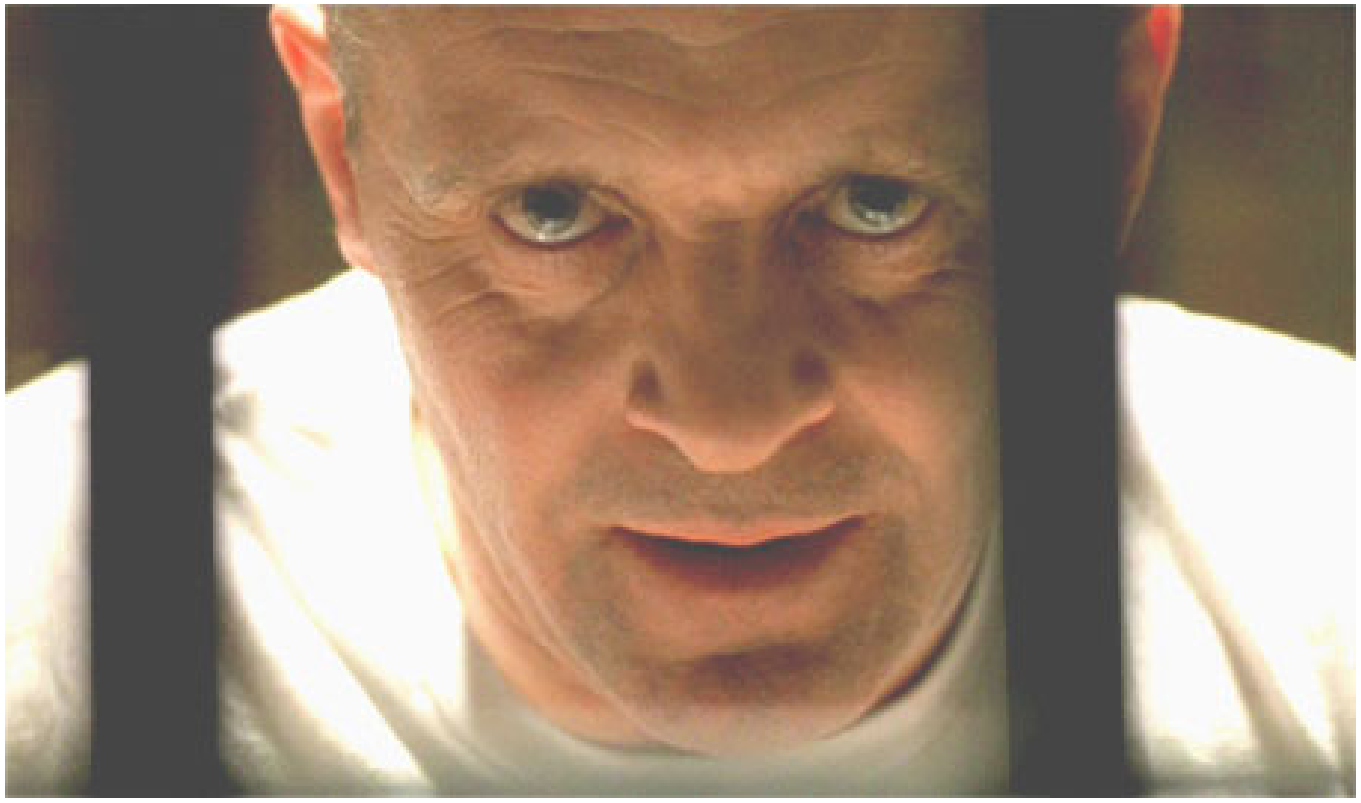


And altering the content?

Man in the Middle attack

Social engineering


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.



Quid pro quo, Clarice...

Phishing

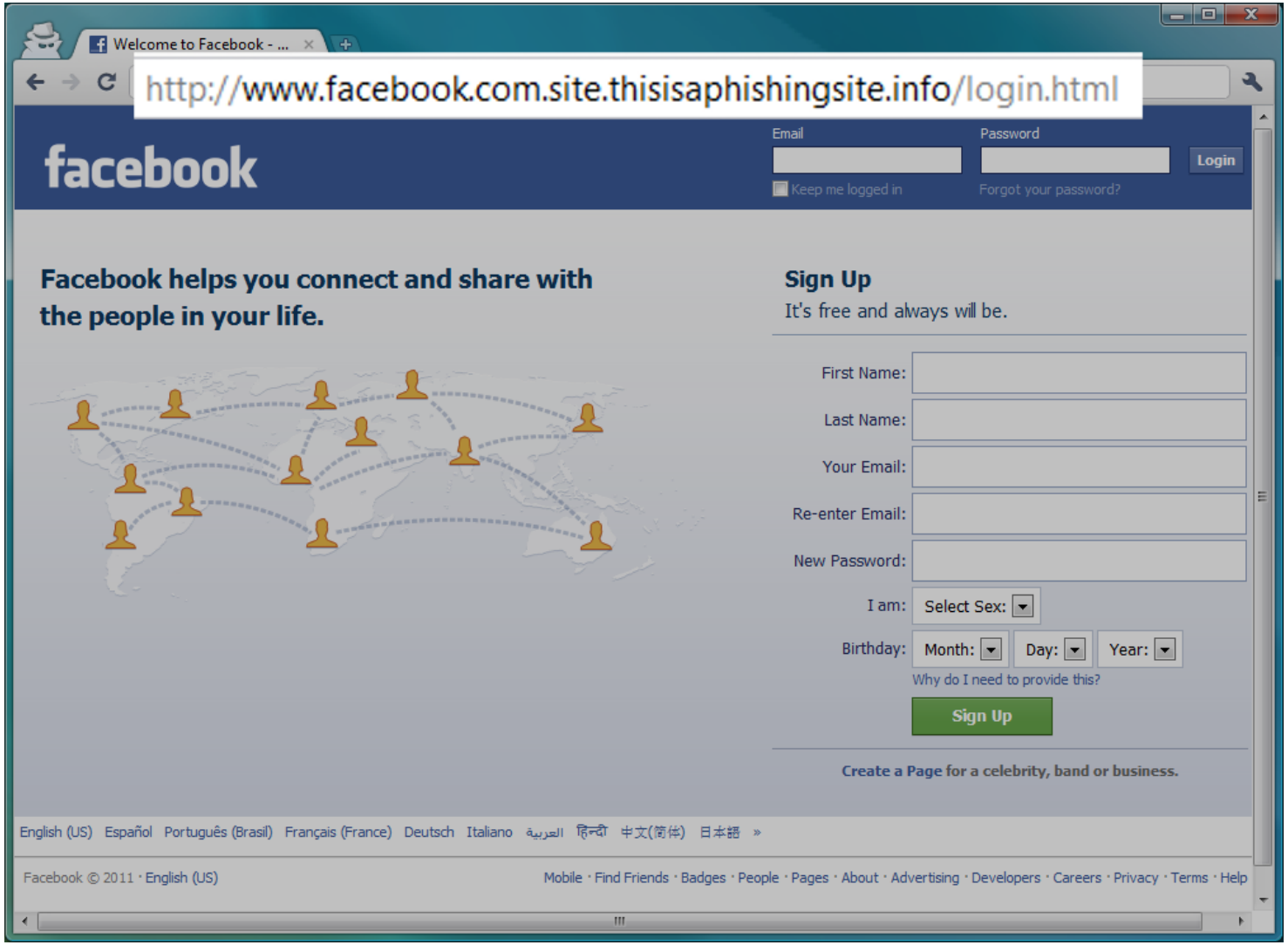
Subject: URGENT - Your bank card has been blocked
From: "Banking Service" <bankservice@service.fr>
Date: Thu, April 7, 2011 3:38 pm
To: [redacted]
Priority: Normal
Options: View Full Header | View Printable Version | Download this as a file | View as plain text

Card Blocked,

following an anormal activity and by measure security, your bank card has been blocked on the websites displaying the logo Verified by Visa, MasterCard SecureCode and in all Automated teller machine (ATM).
In order to be able to buy in all serenity on these sites, and to use your bank card for cash withdrawal, we invite you to fill out a form in a secure manner using the link below:

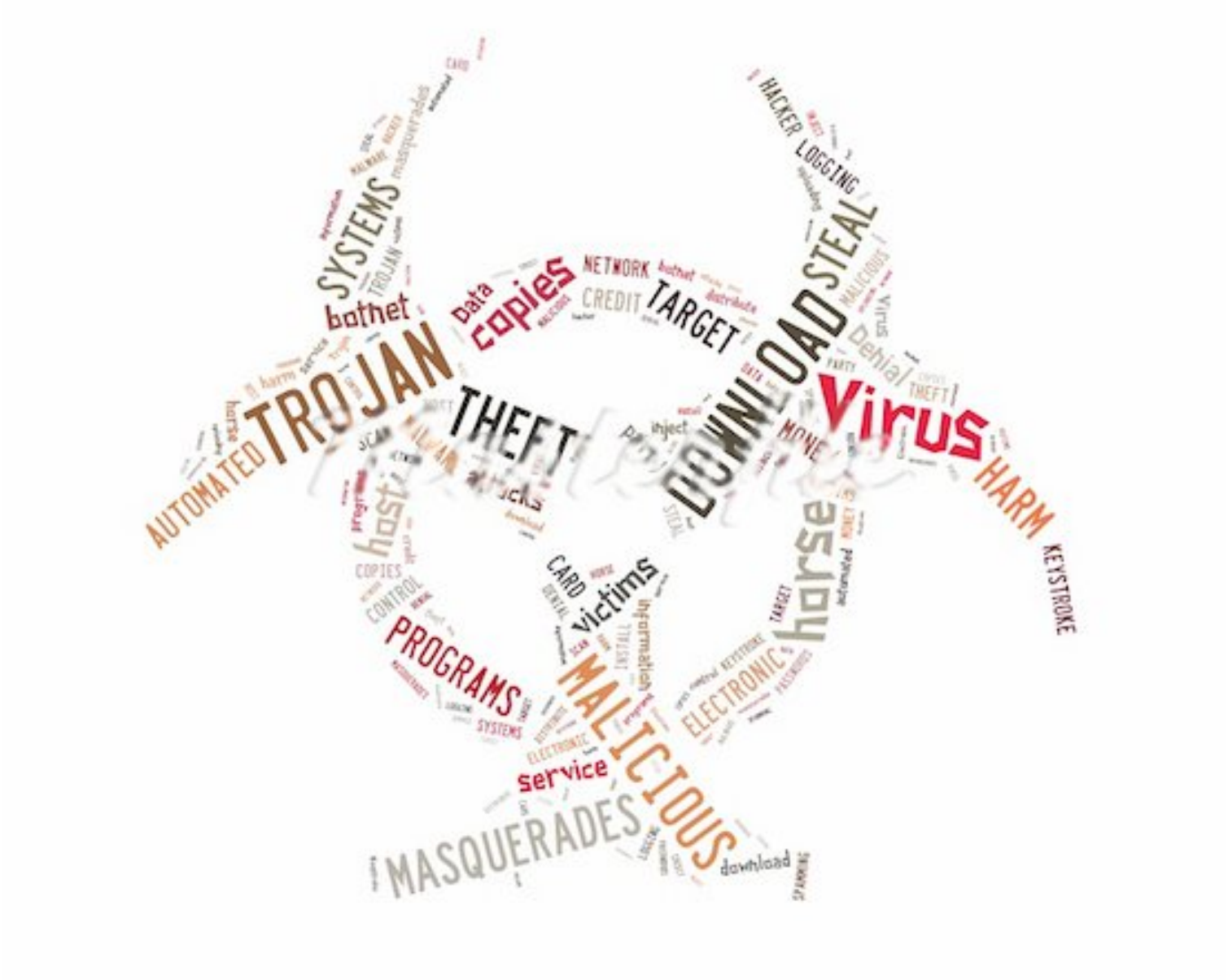
Access to your form.

Cordially,
Customer Service.


Welcome to Facebook - ...
http://www.facebook.com.site.thisisaphishingsite.info/login.html
facebook
Email
Password
Login
Keep me logged in
Forgot your password?
Facebook helps you connect and share with the people in your life.
Sign Up
It's free and always will be.
First Name:
Last Name:
Your Email:
Re-enter Email:
New Password:
I am:
Select Sex:
Birthday:
Month:
Day:
Year:
Why do I need to provide this?
Sign Up
Create a Page for a celebrity, band or business.
English (US) Español Português (Brasil) Français (France) Deutsch Italiano العربية हिन्दी 中文(简体) 日本語 »
Facebook © 2011 · English (US)
Mobile · Find Friends · Badges · People · Pages · About · Advertising · Developers · Careers · Privacy · Terms · Help

Brute force

And so on...


SYSTEMS
botnet
TROJAN
AUTOMATED
Data
copies
NETWORK
CREDIT
TARGET
THEFT
HACKER
LOGGING
STEAL
DOWNLOAD
Virus
HARM
KEYSTROKE
CARD
victims
PROGRAMS
MALICIOUS
service
MASQUERADES
download
horse
ELECTRONIC
CONTROL

Common points

Common points

- Improper input validation

Common points

- Improper input validation
- Missing authorization checks

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation

- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
  - Injection
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
  - Injection
  - Cross-site scripting (XSS)

- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
  - Injection
  - Cross-site scripting (XSS)
  - Session Stealing via XSS
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
  - Injection
  - Cross-site scripting (XSS)
  - Session Stealing via XSS
  - Cross-Site Request Forgery (CSRF)
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
  - Injection
  - Cross-site scripting (XSS)
  - Session Stealing via XSS
  - Cross-Site Request Forgery (CSRF)
  - Unvalidated Forwards and Redirects
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks

- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
  - Insecure Direct Object References
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
  - Insecure Direct Object References
  - Failure to Restrict URL Access
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions

- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
  - Insecure Cryptographic Storage

- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
  - Insecure Cryptographic Storage
  - Insufficient Transport Layer Protection

- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
  - Insecure Cryptographic Storage
  - Insufficient Transport Layer Protection
  - Denial of Service
- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
  - Insecure Cryptographic Storage
  - Insufficient Transport Layer Protection
  - Denial of Service
  - Man in the Middle
- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)
  - Session stealing (cybercafe)

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)
  - Session stealing (cybercafe)
  - Social engineering

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)
  - Session stealing (cybercafe)
  - Social engineering
  - Phishing

Common points

- Improper input validation
- Missing authorization checks
- Misconfiguration
- Other assumptions
- Education factor (social engineering)
  - Session stealing (cybercafe)
  - Social engineering
  - Phishing
  - (Brute Force)

Prevention

Different moments

- Architecture
- Development
- Maintenance

Architecture

The highest-level breakdown of a system into its parts; the decisions that are hard to change; there are multiple architectures in a system; what is architecturally significant can change over a system's lifetime; and, in the end, architecture boils down to whatever the important stuff is.

– Martin Fowler, Patterns of Enterprise Application Architecture

Security design...

Development: tips and tricks

- Passwords

Development: tips and tricks

- Passwords
- Session management

Development: tips and tricks

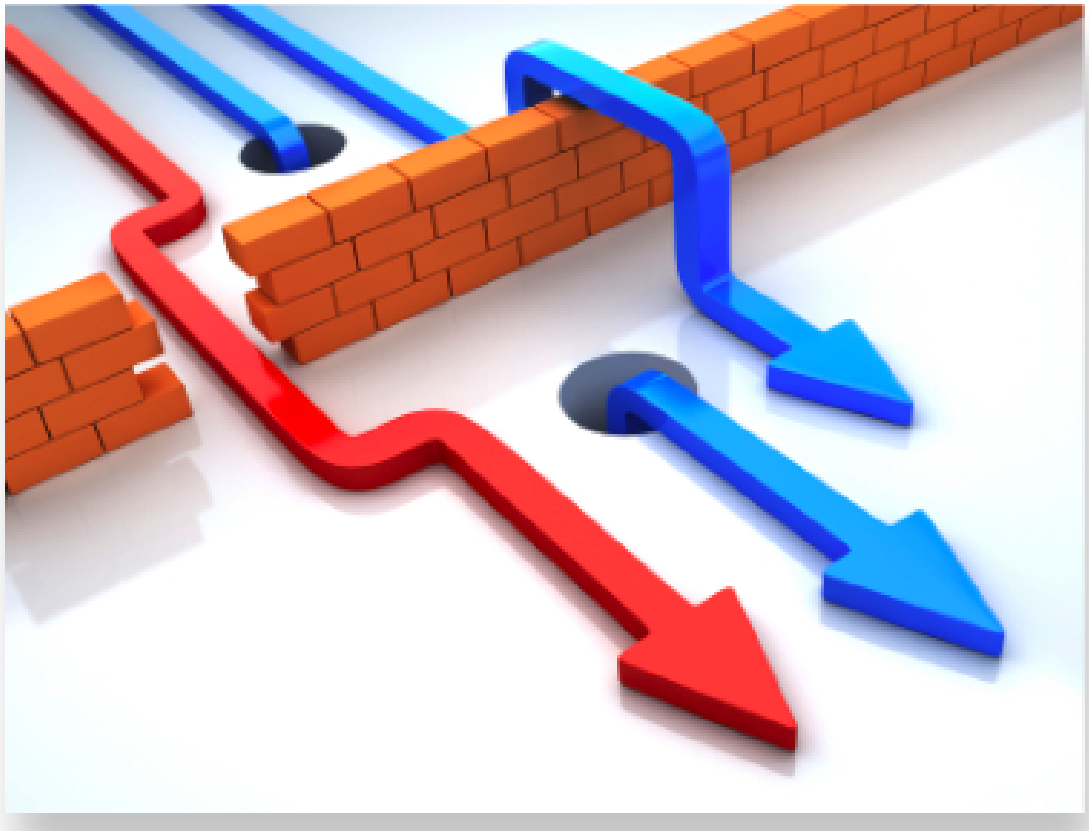
- Passwords
- Session management
- Input validation

Development: tips and tricks

- Passwords
- Session management
- Input validation
- Ad hoc protection

Secure development cycle

Penetration testing



Tools - Vulnerabilities detector

http://www.metasploit.com/

http://www.tenable.com/

Tools - Network

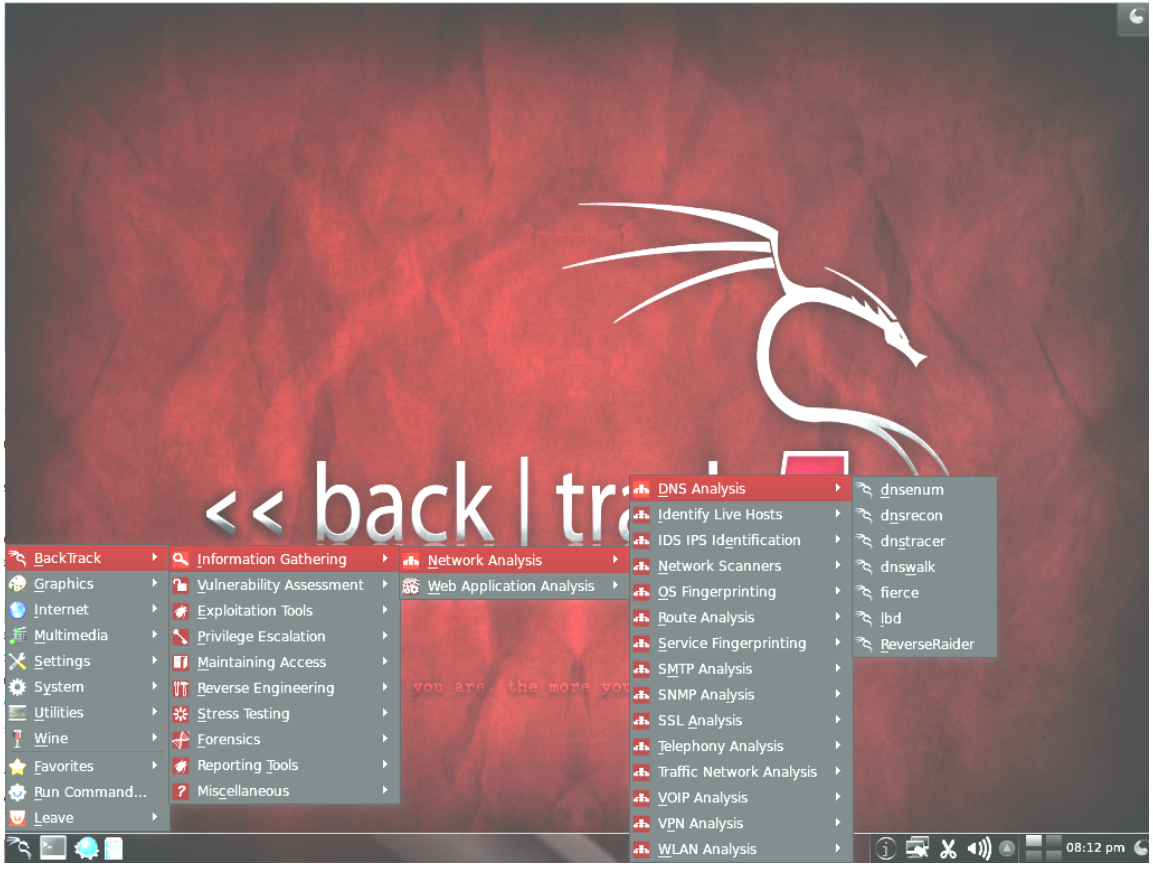
NMAP

`http://nmap.org/`

WebScarab

`https://www.owasp.org/`

Tools - All in one



BackTrack

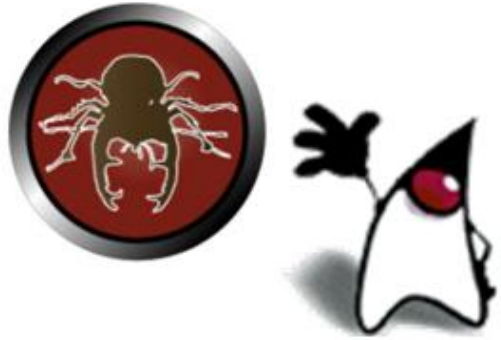
`http://www.backtrack-linux.org/`

Maintenance

Frameworks

Your application is not the first.

Frameworks - examples



ESAPI

https://www.owasp.org/

http://www.springsource.org/

https://shiro.apache.org/

Microsoft®

.NET

https://www.microsoft.com/net

Security in application servers

Application server

The program that runs the web application

- Authentication management
- Directory listing disabling
- User session management
- Error handling
- Input encoding

Web Application Firewall

Secure protocols

SSL **TLS**

Browser restrictions

XHR

XMLHttpResponse
Access control

Sandboxing

Flash

SSL

JavaScript

E-mail

Variety of environments

Detection and Analysis

Honeypot

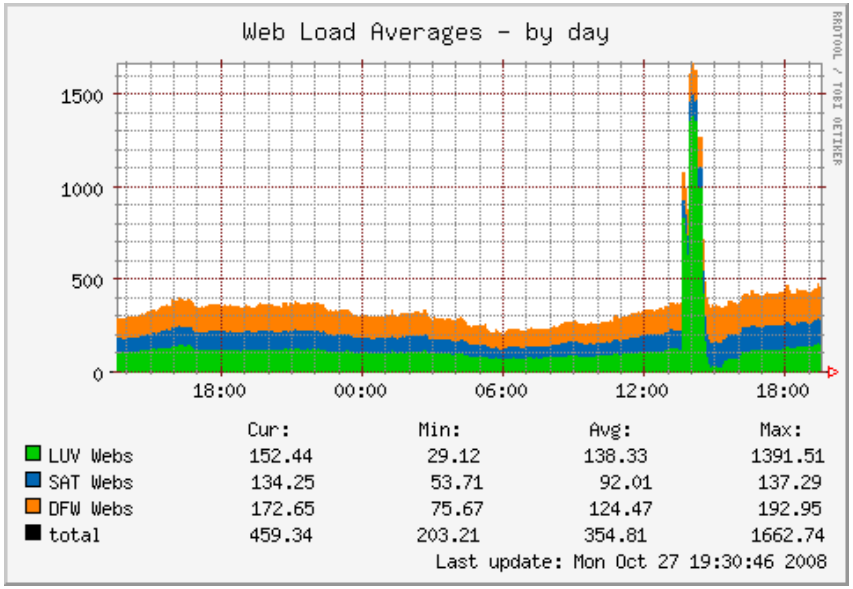
Network Intrusion Detection System

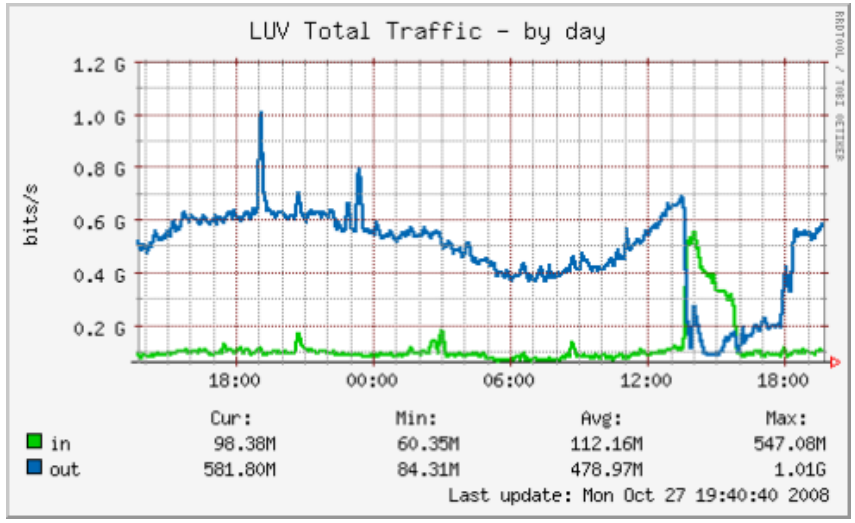
NIDS - detection types

- Signature-based detection
- Statistical anomaly-based detection
- Stateful protocol analysis

After an attack

How to know?


Web Load Averages - by day
1500
1000
500
0
18:00
00:00
06:00
12:00
18:00
Cur: Min: Avg: Max:
LUV Webs 152.44 29.12 138.33 1391.51
SAT Webs 134.25 53.71 92.01 137.29
DFW Webs 172.65 75.67 124.47 192.95
total 459.34 203.21 354.81 1662.74
Last update: Mon Oct 27 19:30:46 2008

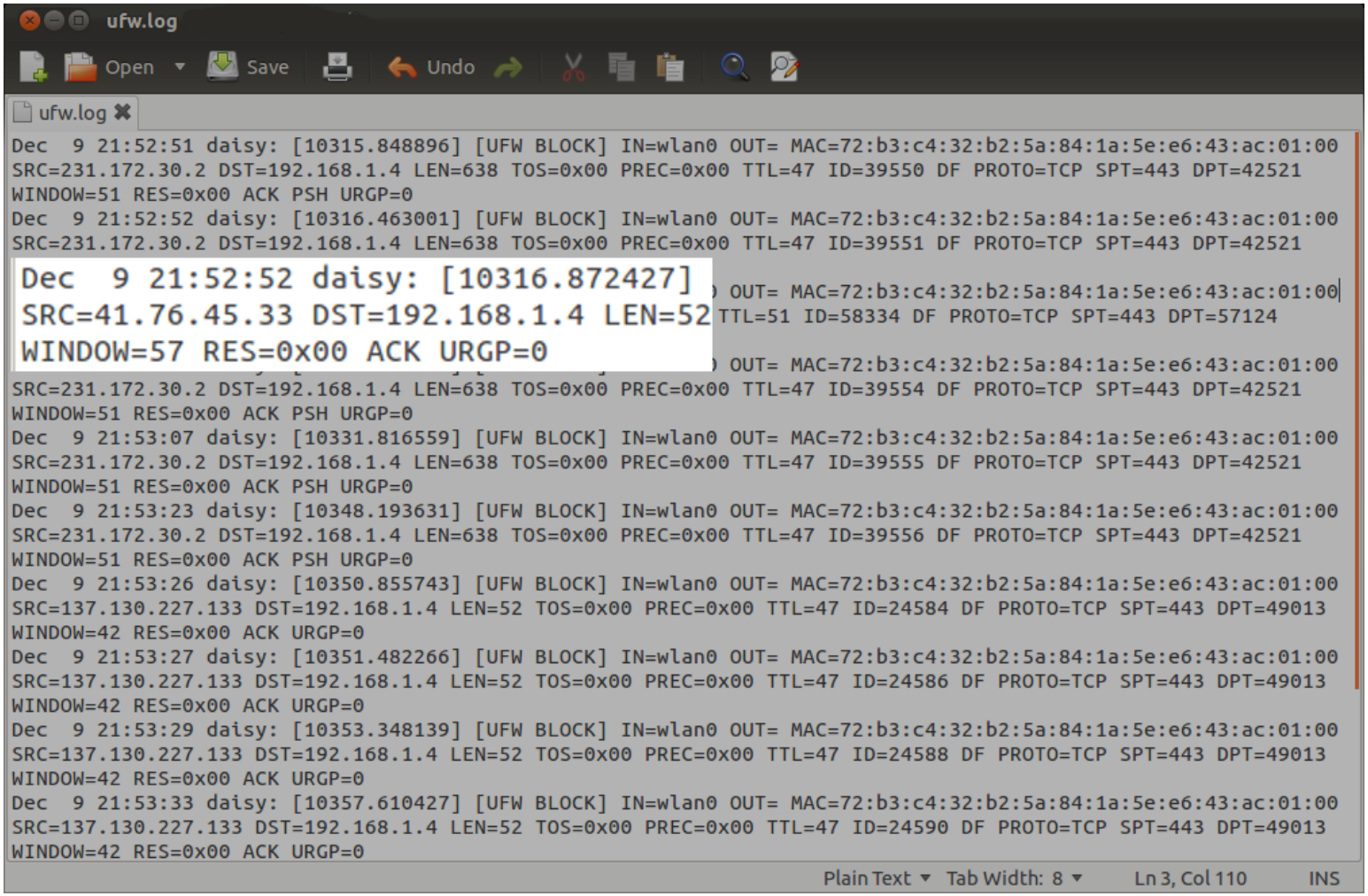

LUV Total Traffic - by day
bits/s
1.2 G
1.0 G
0.8 G
0.6 G
0.4 G
0.2 G
18:00
00:00
06:00
12:00
18:00
Cur: Min: Avg: Max:
in 98.38M 60.35M 112.16M 547.08M
out 581.80M 84.31M 478.97M 1.01G
Last update: Mon Oct 27 19:40:40 2008

Analysis - tools

WebLog Expert

AWStats

Noise as a diversion



Performances

Conclusion


protection
safeguard
identification
datacenter
communication
firewall
fingermark
software
safe
key
shield
access
industry
camera
datacenter key
protection
network
digital
net
detection
security
secret
lock
scan
safety
password
camera
verification
danger
identify
security
internet
data
verification
access
information
firewall
code
access
guard
privacy
padlock
digital
risk
secret
lock
communication
safeguard
identify
web


Thanks

HERMES