Seminar on Secure Software

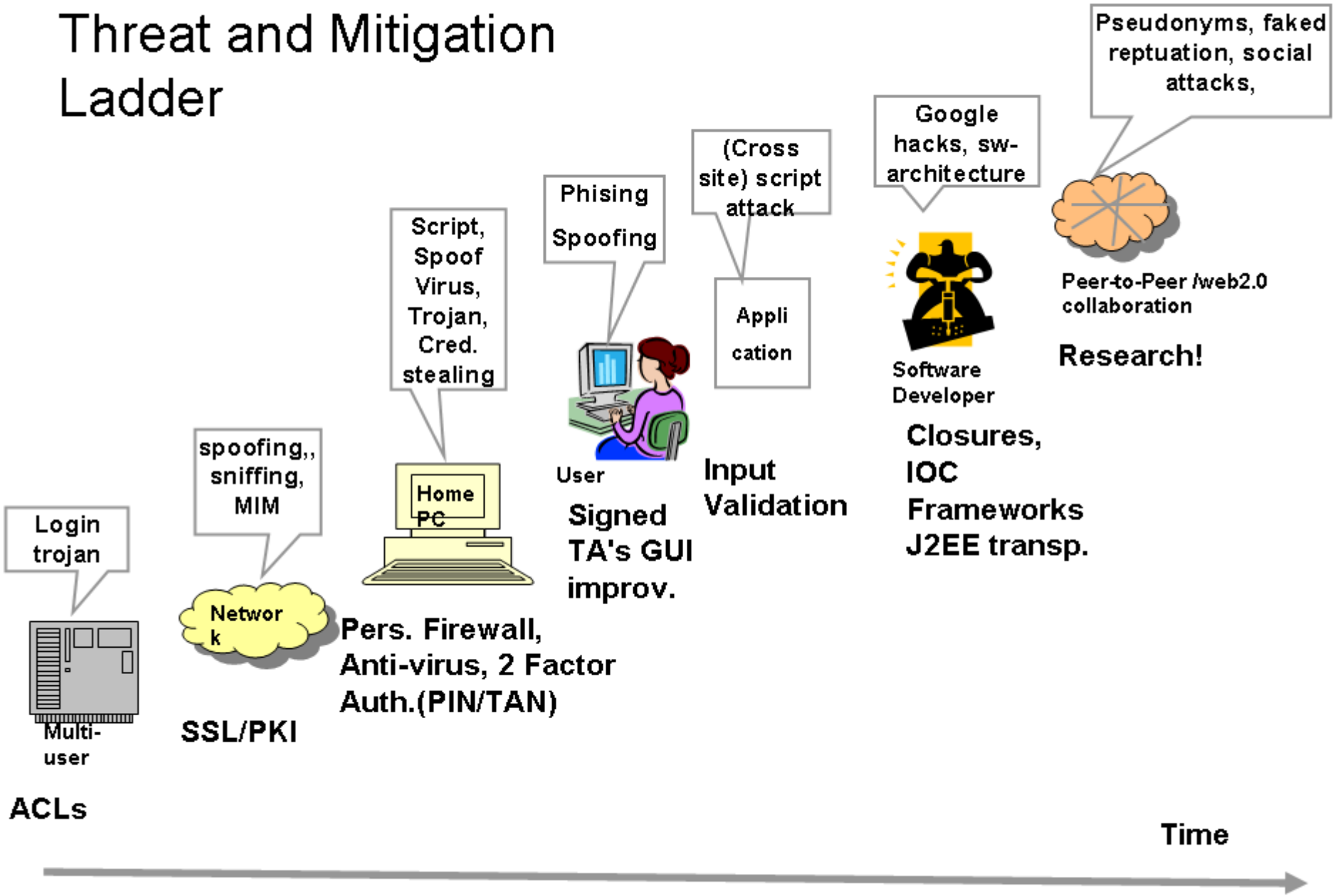
Attacks, Mitigation and fundamental software problems

Input Validation, Filtering and Damage Control as Software Mechanisms

Attack Examples

XSS, XSRF, Buffer Overflows,
Character Aliases etc.

Threat and Mitigation Ladder



Input/Output Related

A1 Unvalidated Input
A4 Cross Site Scripting
A5 Buffer Overflow
A6 Injection Flaws
A7 Improper Error Handling
A9 Application Denial of Service

AAA related

A2 Broken Access Control
A3 Broken Authentication and Session Management
A9 Application Denial of Service

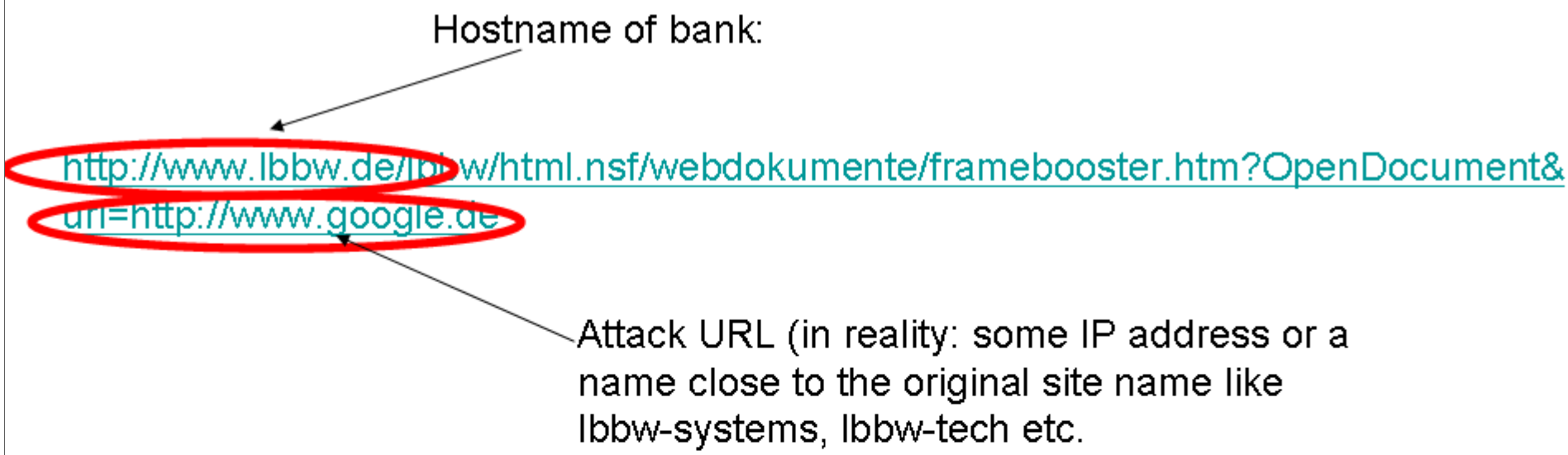
Infrastructure

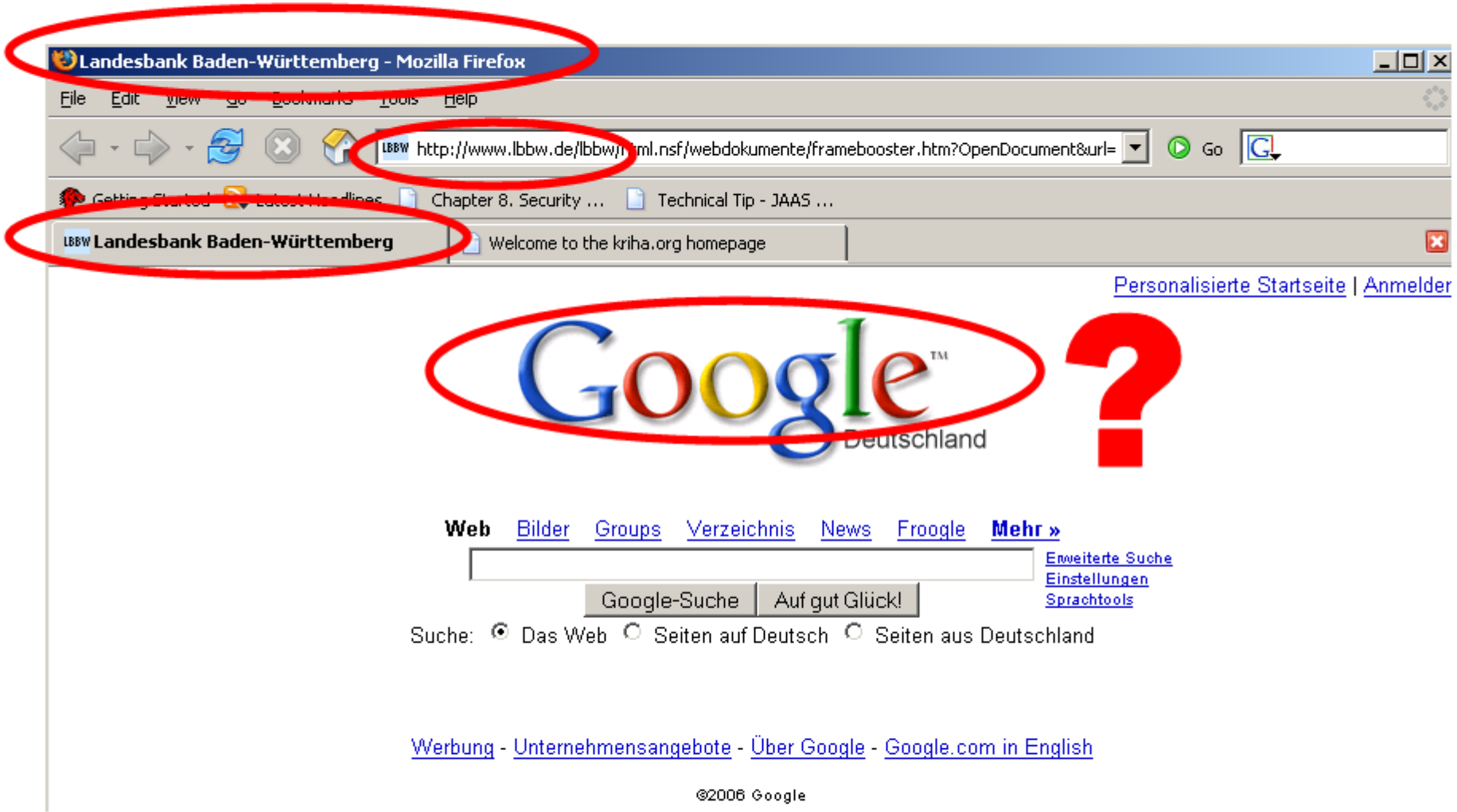
A8 Insecure Storage
A9 Application Denial of Service
A10 Insecure Configuration Management

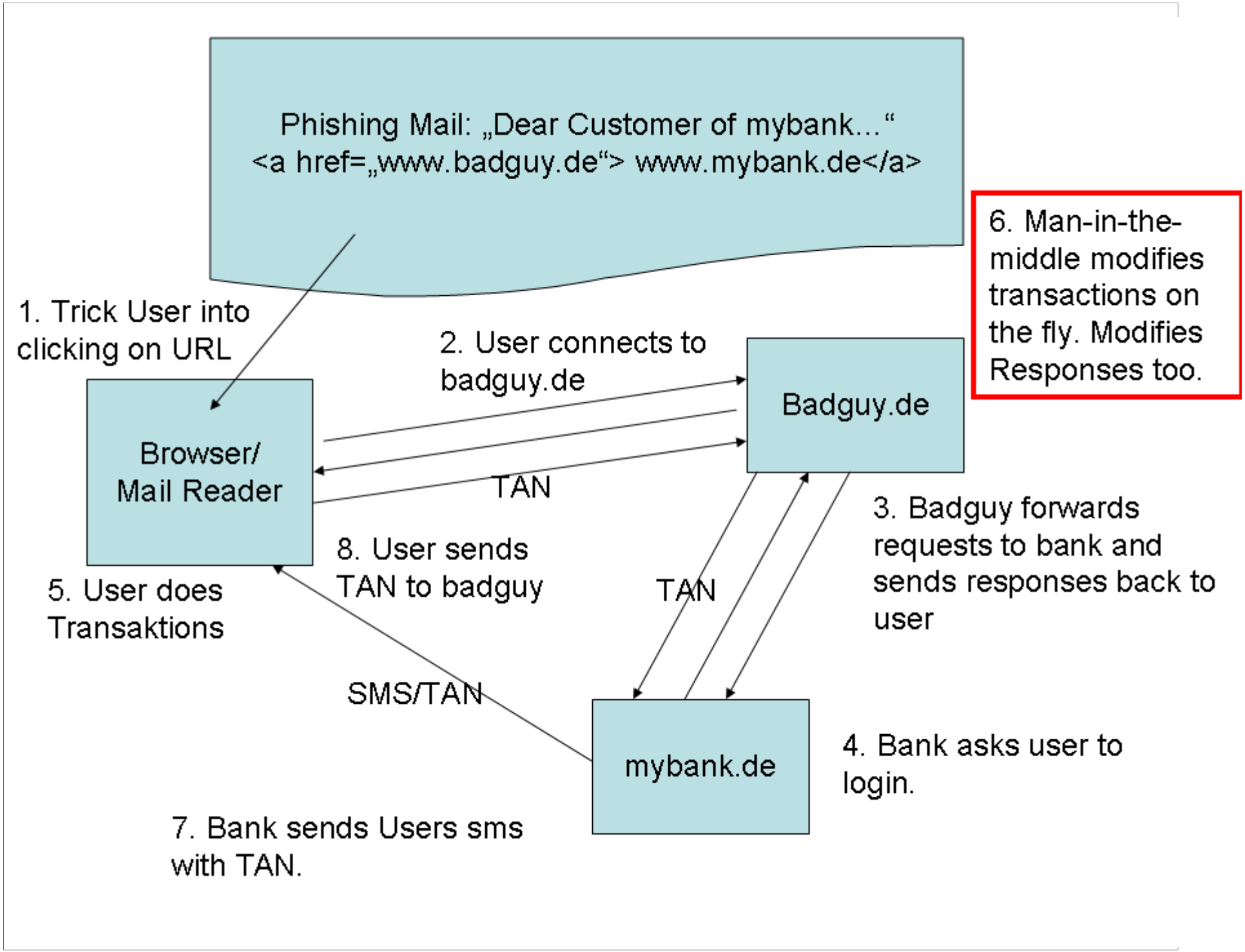
System Engineering

A9 Application Denial of Service

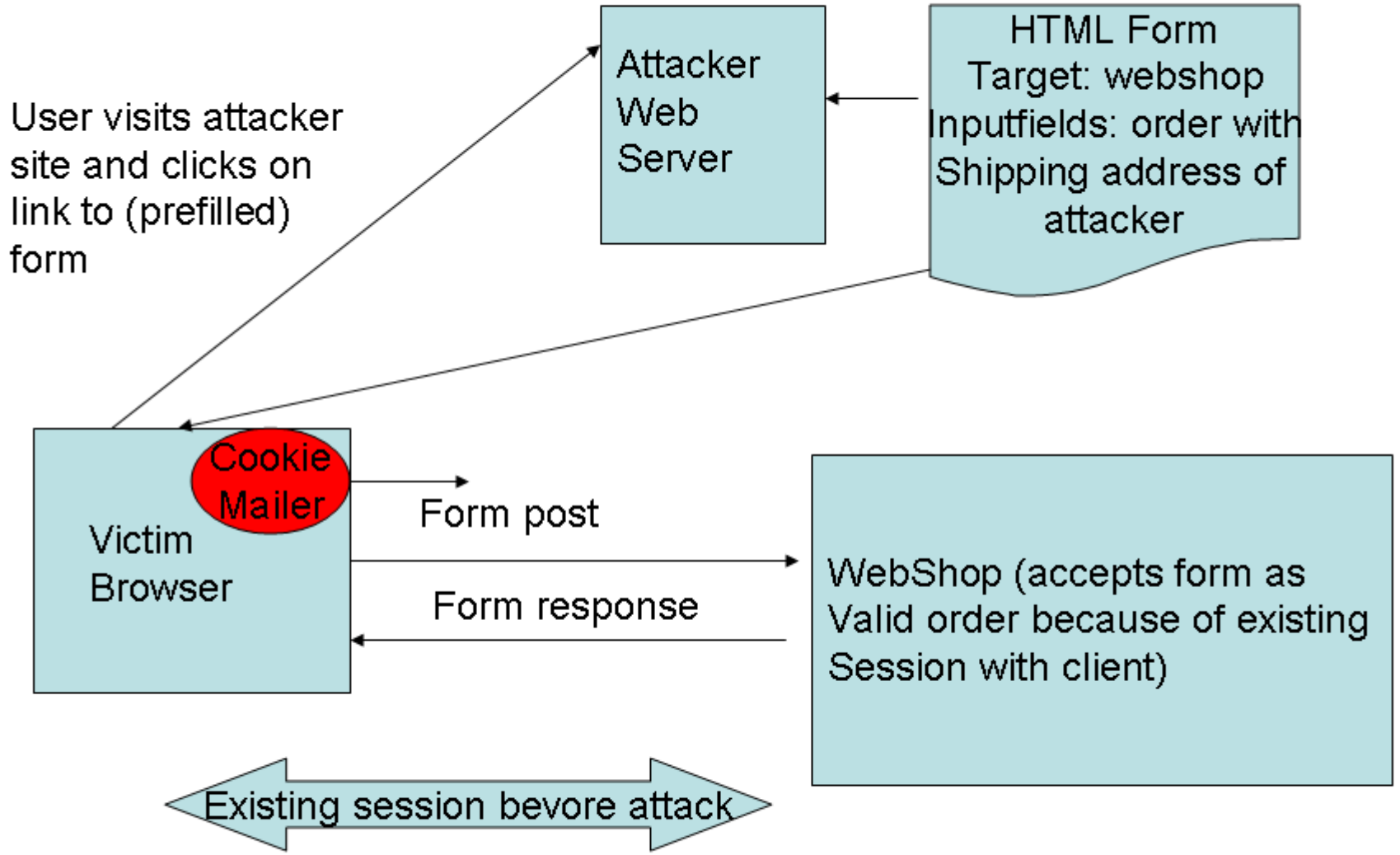
A "Phishing-Link" to LBBW Bank: XSS due to bad input validation



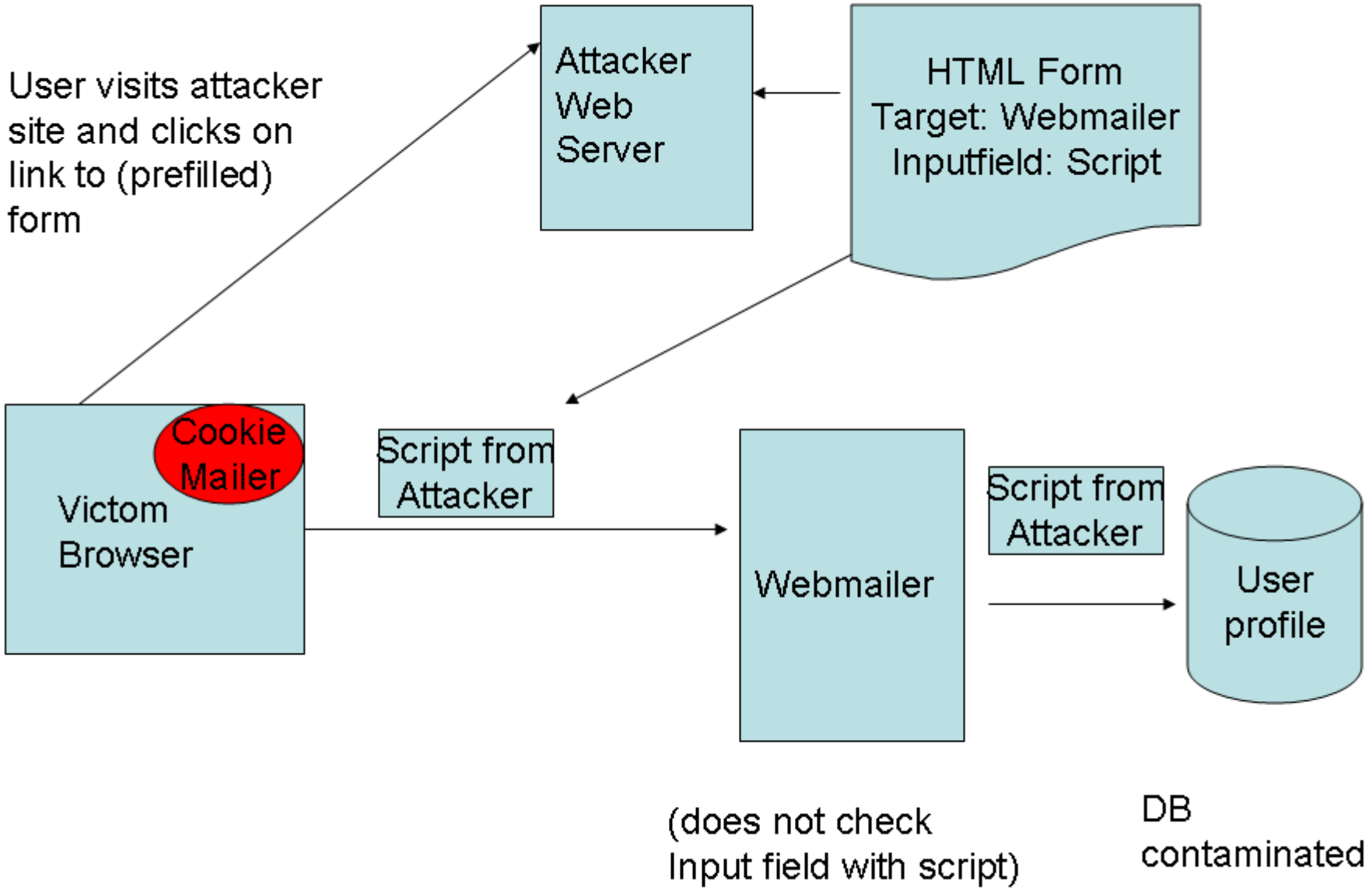

Landesbank Baden-Württemberg - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.lbbw.de/lbbw/ml.nsf/webdokumente/framebooster.htm?OpenDocument&url=
Go
Chapter 8. Security ...
Technical Tip - JAAS ...
Landesbank Baden-Württemberg
Welcome to the kriha.org homepage
Personalisierte Startseite | Anmelder
Google™
Deutschland
?
Web Bilder Groups Verzeichnis News Froogle Mehr »
Erweiterte Suche
Einstellungen
Sprachtools
Google-Suche
Auf gut Glück!
Suche: Das Web Seiten auf Deutsch Seiten aus Deutschland
Werbung - Unternehmensangebote - Über Google - Google.com in English
©2006 Google


Phishing Mail: „Dear Customer of mybank…“
 www.mybank.de
6. Man-in-the-middle modifies transactions on the fly. Modifies Responses too.
1. Trick User into clicking on URL
2. User connects to badguy.de
Badguy.de
Browser/ Mail Reader
TAN
3. Badguy forwards requests to bank and sends responses back to user
8. User sends TAN to badguy
5. User does Transaktions
TAN
SMS/TAN
mybank.de
4. Bank asks user to login.
7. Bank sends Users sms with TAN.

Cross-Site Request Forgery (XSRF or Web-trojan)



Injection Attack



```
#include <stdio.h>

int main(int argc, char** argv)  {

    int foo=0xeeee;
    char myArray[4];
    gets(myArray);
    printf(" print integer first: %x ", foo);
    printf("%s ", myArray);

}
```

Keyboard Input (with return)	Display Output
a	Eeee a
aa	Eeee aa
aaa	Eeee aaa
aaaa	Ee00 aaaa
aaaaaaaaaaaaaa	Core dump with EIP = 6161616161616161 (Hex 61 == `a`)

Our „aaaaaaaa..“ input from keyboard is now the address where the next instruction should be read by the CPU. Now we know how to point the CPU to code we placed on the stack

```
Exception: STATUS_ACCESS_VIOLATION at eip=61616161
eax=00000012 ebx=00000004 ecx=610E3038 edx=00000000 esi=004010AE
edi=610E21A0
ebp=61616161 esp=0022EF08
program=D:\kriha\security\bufferoverflow\over.exe, pid 720, thread main
cs=001B ds=0023 es=0023 fs=003B gs=0000 ss=0023
Stack trace:
Frame     Function  Args
  90087 [main] over 720 handle_exceptions: Exception:
STATUS_ACCESS_VIOLATION
 104452 [main] over 720 handle_exceptions: Error while dumping state
(probably corrupted stack)
```

A program crash is a way into the system!

The kernal trap interface

your code wants to send a message msg to stdout:

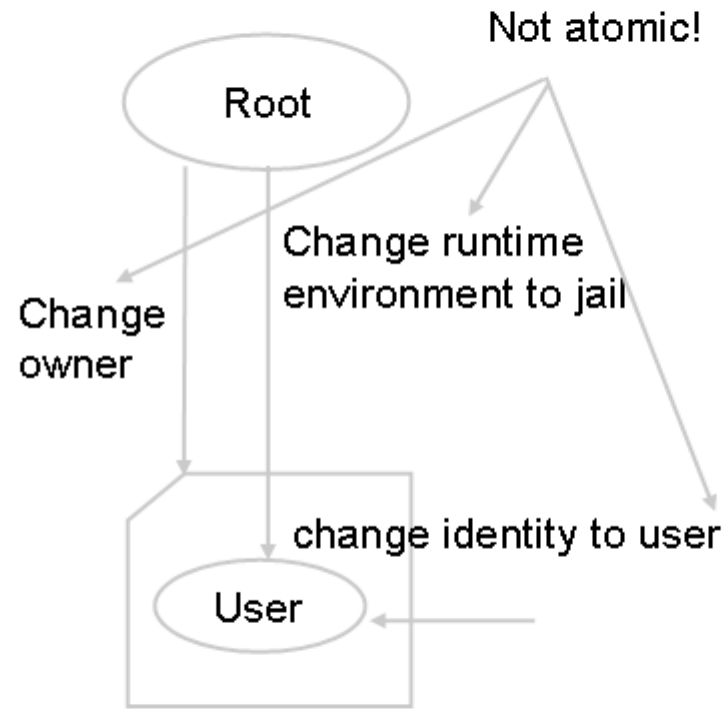
```
push  len      ;message length
 push   msg      ;message to write
 push   1        ;file descriptor (stdout)
  mov     AX, 0x4        ;system call number (sys_write)
 int 0x80         ; kernel interrupt (trap)
add     SP, 12         ;clean stack (3 arguments * 4)
push    0        ;exit code
mov     AX, 0x1        ;system call number (sys_exit)
 int 0x80       ; kernel interrupt we do not return from sys_exit there's no need to clean stack
```

The trap (system call interface) ist very important for attack code because it is POSITION INDEPENDENT! Your code is NOT LINKED with the running program and therefore does not know where specific library functions etc. are located in your program. The kernel interface is always just there and can be used to load Dynamic Link Libraries into the program.

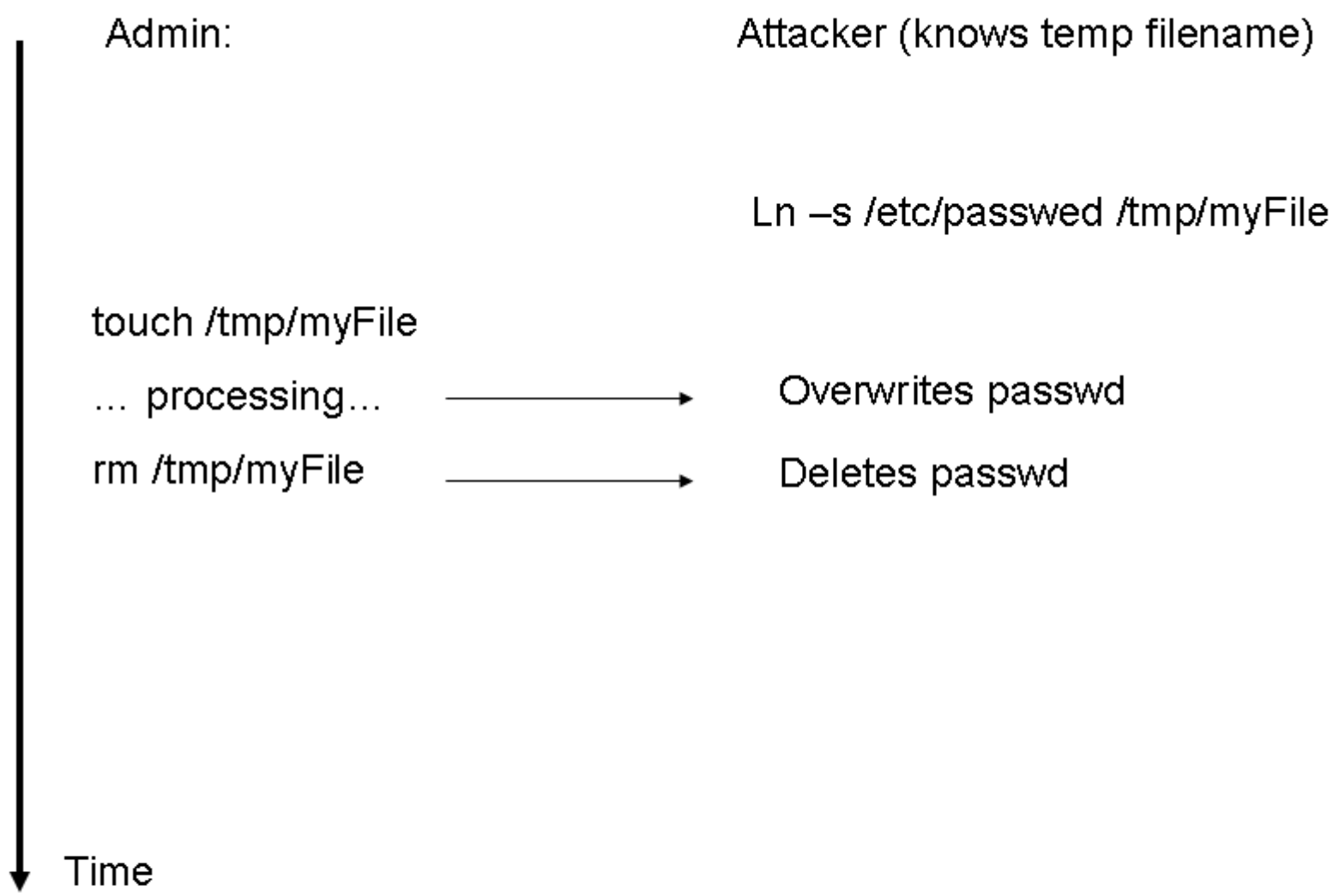
Attack Vectors on Web Services:

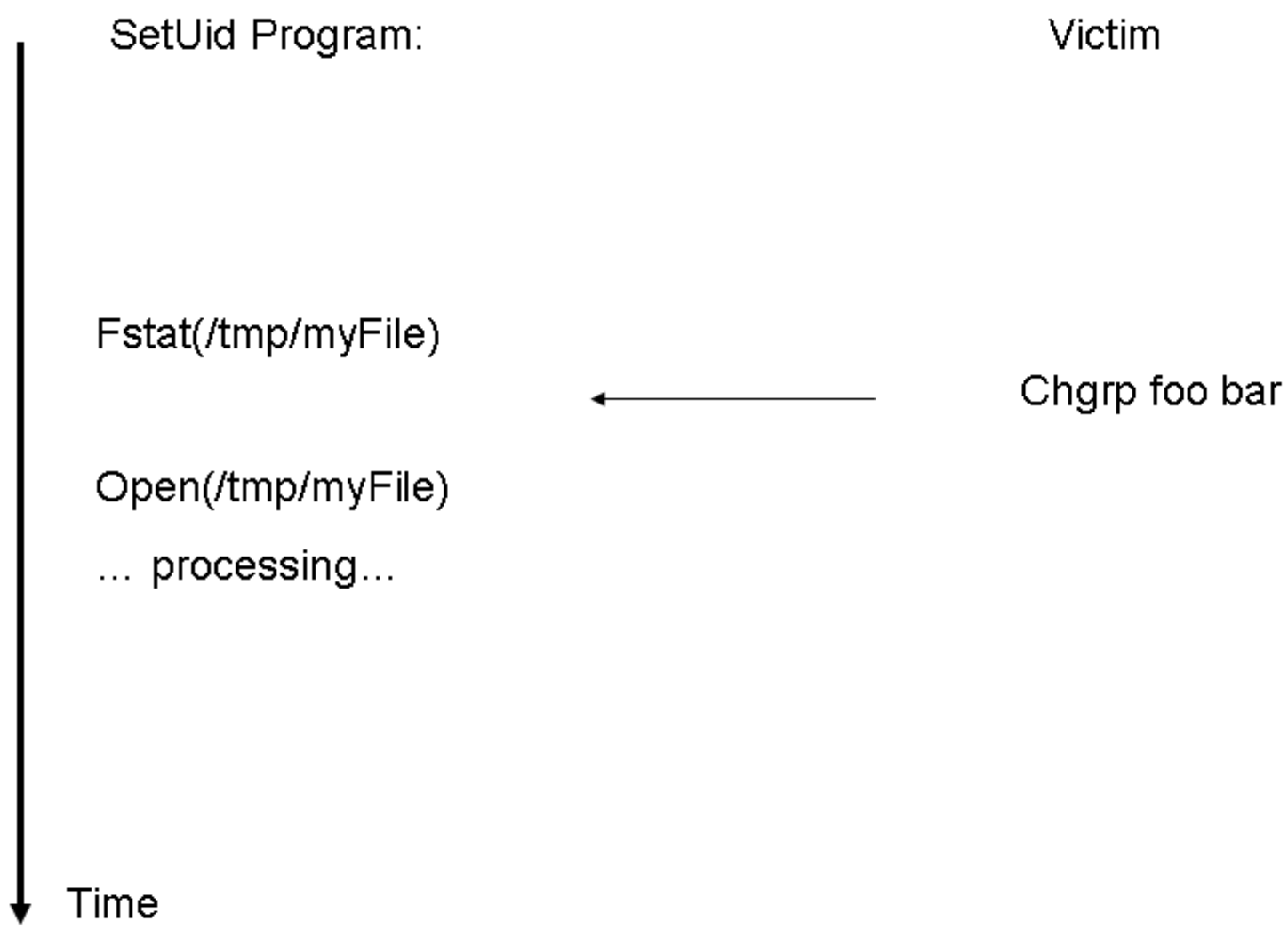
- Wrong input length of variables
- Variables containing wrong characters or meta-characters
- Variables containing SQL commands
- Responses which expose SOAP error codes

Administration and Race Conditions: toc2tou bugs

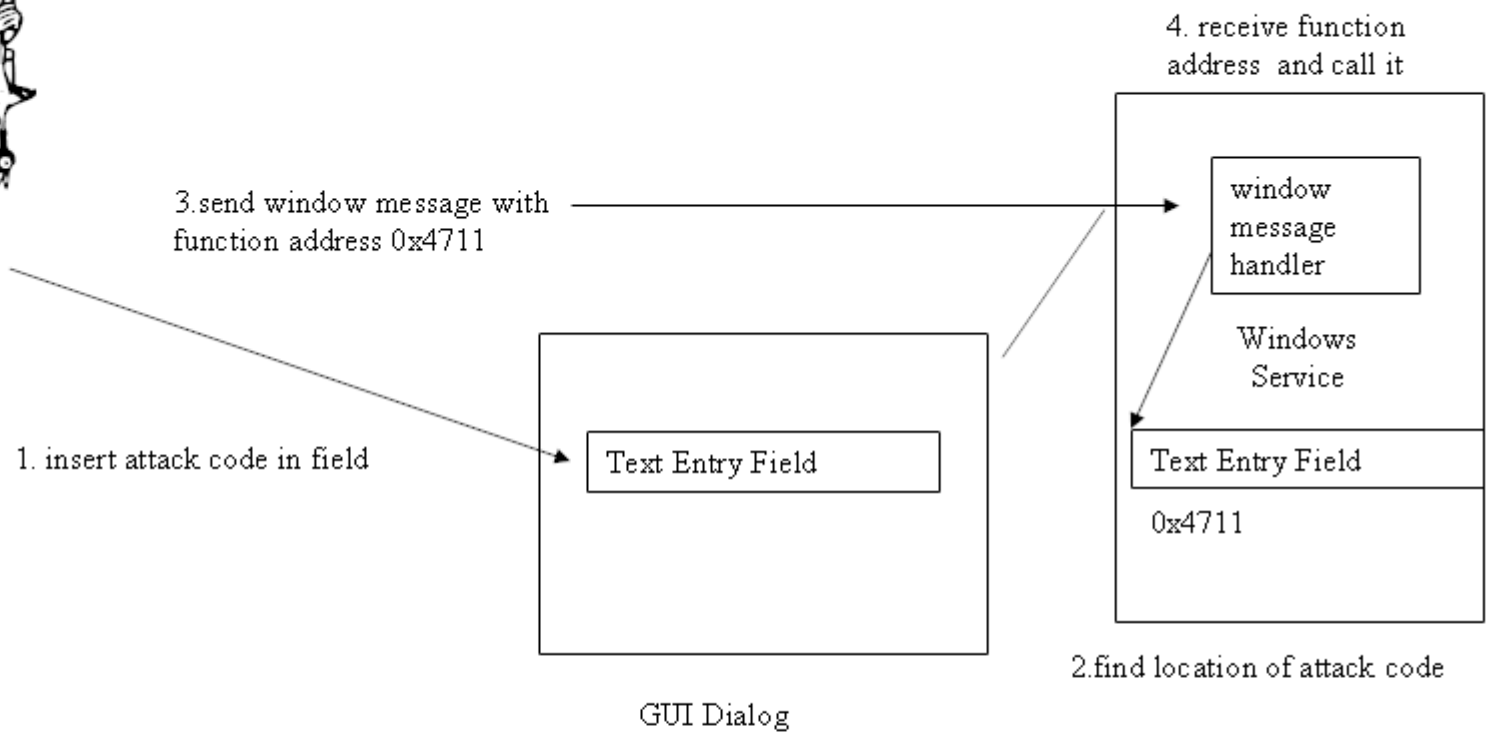


Jails strips off other rights

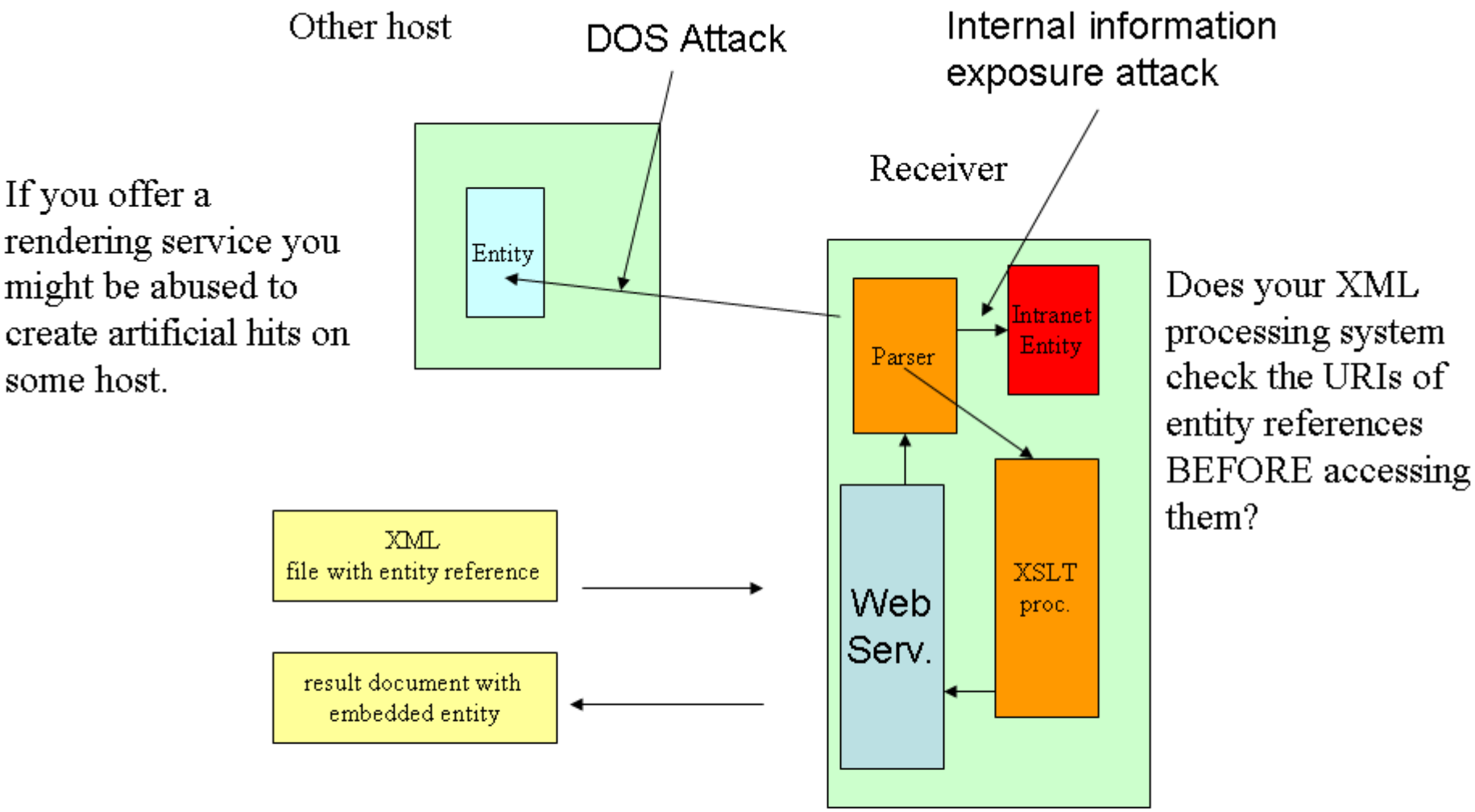

Admin:
Attacker (knows temp filename)
Ln –s /etc/passwed /tmp/myFile
touch /tmp/myFile
… processing…
Overwrites passwd
rm /tmp/myFile
Deletes passwd
Time


SetUid Program:
Victim
Fstat(/tmp/myFile)
Chgrp foo bar
Open(/tmp/myFile)
… processing…
Time

Shatter Attack: fundamental software design flaws

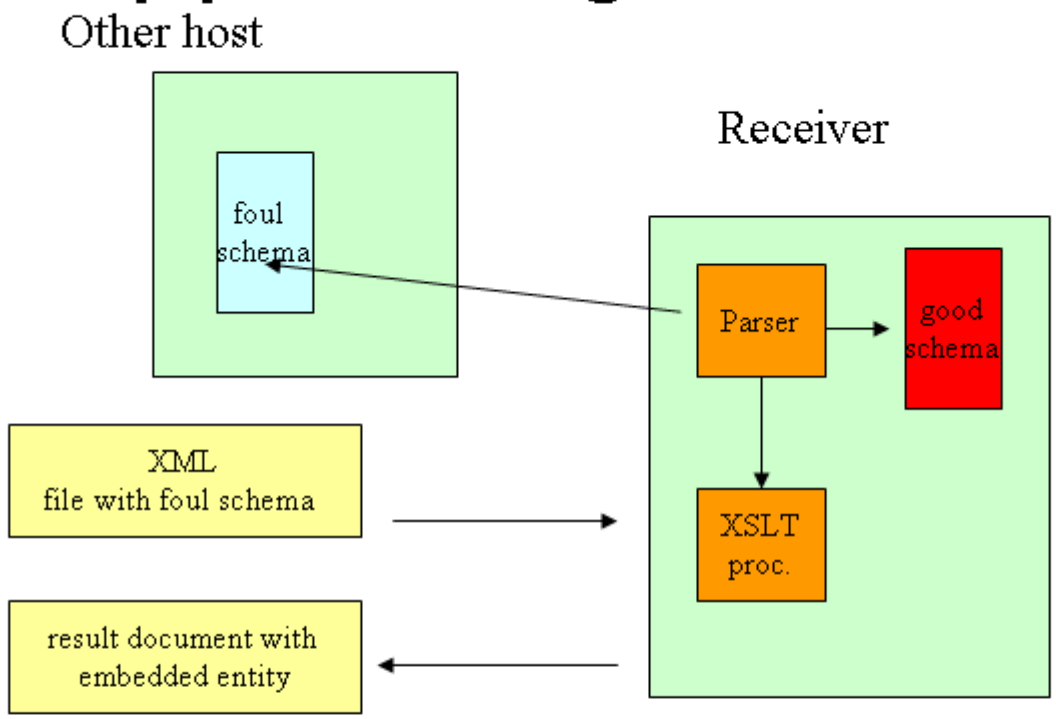


Here the danger is that any program can send certain window messages which contain function addresses IN THE RECEIVERS ADDRESS SPACE. By placing some attack code into the receiver (not hard if a GUI is used by the receiver) the attacker can then direct the receiver message handler to direct control flow to the attack code (step 4 above).

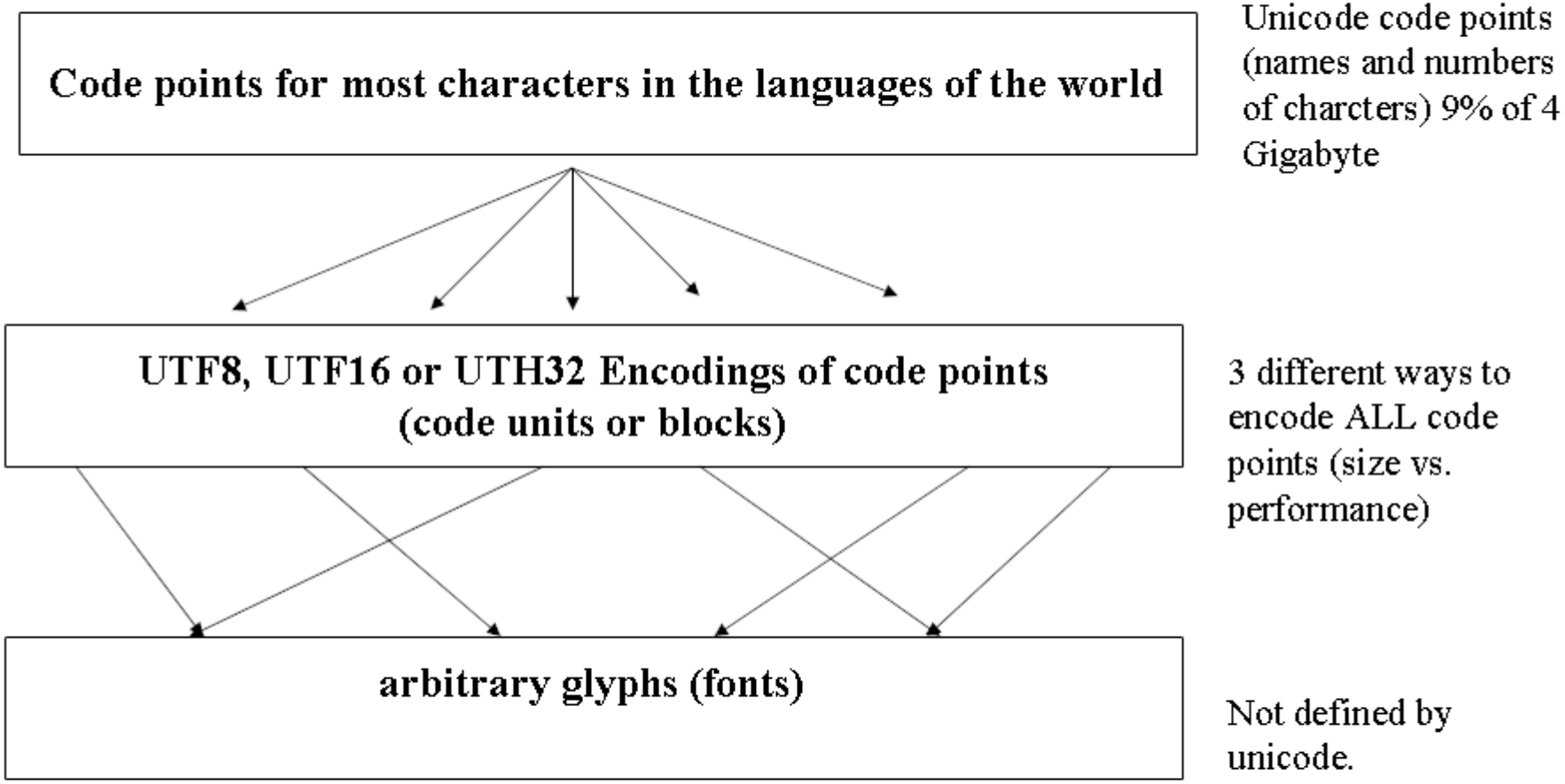

Other host
DOS Attack
Internal information exposure attack
If you offer a rendering service you might be abused to create artificial hits on some host.
Entity
Receiver
Parser
Intranet Entity
XSLT proc.
Web Serv.
XML file with entity reference
result document with embedded entity
Does your XML processing system check the URIs of entity references BEFORE accessing them?

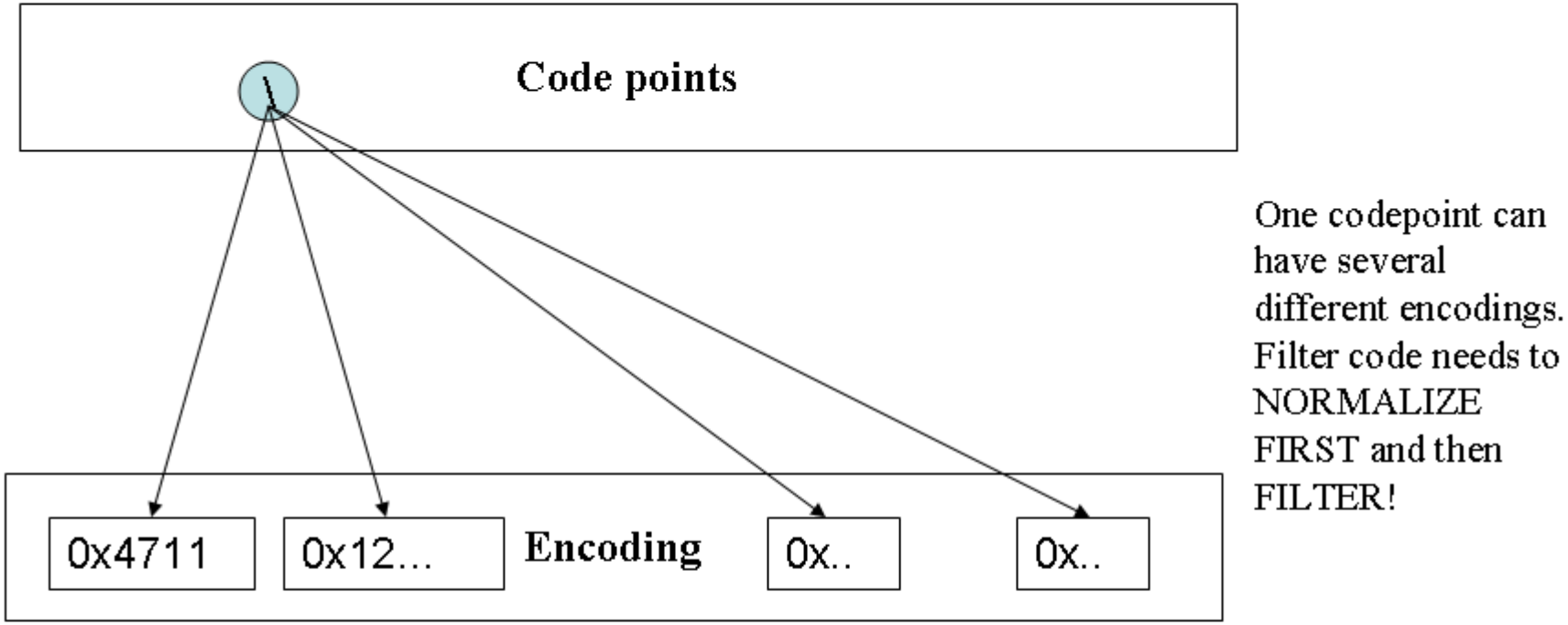
```
<?xml version='1.0'?>

<xsl:stylesheet
xmlns:xsl=http://www.w3.org/1999/XSL/Transform version='1.0'>

<xsl:output method="html„ encoding="ISO-8859-1„ indent="no"/>

<!-- ================================================ -->

<xsl:script language=„java“ implements-prefix=„sy“
src=„java:java.util.system“/>

<xsl:template match="*">

  <xsl:message>

    <xsl:text>No template matches </xsl:text>

    <xsl:value-of select=„sy:exec(…)"/>

    <xsl:text>.</xsl:text>

  </xsl:message>
```

Suppressing Validation



James Clark mentioned recently an especially evil way to work around validation: „Suppose an application is trying to use validation to protect itself from bad input. It carefully loads the schema cache with the namespaces it knows about, and calls validate(). Now the bad guy comes along and uses a root element from some other namespace and uses xsi:schemaLocation to point to his own schema that that has a declaration for that element and uses <xs:any namespace="##any„ processContents="skip"/>. Won't they just have almost completely undermined any protection that was supposed to come from validation?“


Code points for most characters in the languages of the world
Unicode code points (names and numbers of charcters) 9% of 4 Gigabyte
UTF8, UTF16 or UTH32 Encodings of code points (code units or blocks)
3 different ways to encode ALL code points (size vs. performance)
arbitrary glyphs (fonts)
Not defined by unicode.



One codepoint can have several different encodings. Filter code needs to NORMALIZE FIRST and then FILTER!

Filter code to detect ..\..\ attacks:

```
If (encoded == 0x4711)
        removeCharacter();
// what about the other possible
encodings of backslash????
```

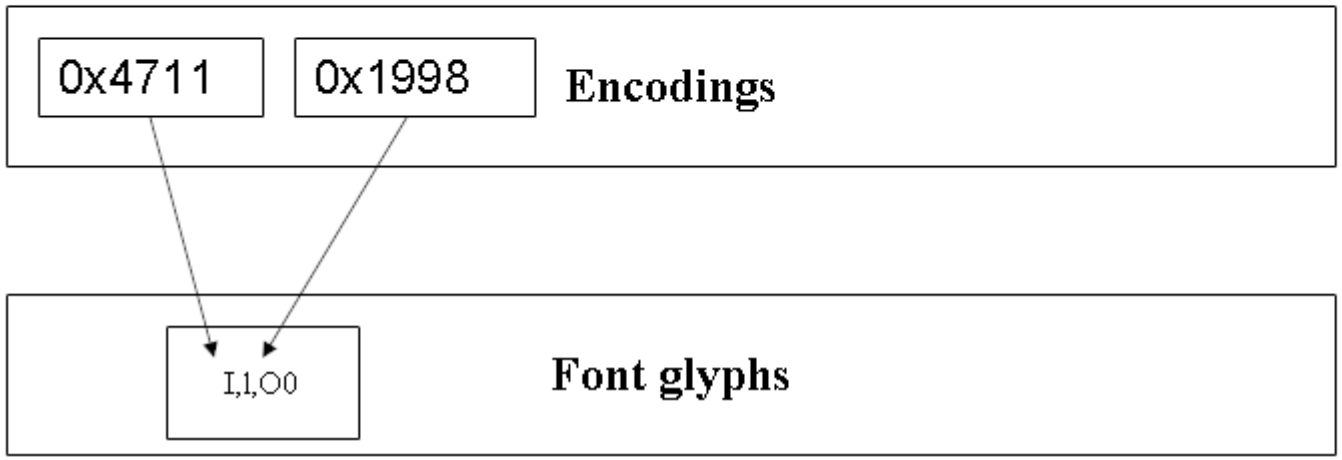
Unicode Exploit

code point U+0000

Unicode code points (names and numbers of charcters) 9% of 4 Gigabyte

encoded as: 0, 110 00000 10 000000, etc.

Processors are not allowed to interpret any encoding other than the shortest form, in this case 0. Otherwise the extended forms could escape filtering and become active during interpretation.



Fonts can display unicode code points any way they want.

One visual „look“ (e.g. lowercase „l“ and uppercase „I“ or greek omicron vs latin o.

Unicode homographs and DNS

Two different code points

ASCII DNS

Unicode Characters DNS

DNS names can now contain Unicode characters

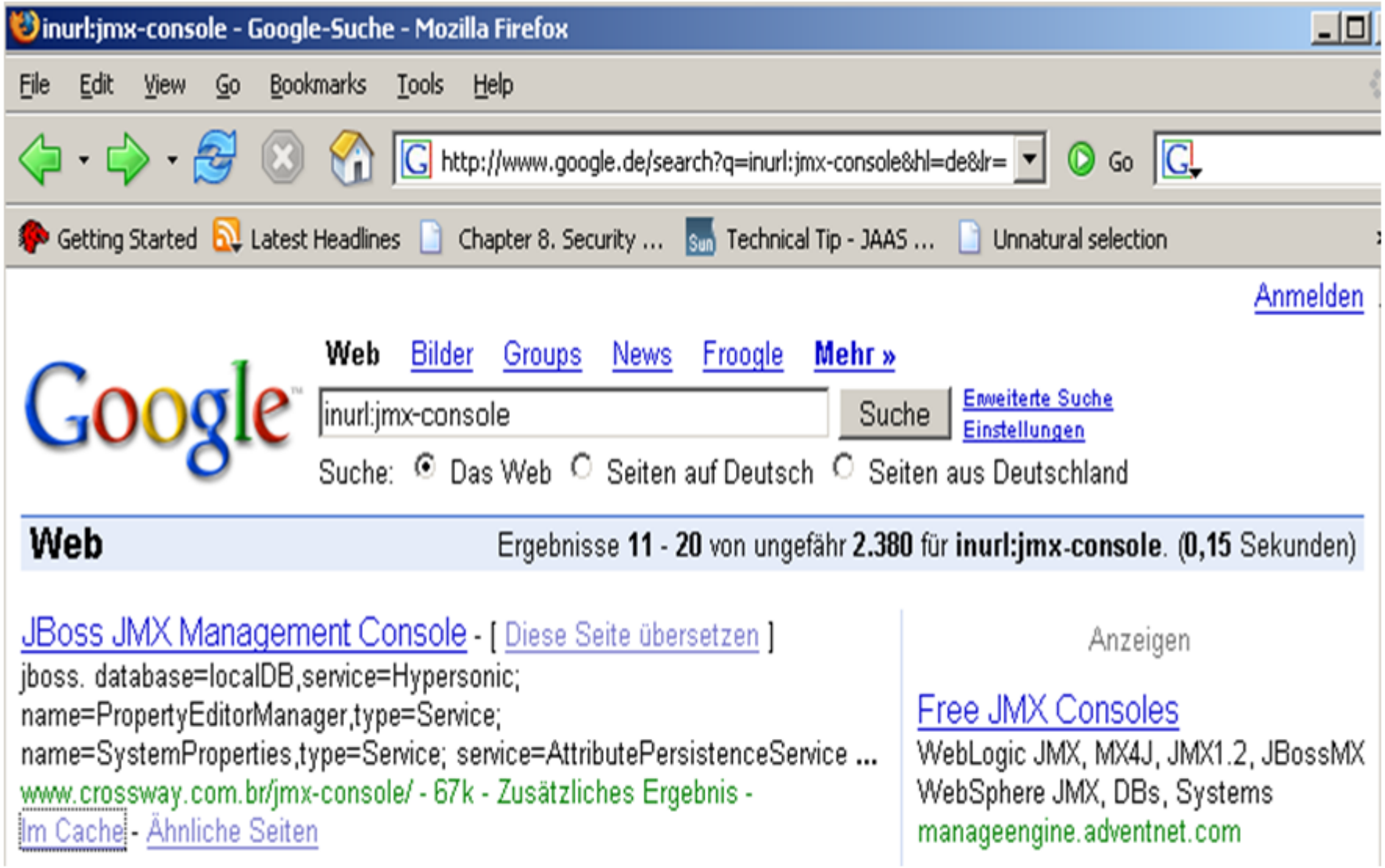
two different fonts

Not defined by unicode.

I,l,O

One visual „look“ (e.g. lowercase „l“ and uppercase „I“ or greek omicron vs latin o.

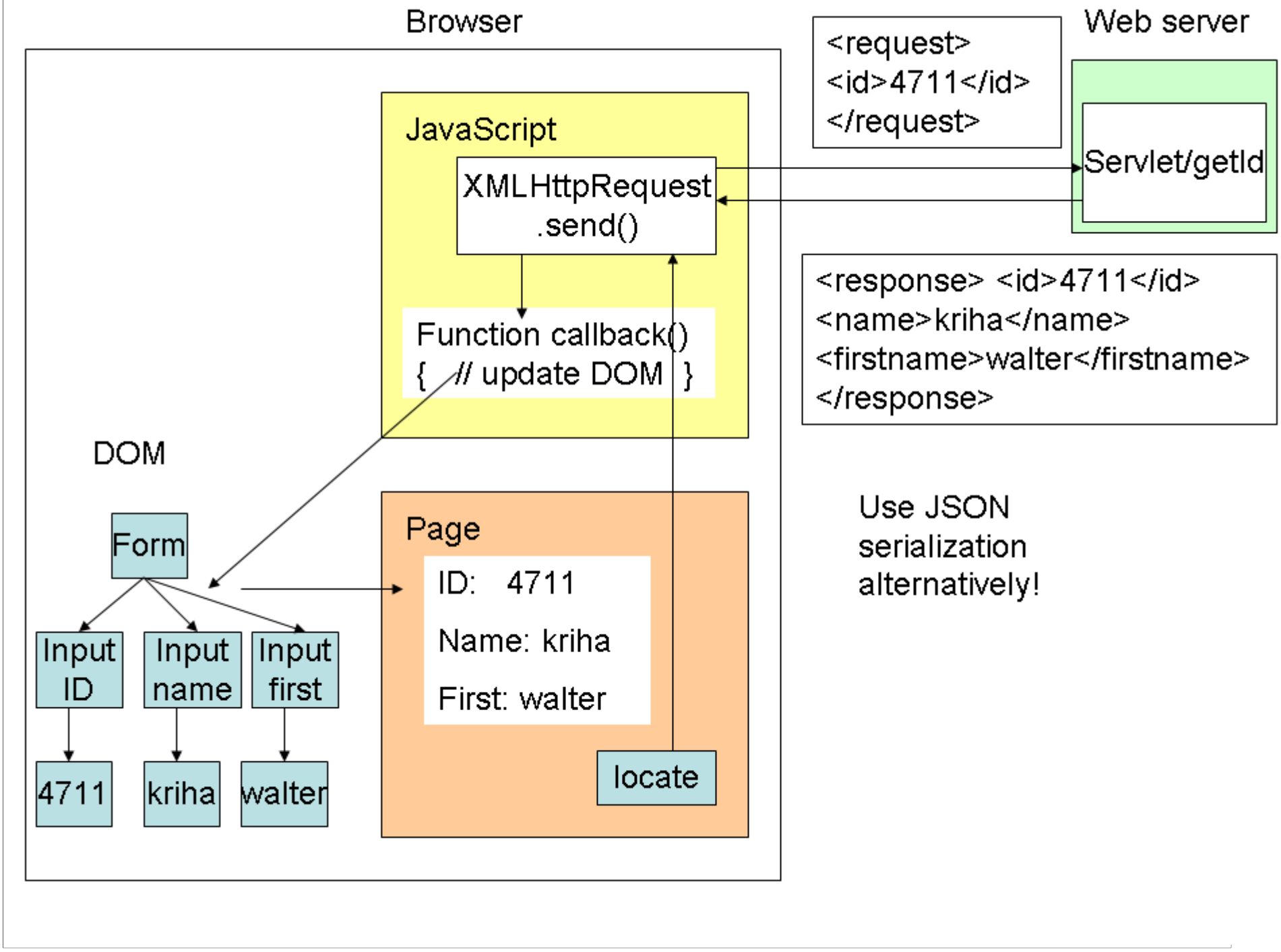
The firefox browser switched back to showing the unicode escape sequences in domain names to allow the user to differenciate e.g. a latin „a“ from a kyrillic „a“. Otherwise the user could be tricked into connecting to www.ebay.com with the „a“ being really the cyrillic version. In this case the user would connect to the wrong site. Expect many more security problems with unicode in the future, especially in the GUI area.

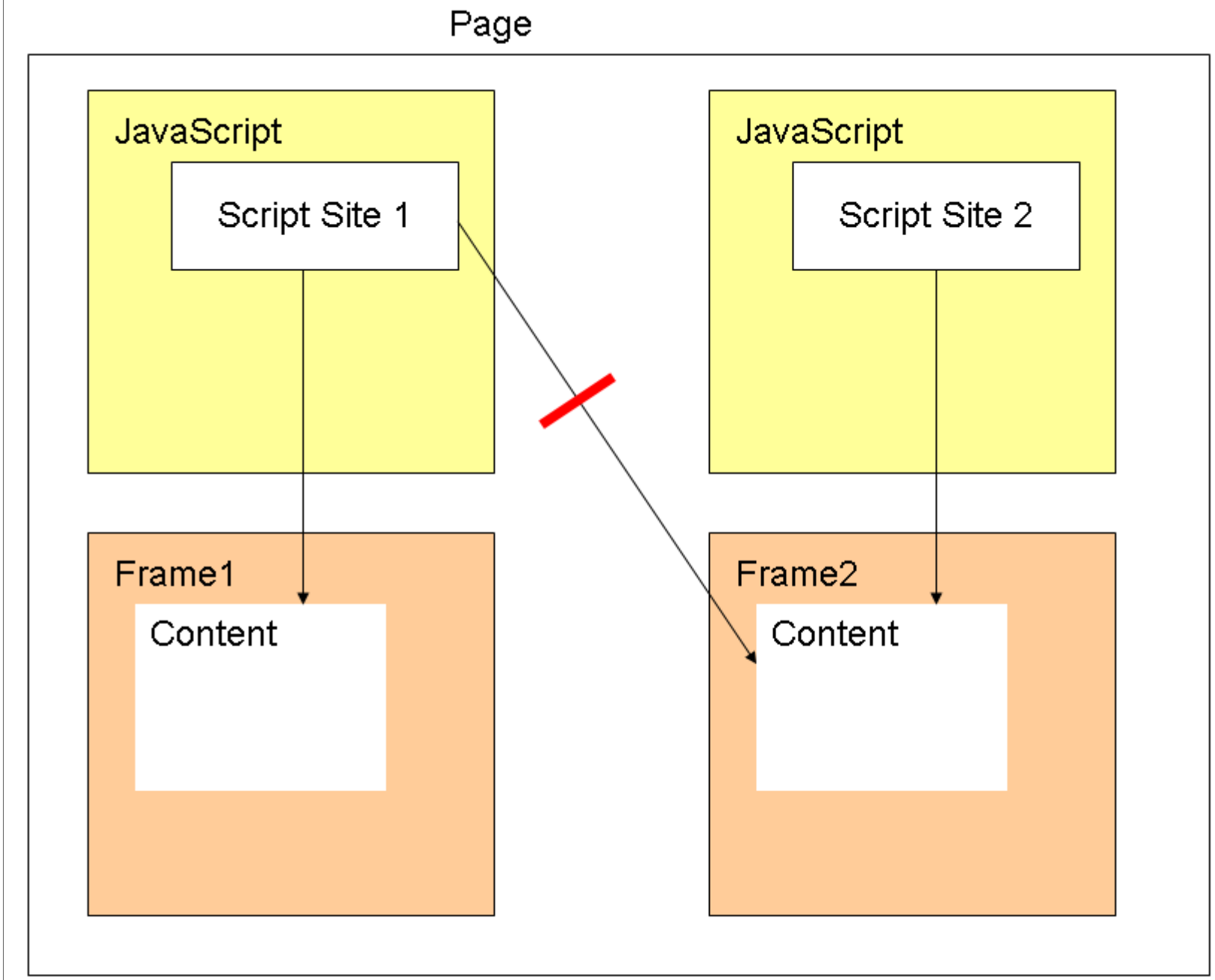

inurl:jmx-console - Google-Suche - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.google.de/search?q=inurl:jmx-console&hl=de&lr=
Go
Getting Started
Latest Headlines
Chapter 8. Security ...
Technical Tip - JAAS ...
Unnatural selection
Anmelden
Web Bilder Groups News Froogle Mehr »
Google
inurl:jmx-console
Suche
Erweiterte Suche
Einstellungen
Suche: Das Web Seiten auf Deutsch Seiten aus Deutschland
Web
Ergebnisse 11 - 20 von ungefähr 2.380 für inurl:jmx-console. (0,15 Sekunden)
JBoss JMX Management Console - [Diese Seite übersetzen]
jboss. database=localDB,service=Hypersonic;
name=PropertyEditorManager,type=Service;
name=SystemProperties,type=Service; service=AttributePersistenceService ...
www.crossway.com.br/jmx-console/ - 67k - Zusätzliches Ergebnis -
Im Cache - Ähnliche Seiten
Anzeigen
Free JMX Consoles
WebLogic JMX, MX4J, JMX1.2, JBossMX
WebSphere JMX, DBs, Systems
manageengine.adventnet.com

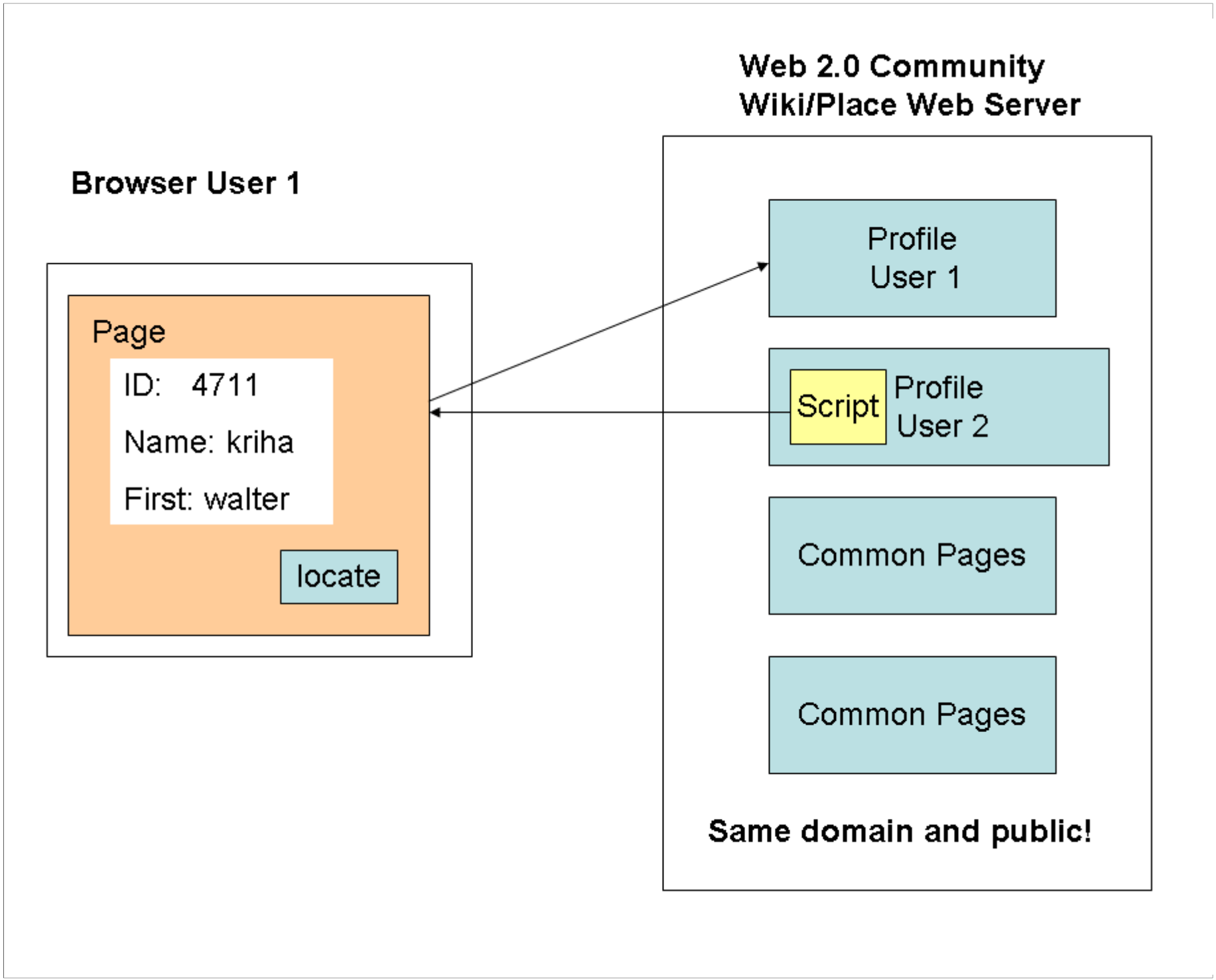
AMAZON E-Commerce Service

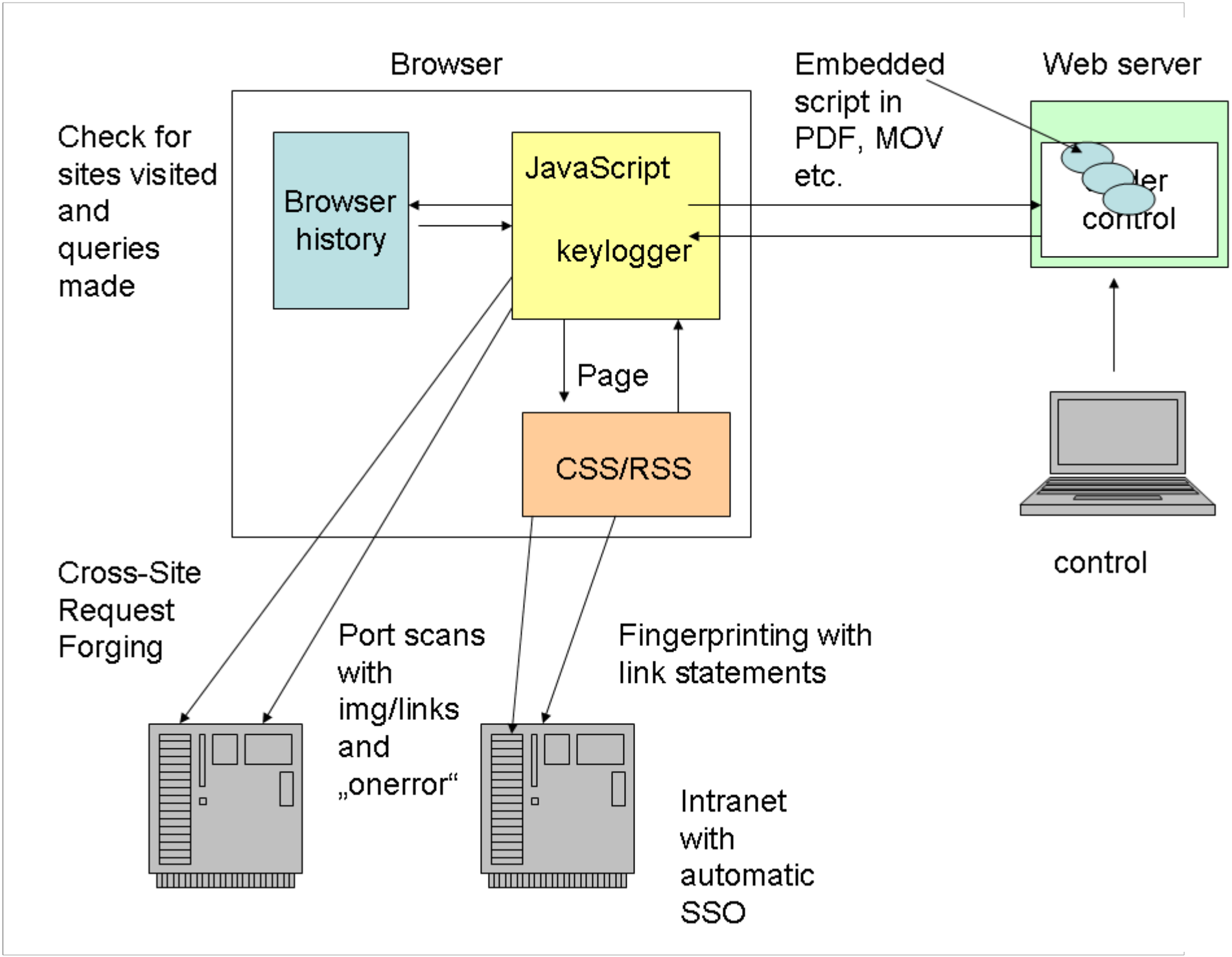
Sample REST Request with Style Parameter

http://webservices.amazon.com/onca/xml?Service=AWSECommerceService &
AWSAccessKeyId=[Your Access Key ID Here] &
Operation=ItemLookup &IdType=ASIN &
ItemId=B00008OE6I &ResponseGroup=Large &
Style=http://www.yourdomain.com/your-xsl-style-sheet.xsl


Browser
Web server
<request>
<id>4711</id>
</request>
JavaScript
XMLHttpRequest
.send()
Servlet/getId
<response> <id>4711</id>
<name>kriha</name>
<firstname>walter</firstname>
</response>
Function callback()
{ // update DOM }
DOM
Form
Page
ID: 4711
Name: kriha
First: walter
Use JSON
serialization
alternatively!
Input
ID
Input
name
Input
first
4711
kriha
walter
locate


Page
JavaScript
Script Site 1
JavaScript
Script Site 2
Frame1
Content
Frame2
Content


Web 2.0 Community
Wiki/Place Web Server
Browser User 1
Page
ID: 4711
Name: kriha
First: walter
locate
Profile
User 1
Script
Profile
User 2
Common Pages
Common Pages
Same domain and public!


Browser
Embedded script in PDF, MOV etc.
Web server
Check for sites visited and queries made
Browser history
JavaScript
keylogger
control
Page
CSS/RSS
control
Cross-Site Request Forging
Port scans with img/links and „onerror“
Fingerprinting with link statements
Intranet with automatic SSO

Known Mitigation Examples

WAF Filtering, Network level filtering

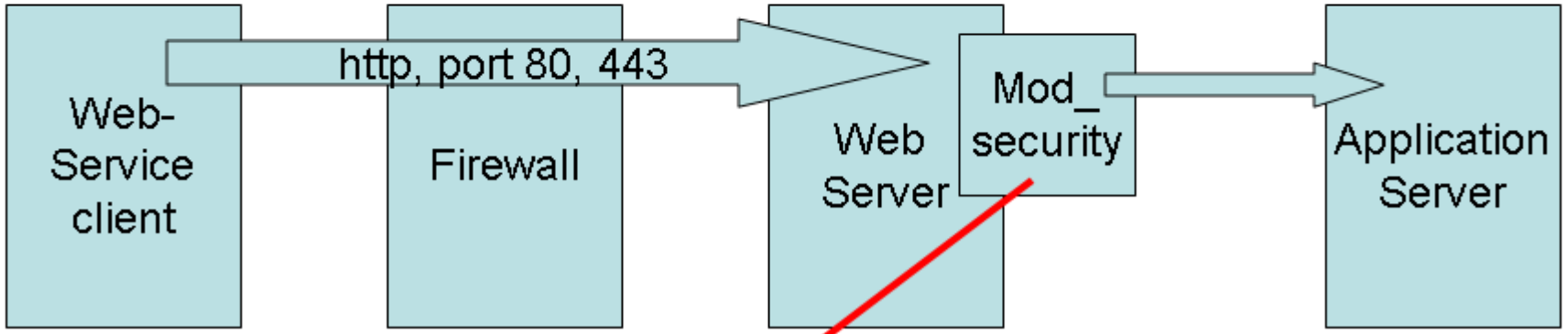
SecFilterSelective Number "!^(|[0-9]{1,9})$"

Check Number for:

- Length
- Characters/Meta
- SQL commands

Check request for

Soap faultcode (avoid exposure of error information)



POST /InStock HTTP/1.1 Host: www.example.org Content-Type: application/soap+xml; charset=utf-8 Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

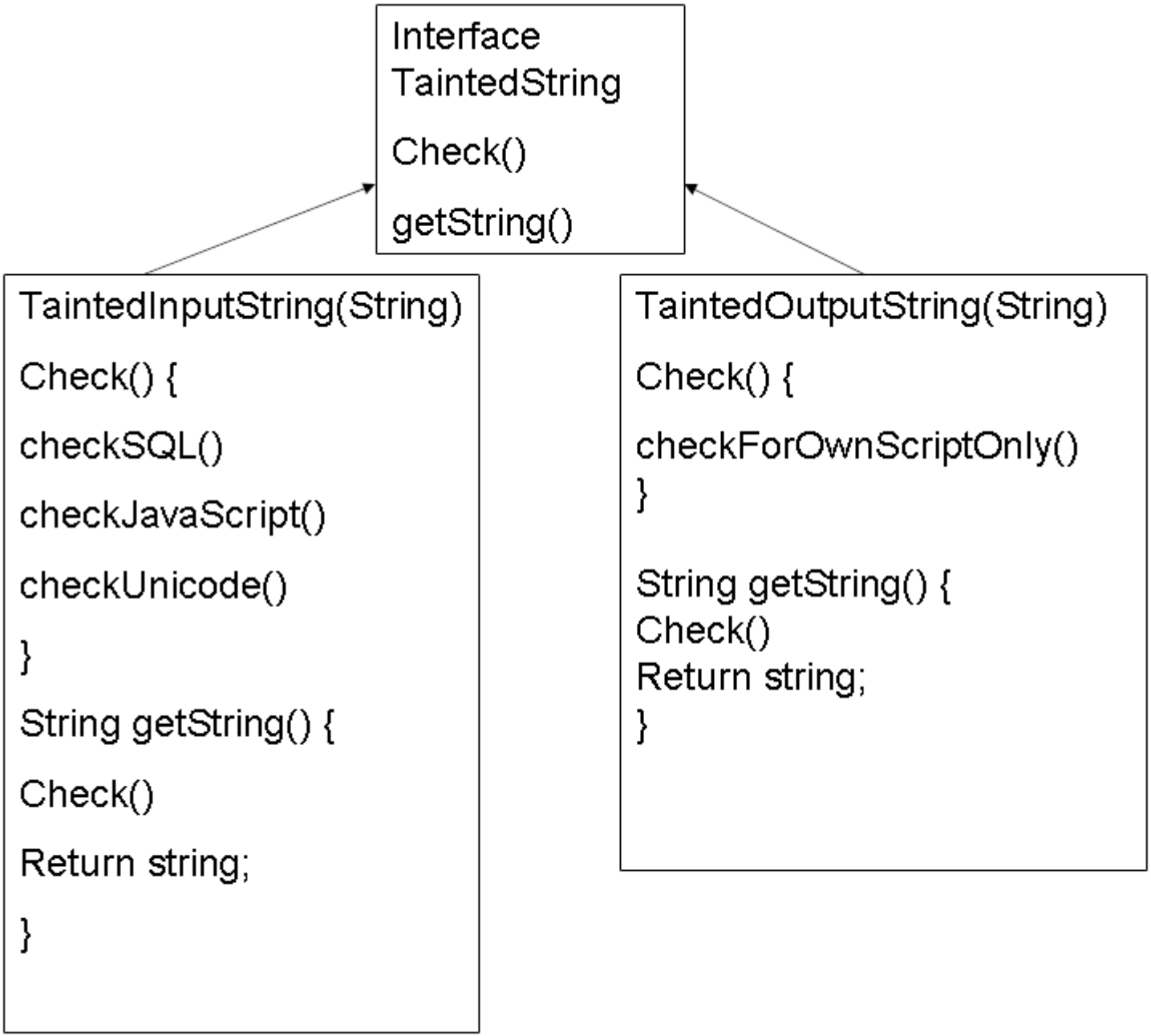
<soap:Body xmlns:k="http://www.kriha.org/number"> <m:GetId> <m:Number>4711</m:Number> </m:GetId>

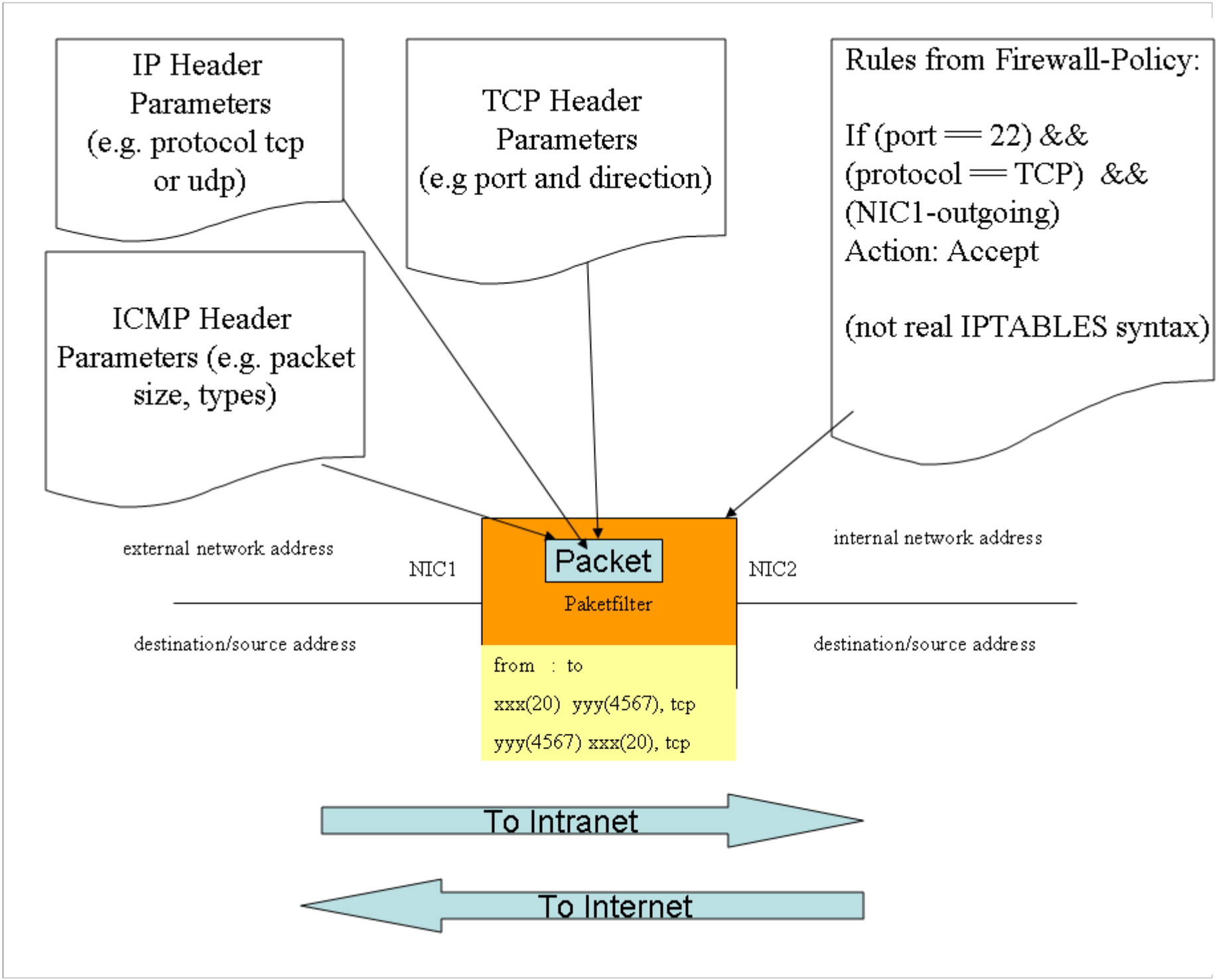
</soap:Body>

</soap:Envelope>

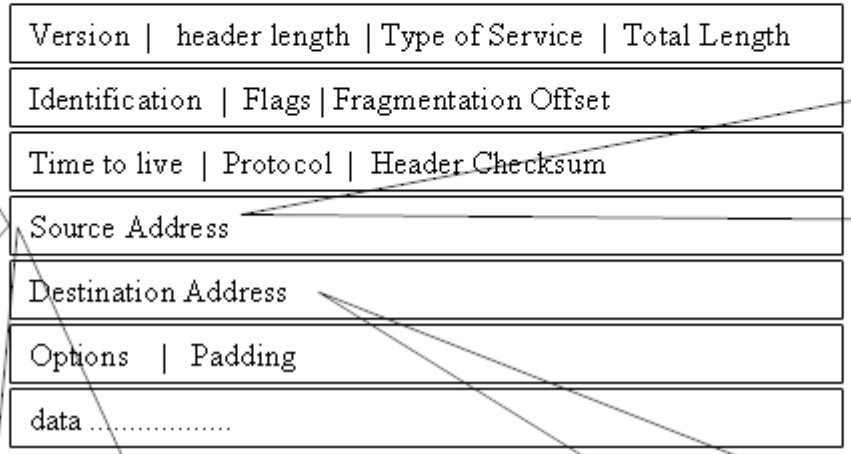
Other security related features of mod_security:

- URL checking
- Unicode normalization
- Message canonicalization for filtering
- Stateful filtering of selected requests
- Stateful connection of input/output values
- Stateful link/request control (did the link come from the server?)


Interface
TaintedString
Check()
getString()
TaintedInputString(String)
Check() {
checkSQL()
checkJavaScript()
checkUnicode()
}
String getString() {
Check()
Return string;
}
TaintedOutputString(String)
Check() {
checkForOwnScriptOnly()
}
String getString() {
Check()
Return string;
}


IP Header Parameters (e.g. protocol tcp or udp)
TCP Header Parameters (e.g port and direction)
Rules from Firewall-Policy:
If (port == 22) && (protocol == TCP) && (NIC1-outgoing)
Action: Accept
(not real IPTABLES syntax)
ICMP Header Parameters (e.g. packet size, types)
external network address
NIC1
Packet
Paketfilter
NIC2
internal network address
destination/source address
destination/source address
from : to
xxx(20) yyy(4567), tcp
yyy(4567) xxx(20), tcp
To Intranet
To Internet

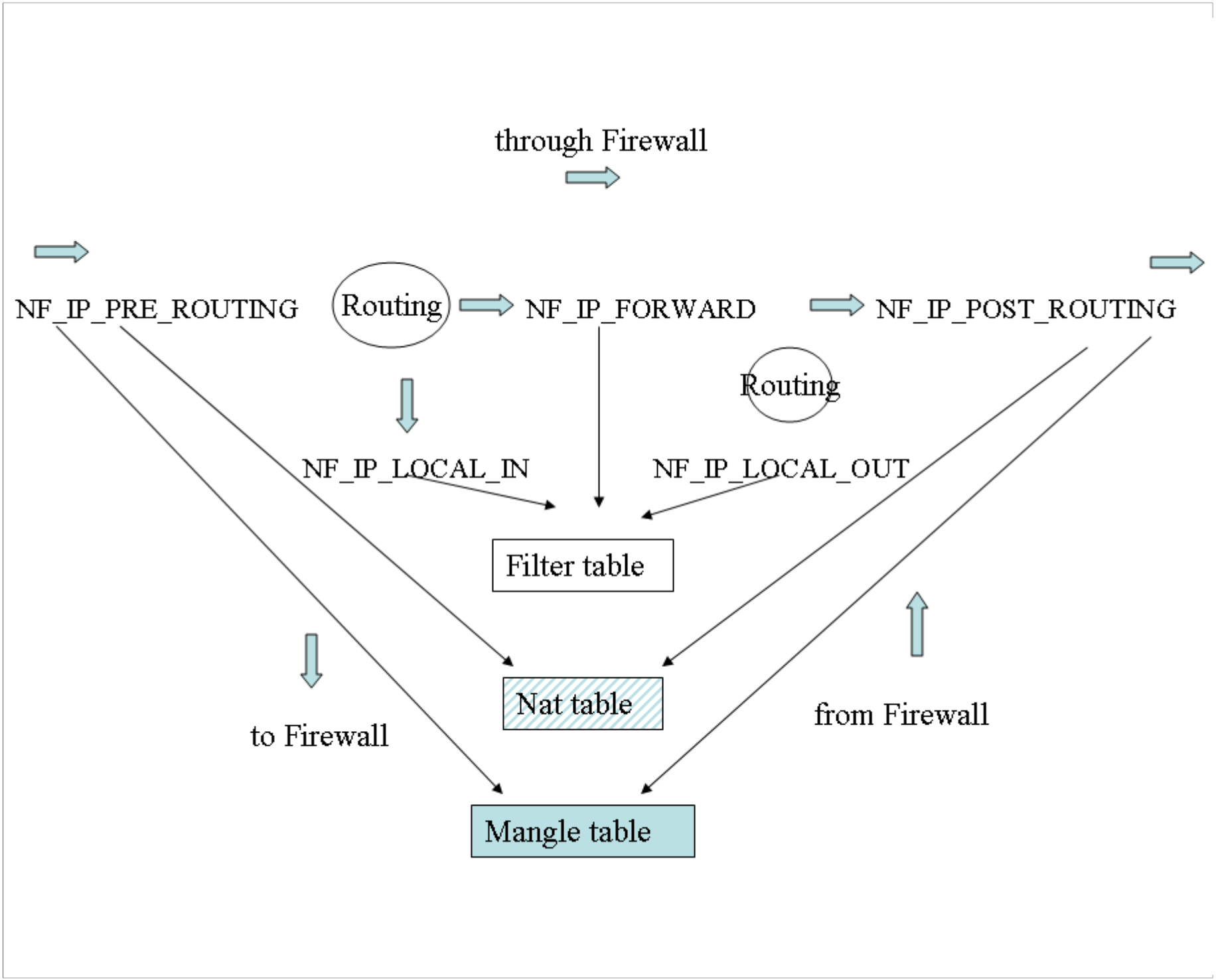
Network Address Translation (NAT) means that the source or destination address of a packet is changed

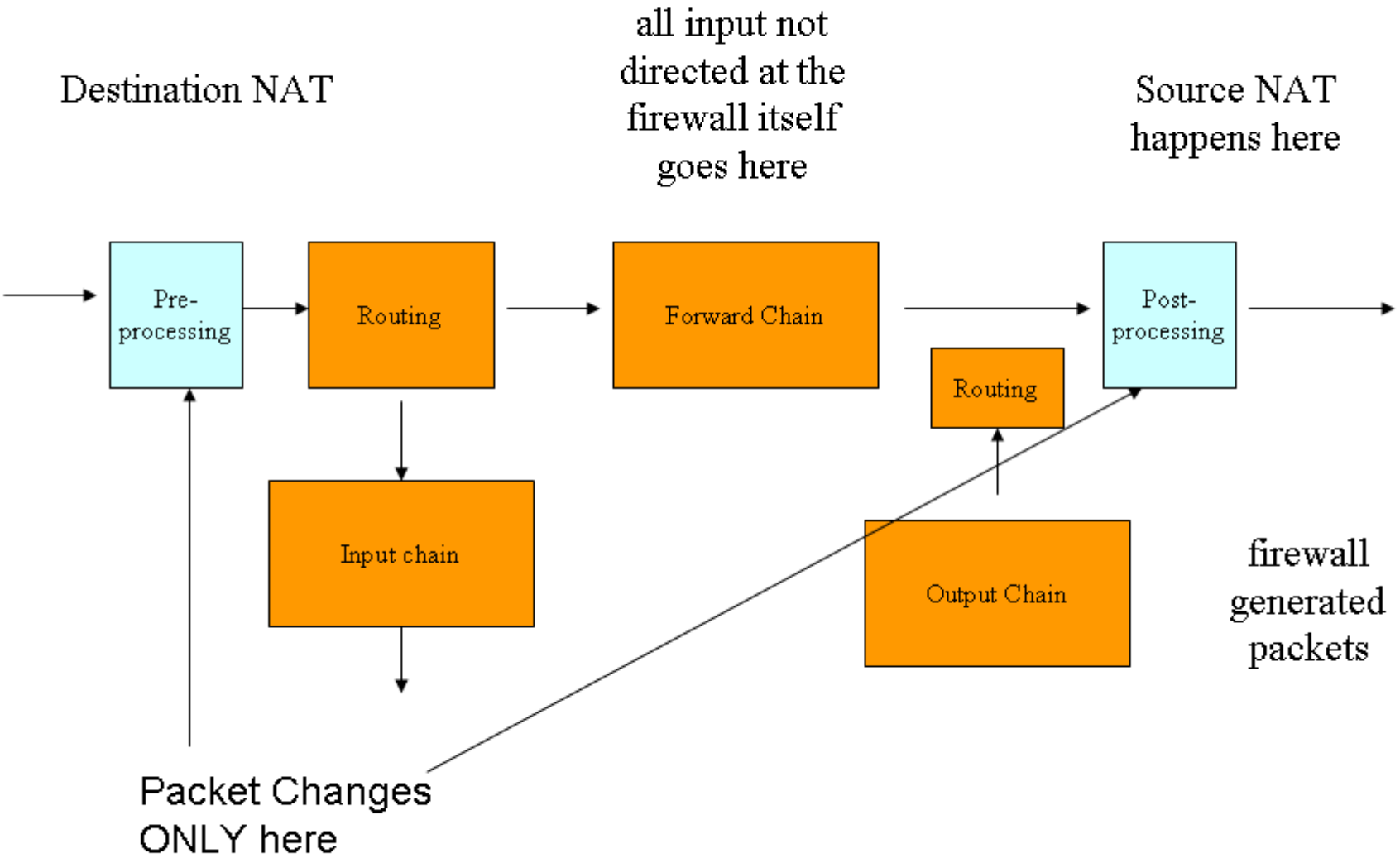


masquerading is almost like SNAT only that there is no static IP address. Instead, the source address is dynamically grabbed from an ISP, e.g via DHCP, pppoe etc.

With Source NAT (SNAT), the source address is changed, e.g. to map from private IP addresses to the real IP address of a firewall, thereby hiding the internal network.

With Destination NAT (DNAT) the target address is changed, e.g. to allow transparent proxying or load-balancing


through Firewall
NF_IP_PRE_ROUTING
Routing
NF_IP_FORWARD
NF_IP_POST_ROUTING
Routing
NF_IP_LOCAL_IN
NF_IP_LOCAL_OUT
Filter table
Nat table
to Firewall
from Firewall
Mangle table


Destination NAT
all input not directed at the firewall itself goes here
Source NAT happens here
Pre-processing
Routing
Forward Chain
Routing
Post-processing
Input chain
Output Chain
firewall generated packets
Packet Changes ONLY here

iptables -t table -command [chain] [match] –j [target/jump]

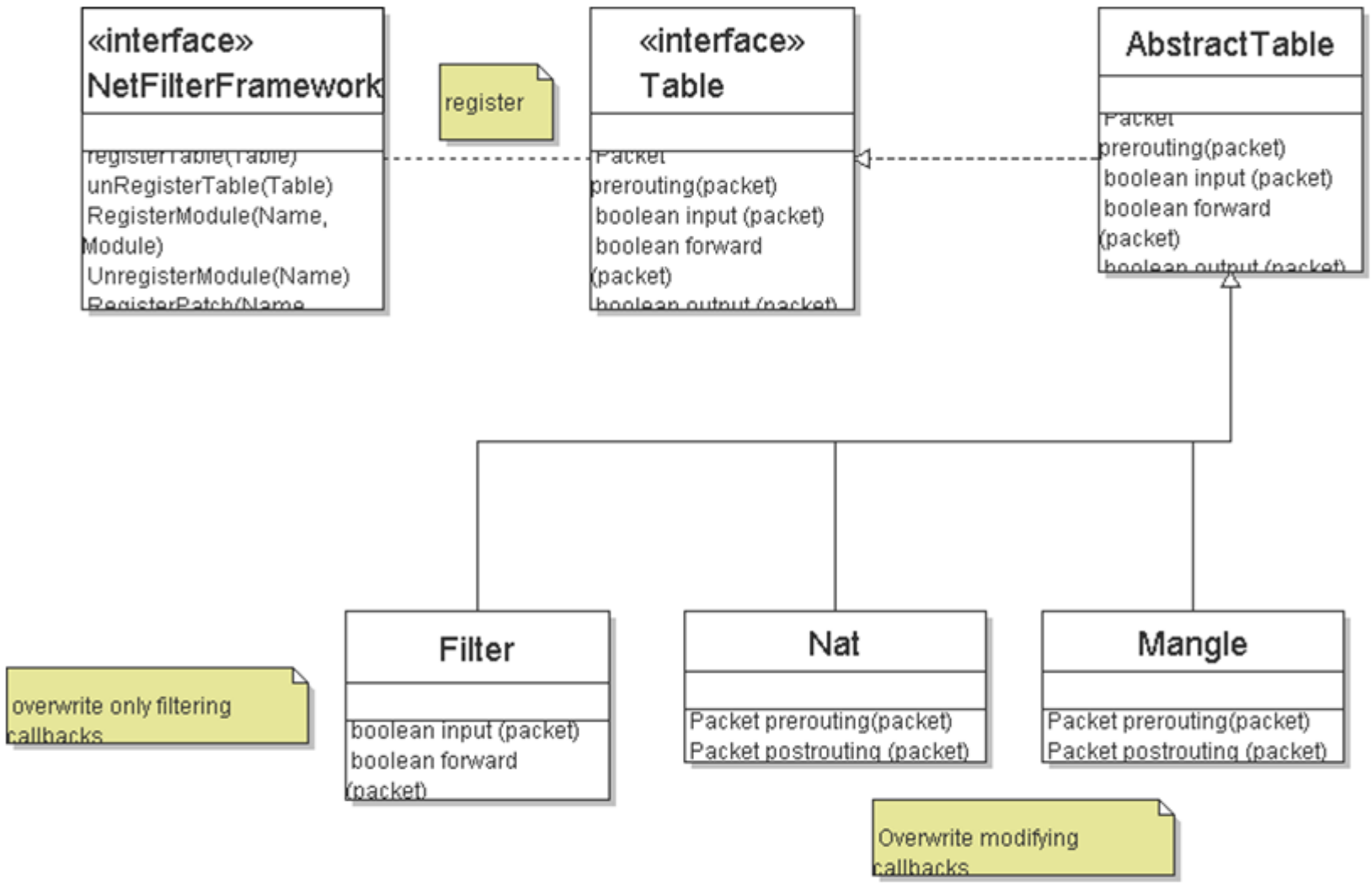
Example:

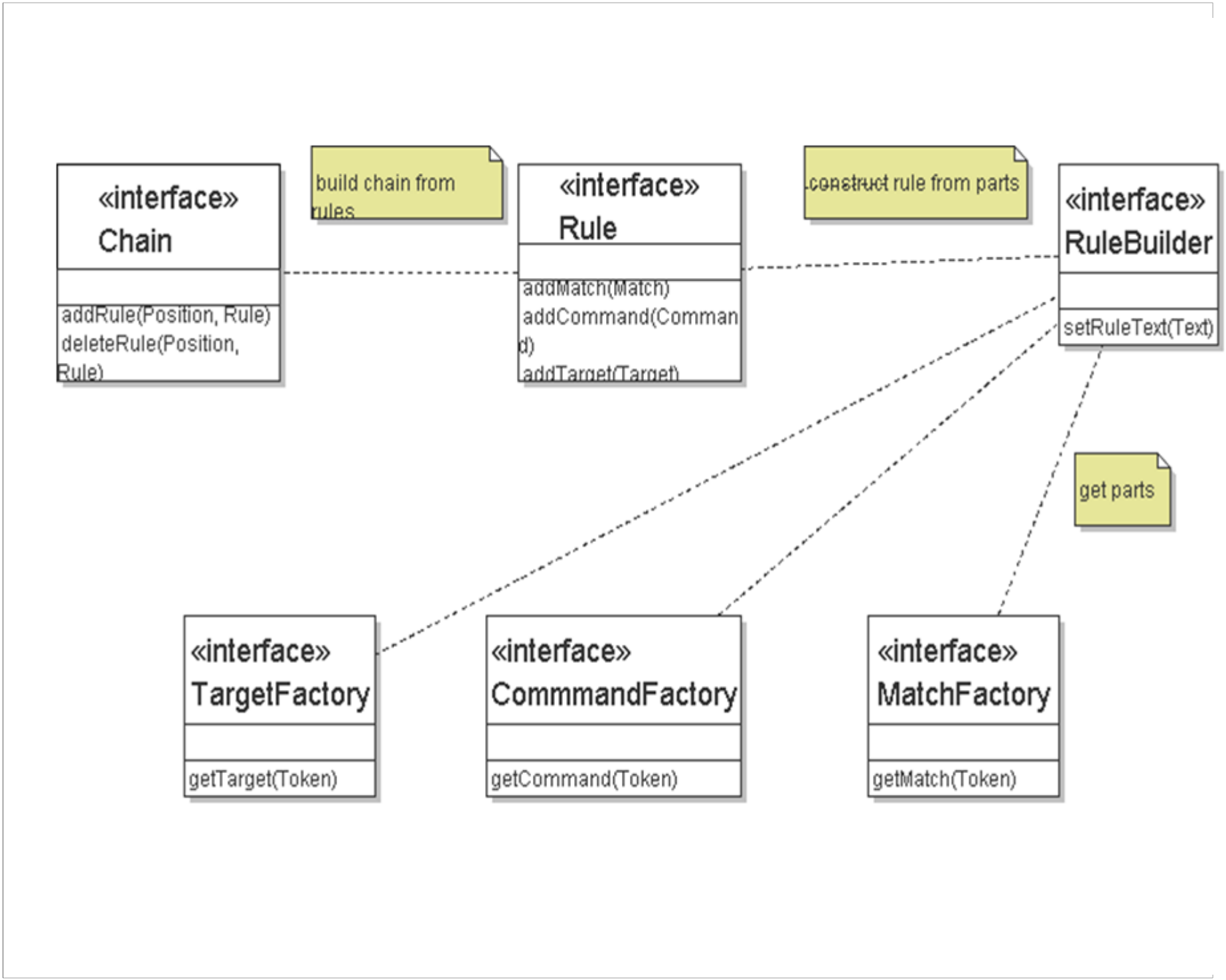
• iptables –T FILTER –A INPUT –i $IFACE –p tcp –sport 80 –m state –state ESTABLISHED –j ACCEPT (allow incoming web traffic if it belongs to a previous outgoing request)

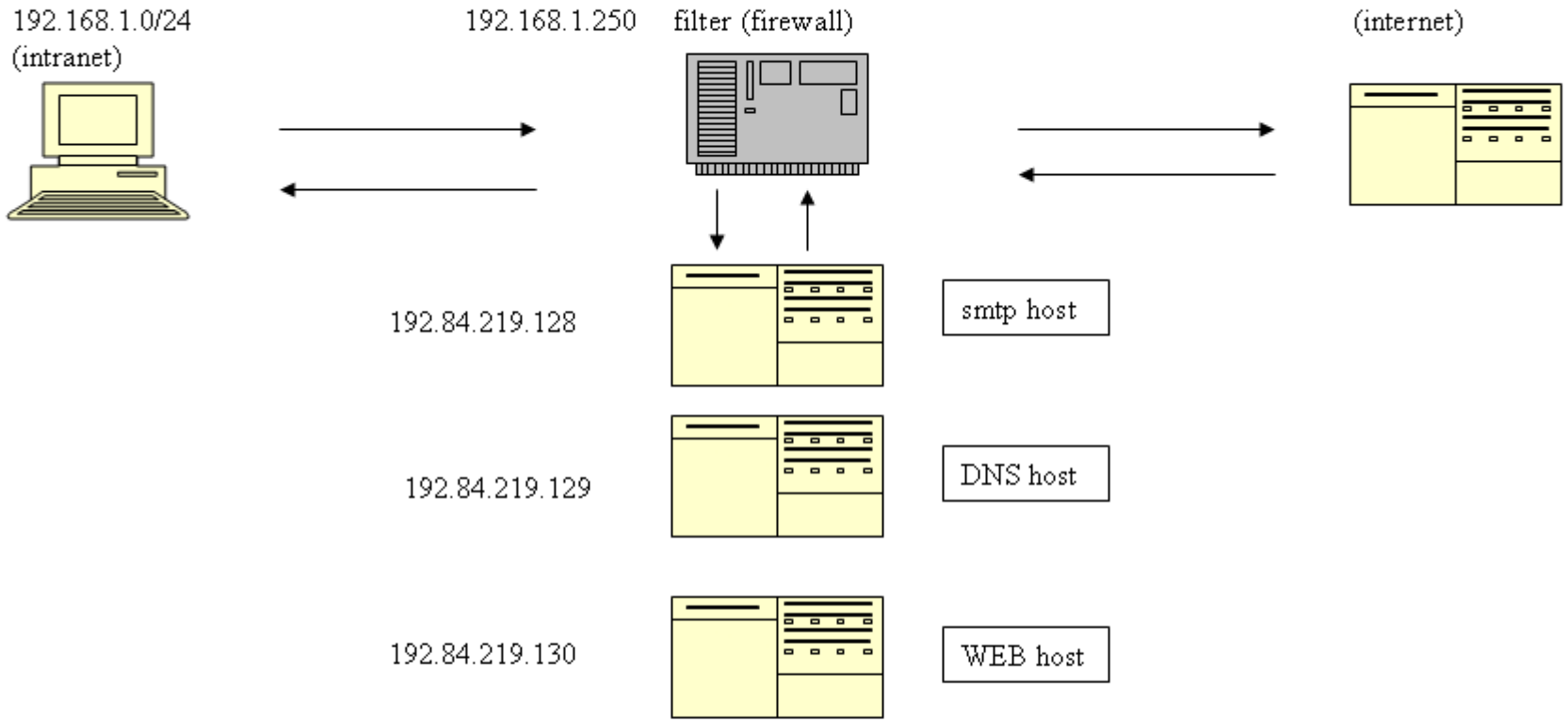
• iptables –A INPUT –i $IFACE –p tcp –sport 20 –m state –state ESTABLISHED, RELATED –j ACCEPT (allow incoming ACTIVE ftp traffic if it belongs to a previous outgoing request, even though the incoming request is for a new – but related - port)

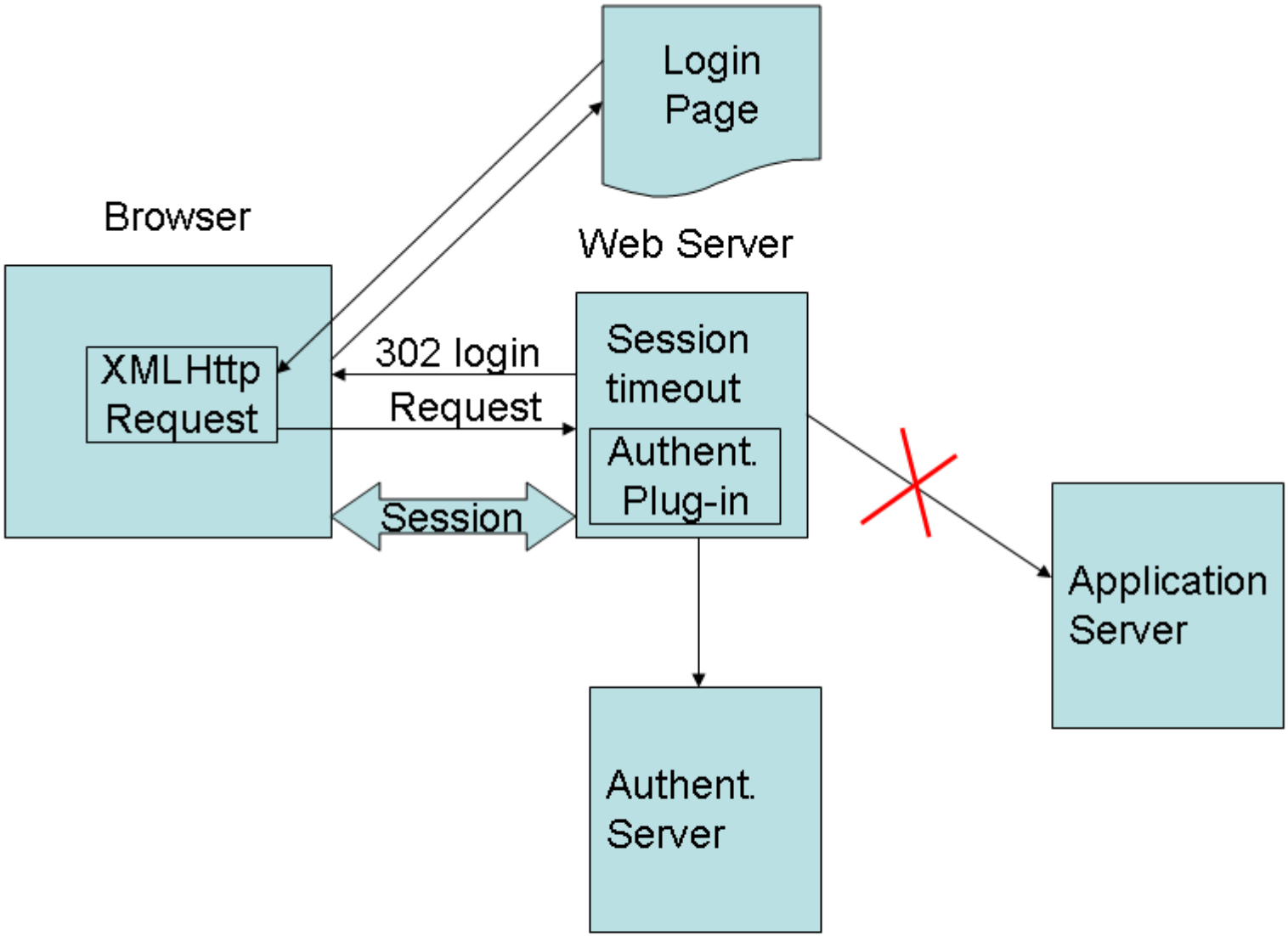
• iptables –A INPUT –i $IFACE – p udp –j LOG –log-prefix „UDP Incoming:“

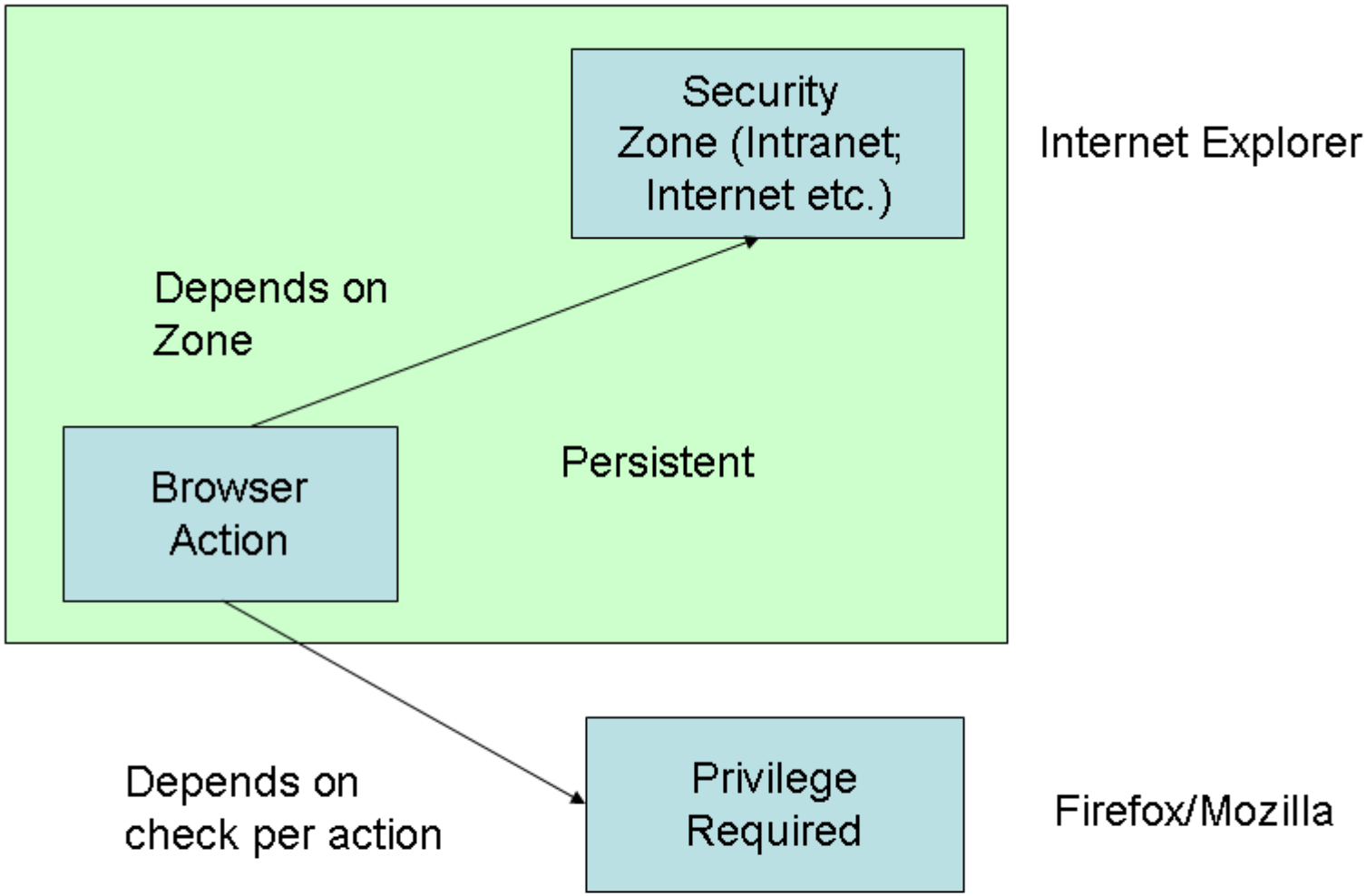
•iptables –A INPUT –i $IFACE – p udp –j DROP (log and drop all udp traffic)


«interface»
NetFilterFramework
registerTable(Table)
unRegisterTable(Table)
RegisterModule(Name, Module)
UnregisterModule(Name)
RegisterPatch(Name
register
«interface»
Table
Packet
prerouting(packet)
boolean input (packet)
boolean forward (packet)
boolean output (packet)
AbstractTable
Packet
prerouting(packet)
boolean input (packet)
boolean forward (packet)
boolean output (packet)
Filter
boolean input (packet)
boolean forward (packet)
Nat
Packet prerouting(packet)
Packet postrouting (packet)
Mangle
Packet prerouting(packet)
Packet postrouting (packet)
overwrite only filtering callbacks
Overwrite modifying callbacks


«interface»
Chain
addRule(Position, Rule)
deleteRule(Position,
Rule)
build chain from
rules
«interface»
Rule
addMatch(Match)
addCommand(Comman
d)
addTarget(Target)
construct rule from parts
«interface»
RuleBuilder
setRuleText(Text)
get parts
«interface»
TargetFactory
getTarget(Token)
«interface»
CommmandFactory
getCommand(Token)
«interface»
MatchFactory
getMatch(Token)


192.168.1.0/24
(intranet)
192.168.1.250
filter (firewall)
(internet)
192.84.219.128
smtp host
192.84.219.129
DNS host
192.84.219.130
WEB host


Login
Page
Browser
Web Server
XMLHttp
Request
302 login
Request
Session
timeout
Authent.
Plug-in
Session
Application
Server
Authent.
Server


Security
Zone (Intranet;
Internet etc.)
Internet Explorer
Depends on
Zone
Persistent
Browser
Action
Depends on
check per action
Privilege
Required
Firefox/Mozilla

Fundamental Questions

Input Validation

- Are Regexp checks enough?
- How do Servlet Filters work?
- How to separate Non-terminals from terminals?
- Forwarding of modified request data – the problem of double-decoding
- Is application input a language? Of what type? How expressed? Design question?
- Tainting as a software mechanism

Filtering

- Anti-patterns of filter use?
- Proof of correctness – is illegal input blocked?
- Proof of liveness – does legal input still get through?
- Mixing of reject and accept statements?
- Filter models and automated checkers?
- Filter positions in software?

Concurrency

- Libraries for safe shell programming?
- Is shared state multithreading reliable and predictable?
- Architectures for safe concurrency (Miller)?
- Active Objects, CSP etc.

Ambient Authority

- How to restrict system call access?
- How to prevent arbitrary initial authority?
- Software architectures to achieve loader isolation?
- Language features for secure software?
- Damage control features in operating systems, languages and applications

Signs and Minds

- How to avoid confusion about identity?
- How to represent system messages reliably and without chance for fake messages?
- Software technology to establish a trusted path for users?
- Charcter sets and representations as fonts?
- Reliable detection of character aliases?