

به نام خدا



«امنیت داده و شبکه»
دکتر امینی

تمرین چهارم

علیرضا دهقانپور فراشاه
۹۸۱۰۱۵۵۵

سوال اول	2
سوال دوم	9
سوال سوم	14
سوال چهارم	18

سوال اول

```
views.py 2 ×  urls.py 1
C: > Users > alireza > Desktop > HW4 > vuln web > xss > views.py > ...
4  # Create your views here.
5  def index(request):
6      return render(request, "xss.html")
7
8  def l1(request):
9      if request.method == "GET":
10         context = {'name': 'Anonymous User'}
11         #template = loader.get_template("./xss/templates/xss_l1.html")
12         return render(request, 'xss_l1.html', context)
13     elif request.method == "POST":
14         name = request.POST.get('name', 'Anonymous User')
15         context = {'name': name}
16         return render(request, 'xss_l1.html', context)
17     else:
18         return "Error"
19
20 def l2(request):
21     if request.method == "GET":
22         context = {'name': 'Anonymous User'}
23         #template = loader.get_template("./xss/templates/xss_l2.html")
24         return render(request, 'xss_l2.html', context)
25     elif request.method == "POST":
26         name = request.POST.get('name', 'Anonymous User')
27         context = {'name': name}
28         return render(request, 'xss_l2.html', context)
29     else:
30         return "Error"
31
```

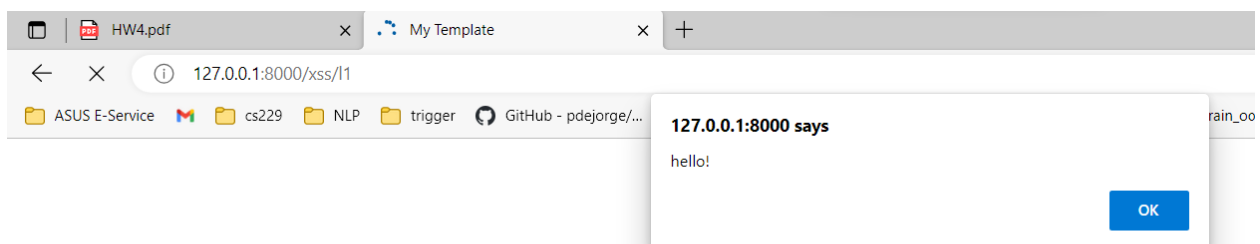
همانطور که در کد مشخص است ابتدا صفحه‌ی XSS رندر می‌شود و وقتی وارد چالش اول یا دوم می‌شویم وارد url آن چالش می‌شویم که توابع l1 و l2 برای آن‌ها است. وقتی وارد آن می‌شویم اولین بار با متد GET وارد می‌شویم که نام ما برابر Anonymous User است. ولی اگر با متد POST باشد نام وارد شده را در یک دیکشنری قرار می‌دهد و صفحه را render می‌کند. حال در ادامه به کد html دو چالش می‌پردازیم.

```
views.py 2  xss_l1.html x  urls.py 1
C: > Users > alireza > Desktop > HW4 > vuln web > xss > templates > xss_l1.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>My Template</title>
5  </head>
6  <body>
7      {% autoescape off %}
8      <h1>Hello, {{ name }}!</h1>
9      {% endautoescape %}
10     <form method="POST" action="" >
11         {% csrf_token %}
12         <label for="name">Enter your name:</label>
13         <input type="text" id="name" name="name">
14         <button type="submit">Submit</button>
15     </form>
16     <br>
17     <a href="{% url 'l2' %}">Next XSS Challenge</a>
18 </body>
19 </html>
```

در کد html چالش اول از name در میان یک تگ h1 استفاده شده است. می توانیم با ورودی دادن یک کد javascript کد خود را اجرا کنیم و یک alert در کد قرار دهیم. برای چالش اول از ورودی زیر استفاده می کنیم.

`<script>alert("hello!")</script>`

می توانید نتیجه ی این چالش را در تصویر زیر مشاهده کنید.

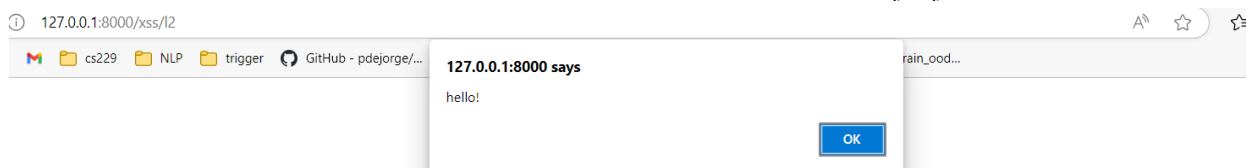


در ادامه کد html چالش دوم را مشاهده می کنید.

```
xss_l1.html xss_l2.html X
C: > Users > alireza > Desktop > HW4 > vuln web > xss > templates > xss_l2.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>My Template</title>
5 </head>
6 <body>
7   {% autoescape off %}
8   <h1 id="header">Level 2: Hello, <script>
9     var myText = "Level 2: Hello, {{ name }}!";
10    var myParagraph = document.getElementById("header");
11    myParagraph.innerHTML = myText;
12  </script>
13 </h1>
14   {% endautoescape %}
15   <form method="POST" action="" >
16     {% csrf_token %}
17     <label for="name">Enter your name:</label>
18     <input type="text" id="name" name="name">
19     <button type="submit">Submit</button>
20   </form>
21   <br>
22   <a href="{% url 'l1' %}">Previous XSS Challenge</a>
23 </body>
24 </html>
```

در اینجا در تعریف متغیر از name استفاده شده است. برای حمله به این بخش ابتدا رشته را پایان می دهیم و سپس کد html خود را در ادامه می گذاریم. برای حمله از کد زیر استفاده می کنیم.
"; </script>

نتیجه ی این حمله را در تصویر زیر مشاهده می کنید.



برای چالش SQL ابتدا به کد مراجعه می کنیم.

```

C: > Users > alireza > Desktop > HW4 > vuln web > sql > views.py > ...
1  from django.shortcuts import render
2  from django.db import connection
3  from django.db.models.expressions import RawSQL
4  from .models import Student
5
6  # Create your views here.
7
8
9  def index(request):
10     if request.method == "GET":
11         context = {'results': ''}
12         return render(request, 'sqli.html', context)
13     elif request.method == "POST":
14         username = request.POST.get('username', '')
15         password = request.POST.get('password', '')
16         cursor = connection.cursor()
17         print("SELECT * FROM sqli_student WHERE username = '" + username + "' AND password = '" + password + "'")
18         cursor.execute("SELECT * FROM sqli_student WHERE username = '" + username + "' AND password = '" + password + "'")
19         results = cursor.fetchone()
20         return render(request, 'sqli.html', {'results': results})
21     else:
22         return "Error"

```

برای حمله کافی است در password یک شرط جدید اضافه کنیم.

username=attack

password=attack' OR 1=1 or '

```

Windows PowerShell
PS C:\Users\alireza\Desktop\HW4\vuln web> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 16, 2023 - 18:06:02
Django version 4.2, using settings 'vuln.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[16/Jun/2023 18:06:05] "GET /sqli/ HTTP/1.1" 200 675
SELECT * FROM sqli_student WHERE username = 'attack' AND password = 'attack' or 1=1 or '
[16/Jun/2023 18:06:23] "POST /sqli/ HTTP/1.1" 200 682

```

نتیجه‌ی اجرای حمله

Enter your username and password to log in

Enter your username:

Enter your password:

Login Successful

برای جلوگیری از حمله‌های XSS می‌توان از escape استفاده کرد که به این معنی است که شما کاراکترهای کلیدی داده را تبدیل می‌کنید (یا علامت گذاری می‌کنید) تا از تفسیر آن در یک زمینه خطرناک جلوگیری کنید.

برای این منظور کدهای html دو چالش را به صورت زیر تغییر می‌دهیم.

```
C: > Users > alireza > Desktop > HW4 > vuln web > xss > templates > xss_l1.html > html > body > h1
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>My Template</title>
5  </head>
6  <body>
7  |   {% autoescape off %}
8  |   <h1>Hello, {{ name | escape }}!</h1>
9  |   {% endautoescape %}
10 |   <form method="POST" action="" >
11 |       {% csrf_token %}
12 |       <label for="name">Enter your name:</label>
13 |       <input type="text" id="name" name="name">
14 |       <button type="submit">Submit</button>
15 |   </form>
16 |   <br>
17 |   <a href="{% url 'l2' %}">Next XSS Challenge</a>
18 </body>
19 </html>
```

```
<> xss_l1.html    <> xss_l2.html X
C: > Users > alireza > Desktop > HW4 > vuln web > xss > templates > xss_l2.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>My Template</title>
5  </head>
6  <body>
7  |   {% autoescape off %}
8  |   <h1 id="header">Level 2: Hello, <script>
9  |       var myText = "Level 2: Hello, {{ name | escape }}!";
10 |       var myParagraph = document.getElementById("header");
11 |       myParagraph.innerHTML = myText;
12 |   </script>
13 | </h1>
14 |   {% endautoescape %}
15 |   <form method="POST" action="" >
16 |       {% csrf_token %}
17 |       <label for="name">Enter your name:</label>
18 |       <input type="text" id="name" name="name">
19 |       <button type="submit">Submit</button>
20 |   </form>
21 |   <br>
22 |   <a href="{% url 'l1' %}">Previous XSS Challenge</a>
23 </body>
24 </html>
```

برای جلوگیری از SQL Injection از توابع خود Django استفاده می‌کنیم و کد ما به صورت زیر می‌شود.


```

C: > Users > alireza > Desktop > HW4 > vuln web > sqli > views.py > index
1  from django.shortcuts import render
2  from django.db import connection
3  from django.db.models.expressions import RawSQL
4  from .models import Student
5
6  # Create your views here.
7
8
9  def index(request):
10     if request.method == "GET":
11         context = {'results': ''}
12         return render(request, 'sqli.html', context)
13     elif request.method == "POST":
14         username = request.POST.get('username', '')
15         password = request.POST.get('password', '')
16         try:
17             student = Student.objects.get(username=username, password=password)
18             results = True
19         except Student.DoesNotExist:
20             results = False
21
22         return render(request, 'sqli.html', {'results': results})
23     else:
24         return "Error"

```

توجه کنید دیگر Query به صورت مستقیم زده نمی‌شود و Django خودش کوئری را هندل می‌کند. اگر نیاز بود Query بزنی می‌توانستیم از raw() برای کلاس مورد نظر استفاده کنیم.

منابع

<https://portswigger.net/web-security/cross-site-scripting>.

<https://portswigger.net/web-security/cross-site-scripting/preventing>

<https://stackoverflow.com/questions/2159724/how-can-escaping-be-used-to-prevent-xss-attacks>

<https://www.stackhawk.com/blog/sql-injection-prevention-django/>

سوال دوم

1) یک فایل با نام q2.txt درست کرده و نام خود را در آن نوشته‌ام.

```
Windows PowerShell
PS C:\Users\alireza\Desktop\HW4> type q2.txt
alireza
PS C:\Users\alireza\Desktop\HW4> |
```

2) حال Integrity Level آن را به high تغییر می‌دهیم.

```
Administrator: Command Prompt
C:\Users\alireza\Desktop\HW4>.\chml.exe q2.txt -i:h

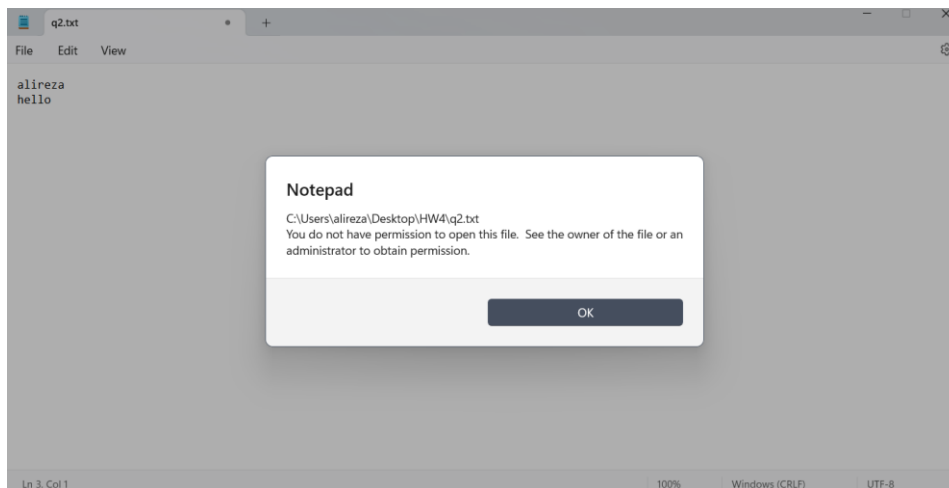
Chml v1.53 -- View and change Windows file/folder integrity levels
by Mark Minasi (c) 2006-2009 www.minasi.com email:help@minasi.com

Successfully changed q2.txt's integrity level.

File q2.txt's integrity level: High
Inheritance flags:
  No inheritance flags
Integrity policies:
  No read up: disabled
  No execute up: disabled
  No write up: enabled

C:\Users\alireza\Desktop\HW4>_
```

برای خواندن مشکلی ایجاد نمی‌شود و فایل باز می‌شود ولی هنگام نوشتن و ذخیره کردن با خطای زیر رو به رو می‌شویم. علت آن نیز این است که هنگامی که integrity را به high تغییر دادیم همانطور که در تصویر بالا نیز مشخص است سیاست No write up فعال یا enable می‌شود ولی سایر سیاست‌ها تغییر نمی‌کند. کاربر عادی در ویندوز دارای سطح medium است و نمی‌تواند در فایلی که high است و سیاست ننوشتن در آن فعال است چیزی بنویسد.



3) اکنون سیاست No read up را فعال می‌کنیم و نتایج را در تصاویر زیر مشاهده می‌کنید.

```
Administrator: Command Prompt

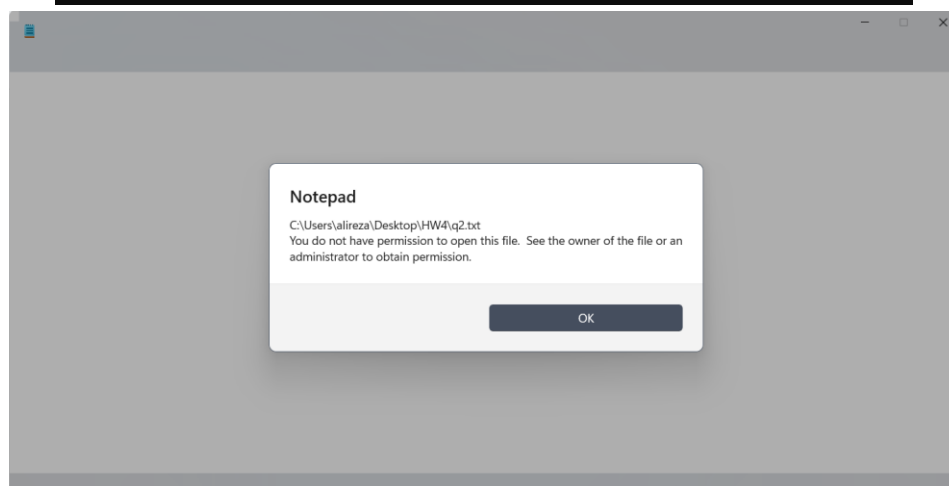
C:\Users\alireza\Desktop\HW4>.\chml.exe q2.txt -i:h -nr

Chml v1.53 -- View and change Windows file/folder integrity levels
by Mark Minasi (c) 2006-2009 www.minasi.com email:help@minasi.com

Successfully changed q2.txt's integrity level.

File q2.txt's integrity level: High
Inheritance flags:
  No inheritance flags
Integrity policies:
  No read up: enabled
  No execute up: disabled
  No write up: disabled

C:\Users\alireza\Desktop\HW4>
```



4) کد به زبان C را در تصویر زیر مشاهده می‌کنید.

```

int main() {
    FILE *ptr;
    char ch;
    ptr = fopen("q2.txt", "r");
    if (NULL == ptr) {
        printf("can not open the file to read!\n");
    }
    if (ptr != NULL) {
        printf("content of the file is:\n");
        while ((ch = fgetc(ptr)) != EOF) {
            printf("%c", ch);
        }
        printf("\nend of the file\n");
        fclose(ptr);
    }
    ptr = fopen("q2.txt", "w");
    if (ptr == NULL) {
        printf("can not open the file to write!\n");
    }
    if (ptr != NULL) {
        fprintf(ptr, "%s", "modified!");
        fclose(ptr);
        printf("write successfully to the file\n");
    }
    ch = getchar();
    return 0;
}

```

5) ابتدا سطح integrity دو فایل را تغییر می‌دهیم و در شکل زیر مشاهده می‌کنید.

```

C:\Users\alireza\Desktop\HW4>.chml.exe q2.txt -i:m

Chml v1.53 -- View and change Windows file/folder integrity levels
by Mark Minasi (c) 2006-2009 www.minasi.com email:help@minasi.com

Successfully changed q2.txt's integrity level.

File q2.txt's integrity level: Medium
Inheritance flags:
  No inheritance flags
Integrity policies:
  No read up: disabled
  No execute up: disabled
  No write up: enabled

C:\Users\alireza\Desktop\HW4>.chml.exe Code.exe -i:l

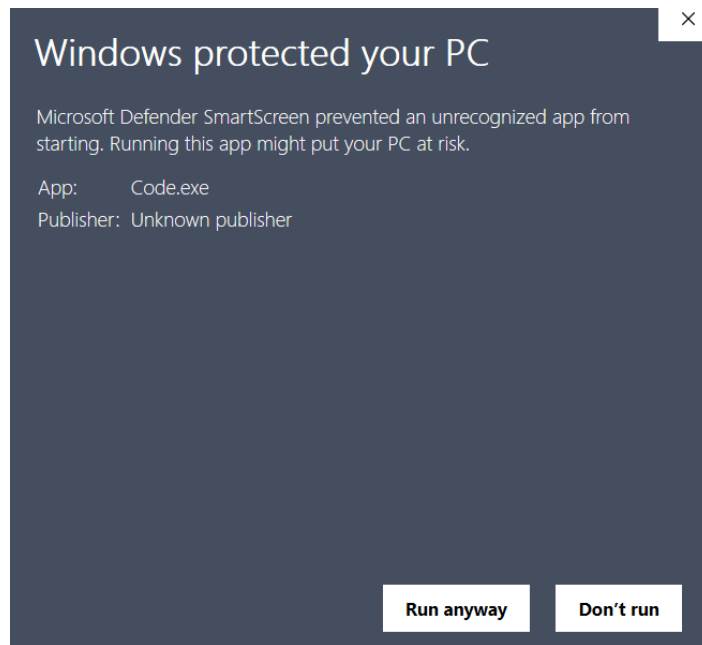
Chml v1.53 -- View and change Windows file/folder integrity levels
by Mark Minasi (c) 2006-2009 www.minasi.com email:help@minasi.com

Successfully changed Code.exe's integrity level.

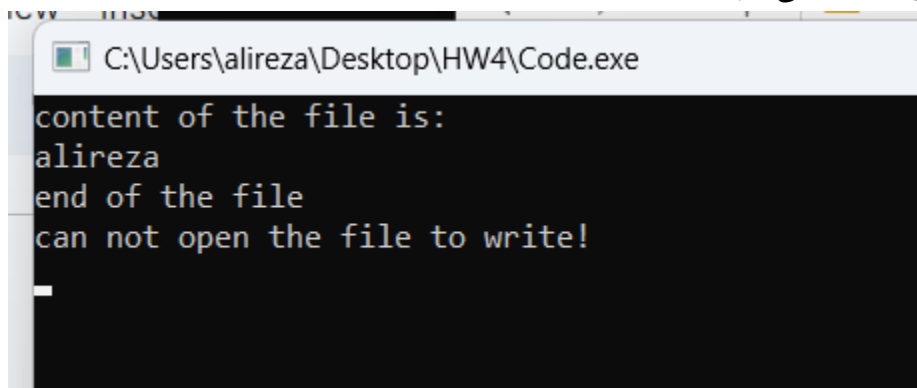
File Code.exe's integrity level: Low
Inheritance flags:
  No inheritance flags
Integrity policies:
  No read up: disabled
  No execute up: disabled
  No write up: enabled

```

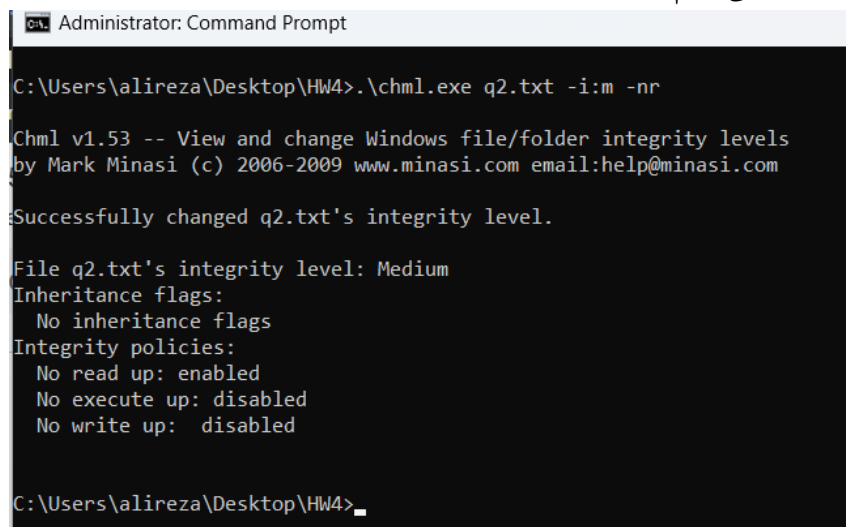
حال فایل exe را اجرا می‌کنیم.



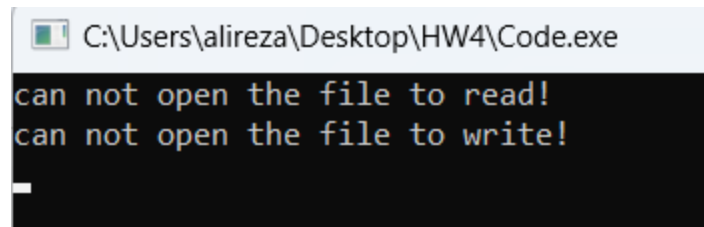
و نتیجه‌ی اجرا را مشاهده می‌کنید.



حال No read up را فعال می‌کنیم.



حال نتیجه‌ی اجرای فایل اجرایی را مشاهده می‌کنید که دیگر اجازه‌ی خواندن نیز ندارد.



(6) برای این بخش ابتدا integrity فایل اجرایی برابر medium و فایل متنی را برابر low قرار می‌دهیم.

```
Administrator: Command Prompt

C:\Users\alireza\Desktop\HW4>.\chml.exe q2.txt -i:l -nr -nw

Chml v1.53 -- View and change Windows file/folder integrity levels
by Mark Minasi (c) 2006-2009 www.minasi.com email:help@minasi.com

Successfully changed q2.txt's integrity level.

File q2.txt's integrity level: Low
Inheritance flags:
  No inheritance flags
Integrity policies:
  No read up: enabled
  No execute up: disabled
  No write up: enabled

C:\Users\alireza\Desktop\HW4>.\chml.exe Code.exe -i:m

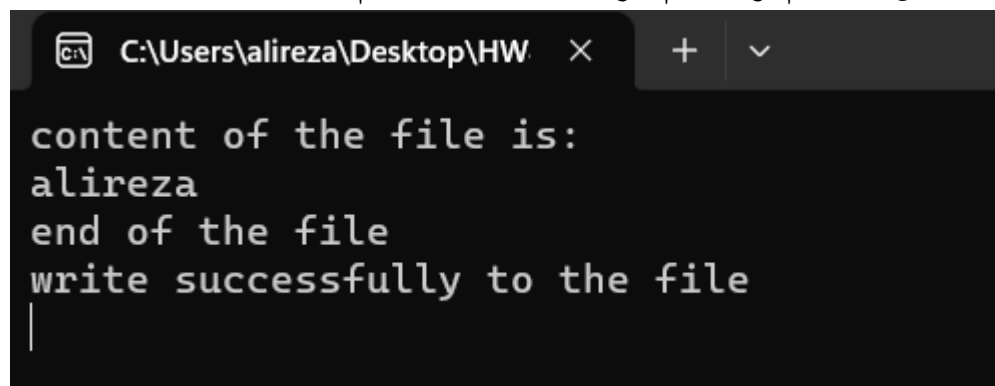
Chml v1.53 -- View and change Windows file/folder integrity levels
by Mark Minasi (c) 2006-2009 www.minasi.com email:help@minasi.com

Successfully changed Code.exe's integrity level.

File Code.exe's integrity level: Medium
Inheritance flags:
  No inheritance flags
Integrity policies:
  No read up: disabled
  No execute up: disabled
  No write up: enabled

C:\Users\alireza\Desktop\HW4>_
```

همانطور که مشخص است هم نوشتن هم خواندن بدون مشکل انجام شده است.



سوال سوم

ابتدا جداول گفته شده را با دستورات زیر می‌سازیم.

Query	Query History
1	CREATE TABLE PHYSICIANS (
2	name VARCHAR(64),
3	specialist VARCHAR(64),
4	phone_number VARCHAR(64),
5	national_id VARCHAR(10),
6	PRIMARY KEY (national_id)
7);

Query	Query History
1	CREATE TABLE PATIENTS (
2	name VARCHAR(64),
3	sickness VARCHAR(64),
4	phone_number VARCHAR(64),
5	national_id VARCHAR(10),
6	PRIMARY KEY (national_id)
7);

Query	Query History
1	CREATE TABLE SICKNESS (
2	sickness VARCHAR(64),
3	national_id VARCHAR(10),
4	FOREIGN KEY (national_id) REFERENCES PATIENTS (national_id),
5	PRIMARY KEY (national_id, sickness)
6);

Query	Query History
1	CREATE TABLE RECOGNITIONS (
2	patient_name VARCHAR(64),
3	physician_name VARCHAR(64),
4	sickness VARCHAR(64)
5);

سپس با دستور زیر تعدادی داده وارد جدول می‌کنیم.

Query	Query History
1	INSERT INTO PHYSICIANS (name, specialist, phone_number, national_id)
2	VALUES ('John Doe', 'Nephrologist', '5555555555', '1234567890');
3	
4	INSERT INTO PATIENTS (name, sickness, phone_number, national_id)
5	VALUES ('Jane Smith', 'Diabetes', '1234567890', '1234567891');
6	
7	INSERT INTO SICKNESS (sickness, national_id)
8	VALUES ('Diabetes', '1234567891');
9	
10	INSERT INTO RECOGNITIONS (patient_name, physician_name, sickness)
11	VALUES ('Jane Smith', 'John Doe', 'Diabetes');

حال با دستورات زیر خواسته‌ی سوال را ایجاد می‌کنیم.

Query	Query History
1	CREATE ROLE Normal_User;
2	GRANT SELECT(name, specialist) ON TABLE PHYSICIANS TO Normal_User;
3	CREATE USER normaluser with password '98101555';
4	GRANT Normal_User to normaluser;
5	
6	CREATE ROLE Nurse;
7	GRANT ALL PRIVILEGES ON TABLE PATIENTS, SICKNESS, RECOGNITIONS TO Nurse;
8	GRANT SELECT ON TABLE PHYSICIANS TO Nurse;
9	CREATE USER nurse1 with password '98101555';
10	GRANT Nurse to nurse1;
11	
12	CREATE ROLE Physician;
13	GRANT SELECT, UPDATE ON TABLE PATIENTS, SICKNESS TO Physician;
14	CREATE USER physician1 with password '98101555';
15	GRANT Physician to physician1;
16	

حال باید با user های ساخته شده وارد شویم و تست کنیم.

• بررسی Normal User

در شکل زیر مشاهده می‌کنید که دسترسی به نام دکترها را دارد.

Query	Query History
1	select name from physicians;

Data Output	Messages	Notifications
<div> <div>name</div> <div>character varying (64)</div> <div>1</div> <div>John Doe</div> </div>		

اما نمی‌تواند کل جدول را بخواند.


```

Query    Query History
1  select * from physicians;

Data Output    Messages    Notifications
ERROR: permission denied for table physicians
SQL state: 42501

```

- بررسی Physician پزشک می‌تواند از جدول بیماران بخواند.

```

Query    Query History
1  select * from patients;

Data Output    Messages    Notifications

```

	name character varying (64)	sickness character varying (64)	phone_number character varying (64)	national_id [PK] character varying (10)
1	Jane Smith	Diabetes	1234567890	1234567891

توانایی آپدیت روی جدول را نیز دارد.

```

Query    Query History
1  update patients set phone_number = '4444' where national_id = '1234567891';

Data Output    Messages    Notifications
UPDATE 1
Query returned successfully in 518 msec.

```

اما امکان insert در جدول را ندارد. توجه کنید برای جدول بیماری‌ها نیز به همین صورت است.

```

Query    Query History
1  INSERT INTO PATIENTS (name, sickness, phone_number,national_id)
2  VALUES ('alireza', 'flu', '0142', '1234567899');

Data Output    Messages    Notifications
ERROR: permission denied for table patients
SQL state: 42501

```

- بررسی Nurse برای نمونه توانایی خواندن روی جدول بیماری را دارند.

Query Query History							
1 select * from sickness							
Data Output Messages Notifications							
<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>							
	<table> <tr> <th>sickness</th><th>national_id</th></tr> <tr> <td>[PK] character varying (64)</td><td>[PK] character varying (10)</td></tr> <tr> <td>1 Diabetes</td><td>1234567891</td></tr> </table>	sickness	national_id	[PK] character varying (64)	[PK] character varying (10)	1 Diabetes	1234567891
sickness	national_id						
[PK] character varying (64)	[PK] character varying (10)						
1 Diabetes	1234567891						

همچنین امکان اضافه کردن داده به جدول تشخیصات را نیز دارند.

Query Query History	
1 INSERT INTO RECOGNITIONS (patient_name, physician_name, sickness) 2 VALUES ('Alireza', 'John Doe', 'flu');	
Data Output Messages Notifications	
INSERT 0 1 Query returned successfully in 123 msec.	

ولی برای مثال امکان حذف از جدول پزشکان ندارند.

Query Query History	
1 delete from physicians where national_id='1234567890';	
Data Output Messages Notifications	
ERROR: permission denied for table physicians SQL state: 42501	

منابع

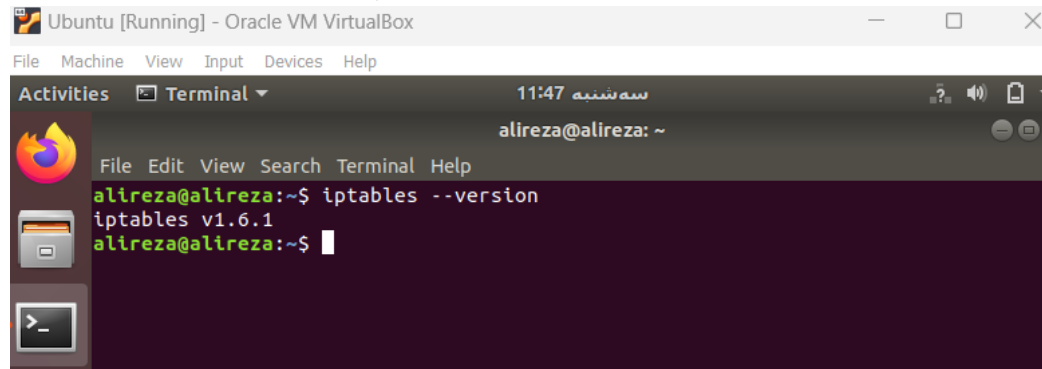
<https://tableplus.com/blog/2018/04/postgresql-how-to-grant-access-to-users.html>

<https://dba.stackexchange.com/questions/151674/how-to-change-username-to-connect-to-server-in-pgadmin4-in-query-tool>

سوال چهارم

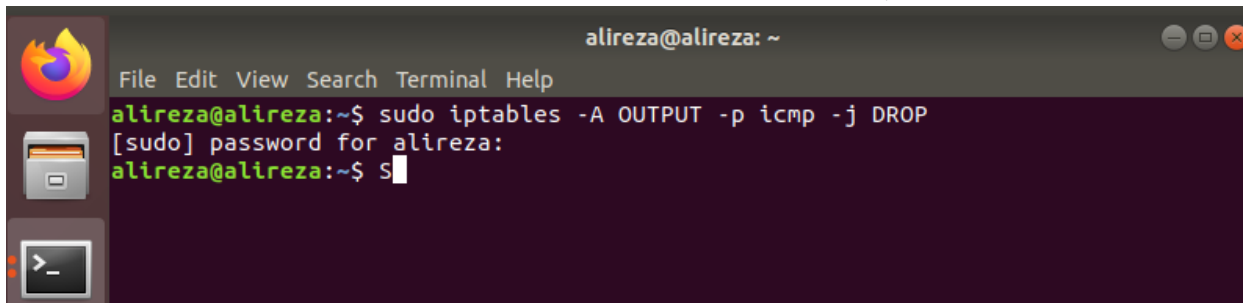
بخش اول

در ابتدا ورژن iptables را در ماشین مجازی مشاهده می کنیم.



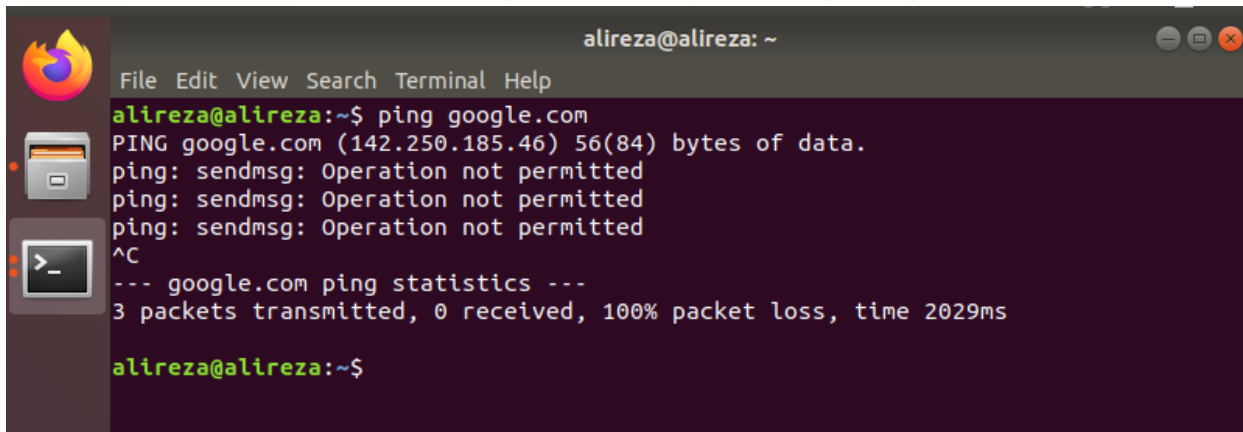
```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal 11:47 سه شنبه
alireza@alireza: ~
File Edit View Search Terminal Help
alireza@alireza:~$ iptables --version
iptables v1.6.1
alireza@alireza:~$
```

- برای این بخش برای تست از دستور ping استفاده می کنیم که این دستور بسته های ICMP ارسال می کند. سپس دستور زیر را می زنیم:



```
alireza@alireza: ~
File Edit View Search Terminal Help
alireza@alireza:~$ sudo iptables -A OUTPUT -p icmp -j DROP
[sudo] password for alireza:
alireza@alireza:~$ S
```

سپس google را ping می کنیم و پیغام خطای زیر را دریافت می کنیم.



```
alireza@alireza: ~
File Edit View Search Terminal Help
alireza@alireza:~$ ping google.com
PING google.com (142.250.185.46) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- google.com ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2029ms

alireza@alireza:~$
```

- برای این کار بایست از یک سیستم دیگر به ماشین مجازی ssh بزنیم. برای این کار openssh-server را نصب می‌کنیم و وضعیت آن را با دستور زیر مشاهده می‌کنیم.

```
alireza@alireza: ~
File Edit View Search Terminal Help
Creating SSH2 RSA key; this may take some time ...
2048 SHA256:pNJKsBLnukmkQV/XUuYymDLuyYCJcbXJDVYehW94WLY root@alireza (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:0Jcv5575v3Utooc52Fm6c7YeLap3s25xm6MEj3cfyPk root@alireza (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:WH97sk0o55gMVdsmfZNTq05nFBdieYum9Bd7B0WW90A root@alireza (ED25519)
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.52) ...
alireza@alireza:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: ena
   Active: active (running) since Tue 2023-06-20 12:27:23 +0430; 35s ago
     Main PID: 2126 (sshd)
       Tasks: 1 (limit: 4664)
      CGroup: /system.slice/ssh.service
              └─2126 /usr/sbin/sshd -D

شماره 20 12:27:23 alireza systemd[1]: Starting OpenBSD Secure Shell server...
شماره 20 12:27:23 alireza sshd[2126]: Server listening on 0.0.0.0 port 22.
شماره 20 12:27:23 alireza sshd[2126]: Server listening on :: port 22.
شماره 20 12:27:23 alireza systemd[1]: Started OpenBSD Secure Shell server.
lines 1-12/12 (END)
```

حال از دستگاه host به ماشین مجازی ssh می‌زنیم و نتیجه‌ی آن به شرح زیر است.

```
alireza@alireza: ~
C:\Users\alireza>ssh alireza@192.168.1.106
The authenticity of host '192.168.1.106 (192.168.1.106)' can't be established.
ED25519 key fingerprint is SHA256:WH97sk0o55gMVdsmfZNTq05nFBdieYum9Bd7B0WW90A.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.106' (ED25519) to the list of known hosts.
alireza@192.168.1.106's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-84-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

297 updates can be applied immediately.
263 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Your Hardware Enablement Stack (HWE) is supported until April 2023.

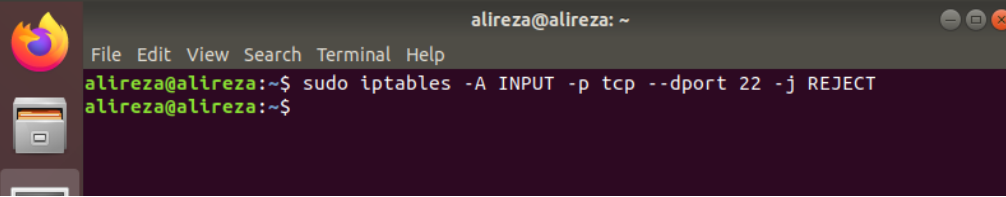
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

alireza@alireza:~$
```

حال دستور زیر را در iptables می‌زنیم که ترافیک پروتکل tcp به پورت ورودی ۲۲ را reject کند.

دوباره

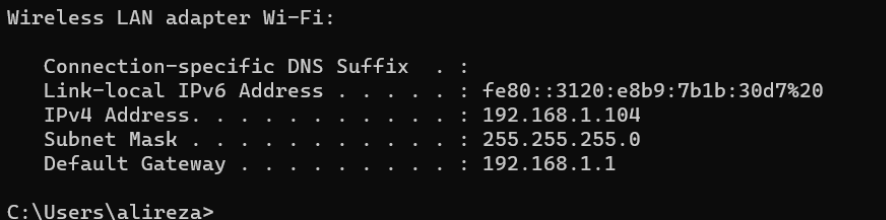


```
alireza@alireza: ~  
File Edit View Search Terminal Help  
alireza@alireza:~$ sudo iptables -A INPUT -p tcp --dport 22 -j REJECT  
alireza@alireza:~$
```


تلاش می‌کنیم تا ssh بزیم ولی نتیجه به شرح زیر است.

```
C:\Users\alireza>ssh alireza@192.168.1.106  
ssh: connect to host 192.168.1.106 port 22: Connection timed out  
C:\Users\alireza>
```

- ابتدا ip دستگاه host و ماشین مجازی را بدست می‌آوریم

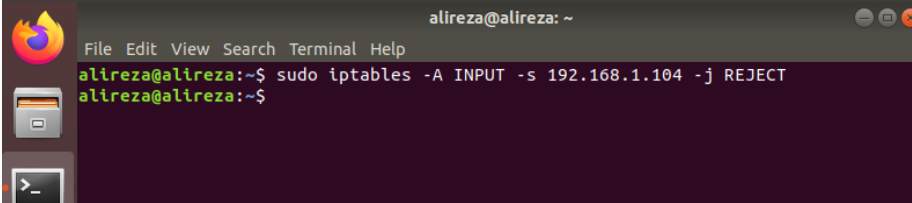


```
Wireless LAN adapter Wi-Fi:  
  
Connection-specific DNS Suffix . . :  
Link-local IPv6 Address . . . . . : fe80::3120:e8b9:7b1b:30d7%20  
IPv4 Address. . . . . : 192.168.1.104  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.1.1  
C:\Users\alireza>
```



```
alireza@alireza: ~  
File Edit View Search Terminal Help  
alireza@alireza:~$ ip r  
default via 192.168.1.1 dev enp0s3 proto dhcp metric 100  
default via 10.0.3.2 dev enp0s8 proto dhcp metric 101  
10.0.3.0/24 dev enp0s8 proto kernel scope link src 10.0.3.15 metric 101  
169.254.0.0/16 dev enp0s8 scope link metric 1000  
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.106 metric 100  
alireza@alireza:~$
```

سپس با دستورات زیر ترافیک ورودی از این ip را ریجکت می‌کنیم.



```
alireza@alireza: ~  
File Edit View Search Terminal Help  
alireza@alireza:~$ sudo iptables -A INPUT -s 192.168.1.104 -j REJECT  
alireza@alireza:~$
```

نتیجه‌ی Ping کردن ماشین مجازی به شرح زیر است.

```
C:\Users\alireza>ping 192.168.1.106  
  
Pinging 192.168.1.106 with 32 bytes of data:  
Reply from 192.168.1.106: Destination port unreachable.  
Reply from 192.168.1.106: Destination port unreachable.  
Reply from 192.168.1.106: Destination port unreachable.  
Reply from 192.168.1.106: Destination port unreachable.  
  
Ping statistics for 192.168.1.106:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
C:\Users\alireza>
```

بخش دوم

- iptables -A INPUT -p tcp --dport 22 ! -s 192.168.1.0/24 -j ACCEPT
- iptables -A INPUT -m state ! --state ESTABLISHED -j REJECT
- iptables -t nat -I PREROUTING -p tcp --dport 80 -j DNAT --to-destination <ip:port>
- iptables -t nat -I PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080

- برای این بخش سه rule به صورت زیر تعریف می‌کنیم. در قانون اول اگر تعداد connection برای یک IP مشخص بیشتر از ۱۱۱ باشد بسته REJECT می‌شود. در ادامه برای دنبال کردن اتصالات از conntrack استفاده می‌کنیم که اتصالات را ردیابی می‌کند. این ردیابی معمولاً به صورت یک جدول بزرگ با حداقل ۶ ستون اجرا می‌شود: پروتکل IP مبدا، پورت مبدا، IP مقصد، پورت مقصد و وضعیت اتصال. در قانون دوم چک می‌شود تعداد connection های جدید حداکثر ۲۰ تا باشد و تعداد درخواست‌های ارتباط جدید باید کمتر از ۶۰ تا در ثانیه باشد. اگر این شرایط برقرار بود accept می‌شود. در غیر این صورت بسته drop می‌شود.

```
iptables -A INPUT -p tcp -m connlimit --connlimit-above 111 -j REJECT
```

```
iptables -A INPUT -p tcp -m conntrack --ctstate NEW -m limit --limit 60/s --limit-burst 20 -j ACCEPT
```

```
iptables -A INPUT -p tcp -m conntrack --ctstate NEW -j DROP
```

منابع

<https://averagelinuxuser.com/ssh-into-virtualbox/>

<https://gist.github.com/mattia-beta/bd5b1c68e3d51db933181d8a3dc0ba64>

<https://superuser.com/questions/1071656/whats-the-difference-between-iptables-state-and-ctstate>