Secure Gateway & Information Security using Rubik's Cube Algorithm

J Component Report

Information Security Audit & Analysis

Fall 2021-2022

Submitted by

Anitej Srivastava (19BCE0835)

Kulvir Singh (19BCE2074)

under the guidance of

Dr. Siva Shanmugam

in partial fulfillment for the award of the degree of

B. Tech

in

Computer Science and Engineering

VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

November, 2021

Abstract

A directory traversal (or path traversal) attack exploits insufficient security validation or sanitization of user-supplied file names, such that characters representing "traverse to parent directory" are passed through to the operating system's file system API. An affected application can be exploited to gain unauthorized access to the file system.

Directory traversal is also known as the ../ (dot dot slash) attack, directory climbing, and backtracking. Some forms of this attack are also canonicalization attacks.

In this project, we demonstrate the directory traversal attacks that can be done using various techniques and we also show ways to mitigate it effectively.

Introduction

We will be exploiting the traversal vulnerability. The directory traversal and file traversal attacks will be dealt with through our project work. These attacks are majorly focused on web applications/sites and can allow the hacker to get access to secure files that the website creator wants to keep hidden from the public. A full demonstration of the attack will be shown with Kali Linux along with a process to mitigate it.

We will incorporate the algorithms and knowledge gained via recent articles/research papers to formulate a novel and improved technique/algorithm to counter and exterminate these vulnerabilities.

The most effective way to prevent file path traversal vulnerabilities is to avoid passing user-supplied input to the file system. The application would validate the user input before processing it. Ideally, the application would validate the canonical path of a file based on user input.

Survey Report

S.No	Title	Summary	Advantages & Disadvantages	Link
1.	[1]Michael Flanders, A Simple and Intuitive Algorithm for Preventing Directory Traversal Attacks (2019), Cornell Univ.	This paper presents an analysis of some currently used directory traversal attack defenses and presents a new, stack-based algorithm to help prevent these attacks by safely canonicalizing user-supplied path strings. The goal of this algorithm is to be small, easy to test, cross-platform compatible, and above all, intuitive.	The algorithm used is small, cost effective, intuitive, easy to test and above all, cross-platform. There is proof of correctness and verification strategies using symbolic execution for the algorithm. The algorithm could prove to be simple & ineffective for more complex attacks with more powerful computers.	https://arxiv.org/abs/1908.04502
2.	[2]Hegui Zhu, Lewen Dai, Yating Liu, Lijun Wu, A three-dimensional bit-level image encryption algorithm with Rubik's cube method, Mathematics and Computers in Simulation (2021)	In the permutation process, the Rubik's cube method and bit-level encryption principle are combined to realize the image scrambling operation in three-dimensional space. Through the contraction mapping principle, the ascending and reducing dimension operation can be implemented effectively. In the diffusion process, they design an improved two-dimensional diffusion structure, which can spread slight change in plain image to the whole cipher image.	Experimental results and simulation analysis illustrate the security and the validity of the proposed scheme. The 3-D bit level encryption proves to be costly and has high time complexity.	https://www.sciencedirect.com/science/article/abs/pii/S0378475421000483

3.	[3]N. Muraleedharan, Anna Thomas (2020), A Traffic Monitoring and Policy Enforcement Framework for HTTP	The paper describes a policy enforcement and web attack detection framework for HTTP protocol. The proposed framework can monitor and analyze HTTP traffic to detect injection, misconfiguration and directory traversal attacks. Moreover, this framework can be used to enforce web application access policies involving content type, URL and device level access.	The paper effectively monitors and analyses HTTP traffic to detect injection, misconfiguration and directory traversal attacks and exposes vulnerabilities effectively. One disadvantage is that this paper is not tested for very high traffic & could prove to be ineffective against severe DDoS attacks.	https://ieeexplore.ieee.org/abstract/document/9079333/authors#authors
4.	[4]Zainab S. Alwan1, Manal F. Younis2, Detection and Prevention of SQL Injection Attack: A Survey (2017)	This paper presents classical and modern types of SQLIA and display different existing technique and tools which are used to detect or prevent these attacks. Applications that do not properly validate the user's input make them vulnerable against SQL injection. This paper talks about SQL Injection Attacks & their occurence when an attacker is able to insert a series of malicious SQL statements into a —query‖ through manipulating user input data for execution by the back-end database.	This paper gives us an idea of the existing methodologies of SQLInjection detection & prevention in a descriptive manner & also about their advantages and shortcomings. A useful table is also given to summarise the entire survey. This paper does not present a new or improved methodology but only summarises the exiting ones. Some of the surveys may not act in preferred manner in certain situations.	https://www.researchgate.net/profile/Zainab-Alwan-5/publication/320108029_Detection_and_Prevention_of_SQL_Injection_Attack_A_Survey/links/59ce63840f7e9b4fd7e1b495/Detection-and-Prevention-of-SQL-Injection-Attack-A-Survey.pdf
5.	[5]Matthew Denis, Carlos Zena, (2016), Penetration testing: Concepts, attack methods, and defense strategies	In this paper, the researchers have performed different penetration tests using a private networks, devices, and virtualized systems and tools. They have used tools within the Kali Linux suite. The attacks performed	This paper details extensive penetrations testing using relevant methodologies & Kali Linux which is also used by attackers. This paper covers important aspects of smartphone & PC attacks & also exposes serious vulnerabilities in	https://ieeexplore.ieee.org/abstract/document/7494156

		included: smartphone penetration testing, hacking phones Bluetooth, traffic sniffing, hacking WPA Protected Wifi, Man-in-the-Middle attack, spying (accessing a PC microphone), hacking phones Bluetooth, and hacking remote PC via IP and open ports using advanced port scanner. The results are then summarized and discussed.	those systems. The paper does not discuss any relevant countermeasure or mitigation techniques for the vulnerabilities identified by the penetration tests.	

S.No	Title	Summary	Gap Analysis	Link
6	Rubik's cube principle based image encryption algorithm implementation on mobile devices. Ionescu, V. M., & Diaconu, A. V. (2015, June).	The idea developed around the use of Rubik's cube principle within image cryptosystems is that the processing This paper presents the implementation of a communication system between multiple mobile devices with imaging sensors, that encrypt the images using Rubik cube encryption algorithm, and a server. The performance and the suitability of the algorithm for mobile devices is investigated.	There is an in depth explanation about the Rubik’s cube algorithm and its process of implementation. Decryption module was not a primary focus for the paper. A web app implementation is not been stated	https:/ /ieeex plore.i eee.or g/abst ract/d ocum ent/73 01247

7	Analysis and Exploit of Directory Traversal Vulnerability on VMware Bai, Y., & Chen, Z. (2015, November).	This paper focuses on directory traversal vulnerability on VMware. The principle, triggering conditions and the exploit process of this vulnerability is discussed. Furthermore, the experimental environment is established to demonstrate the attack method. Experimental configurations and results discussion are given in detail. Finally, generalized recommendations are stated that can be applied to achieve secure virtualized implementations.	The paper gives a major insight to directory traversal attacks and the exploitation technique. This also gives us a better understanding of the tree structure of a directory which will help us in sanitizing. the query for mitigation. The paper does not discuss any specific countermeasure for a real environment. Moreover it has been performed on a virtual machine hence we cannot apply prevention methods given in a web app.	https:/ /www.spring erprofession al.de/en/analysis-and-exploit-of-directory-traversal-vulnerability-on-vmw/6961426

8	Detection of Web Application Attacks with Request Length Module and Regex Pattern Analysis Han, E. E. (2015, August)	This paper analyzes web server log files, which includes normal and malicious users' access patterns with their relevant links. This uses a web server log file dataset for the detection of web application attacks. This system intends to analyze normal and attack behaviors from web server log and then classify attack types which are included in the dataset. In this system, three types of attacks are detected namely, SQL injection, XSS and directory traversal attacks. Attack analysis stage is done by request length module and regular expressions for various attack patterns.	The attack detection is analyzed at large and a proper understanding of the attacks made on a web system is given. There is an effective use of regular expression at identifying attacks which will help us in our project. This paper does not give an insight to prevention of these attacks in a real time situation.	https:/ /link.springe r.com/chapt er/10. 1007/978-3- 319-23207- 2_16

9	**Information security risks management framework – A step towards mitigating security risks in university network** **Joshi, C., & Singh, U. K. (2017)**	**This paper analyzed the security threats specifically evolving in University's network, and with consideration of these issues, proposed an information security framework for University network environment. The proposed framework reduces the risk of security breach by supporting three phase activities; the first phase assesses the threats and vulnerabilities in order to identify the weak point in educational environment, the second phase focuses on the highest risk and create actionable remediation plan, the third phase of risk assessment model recognizes the vulnerability management compliance requirement in order to improve University's security position.**	**This paper deals with the attack and mitigation on a web server. It has staged the entire process in 3 phases. It identifies the weak point in the server and rectifies it.** **The paper only talks about a specific university web network where the vulnerability and attack management is not fully tested and is not fail safe.**	**https:/ /www.scienc edirect.com/ science/articl e/abs/pii/S2214212616301806**

10	**Cyber security analysis using vulnerability assessment and penetration testing** **Shinde, P. S., & Ardhapurkar, S. B. (2016, February)**	**Vulnerability Assessment and Penetration Testing (VAPT) techniques help them to go looking out security loopholes. VAPT helps organizations to determine whether their security arrangements are working properly. This paper aims to elucidate overview and various techniques used in vulnerability assessment and penetration testing (VAPT). Also focuses on implementing cyber security awareness and its importance at various levels of an organization for adoption of required up to date security measures by the organization to stay protected from various cyber-attacks.**	**This paper gives an in-depth knowledge about the various attacks that can be made and the loopholes that can be exploited by hackers. Prevention of attacks was also implemented successfully.** **This paper has covered all aspects with attack and its prevention, however it is not fail proof and fail safe. This can be stated as a drawback of the following research.**	**https://ieeexplore.ieee.org/document/7583912**

Existing System

Based on the literature survey, we gathered from paper [1] about a stack-based algorithm to help prevent directory traversal attacks by safely canonicalizing user-supplied path strings. The algorithm is small, easy to test, cross-platform compatible, and above all, intuitive. From paper [7] we learnt about an existing system where principle, triggering conditions and the exploit process of a VMWare vulnerability is demonstrated. Along with that generalized recommendations are stated that can be applied to achieve secure virtualized implementations.

From paper [10], we learnt about an existing system in place for vulnerability assessment and penetration testing system and from paper [8], an existing detection system of web app attacks was learnt about. This system uses a web server log file dataset for the detection of web application attacks. This system analyzes normal and attack behaviors from web server log and then classifies attack types which are included in the dataset. The attack types include SQL injection, directory traversal etc.

For the Rubik's Cube Algorithm, we learnt from paper [2], that the Rubik's cube method and bit-level encryption principle are combined to realize the image scrambling operation in three-dimensional space using the permutation process. From paper [6], we were presented with the implementation of a communication system between multiple mobile devices with imaging sensors, that encrypt the images using Rubik cube encryption algorithm, and a server.

Gap Analysis

In paper [1], the algorithm could prove to be simple & ineffective for more complex attacks with more powerful computers whereas in paper [2], the 3-D bit level encryption proves to be costly and has high time complexity. For paper [3], the technique is not tested for very high traffic & could prove to be ineffective against severe DDoS attacks. In paper [6], implementing Rubik's cube encryption, the decryption module was not a primary focus and a web app implementation has not been stated to test it out.

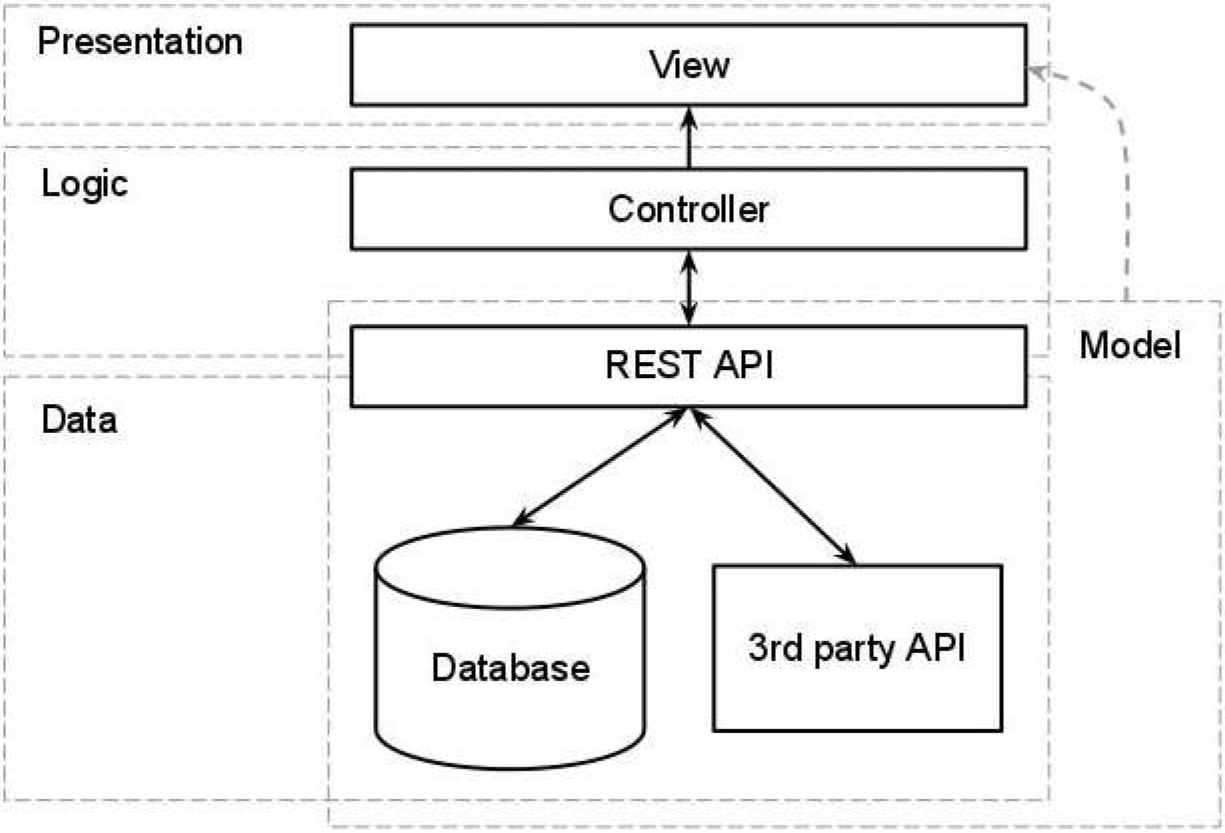
In paper [7], where directory traversal analysis and vulnerability exploitation has been done for VMWare, there has been no discussion of any specific countermeasure for a real environment. Moreover it has been performed on a virtual machine hence we cannot apply prevention methods given in a web app.

Paper [10] has covered most aspects with attack and its prevention, however it is not fail proof and fail safe. This can be stated as a drawback of the following research.

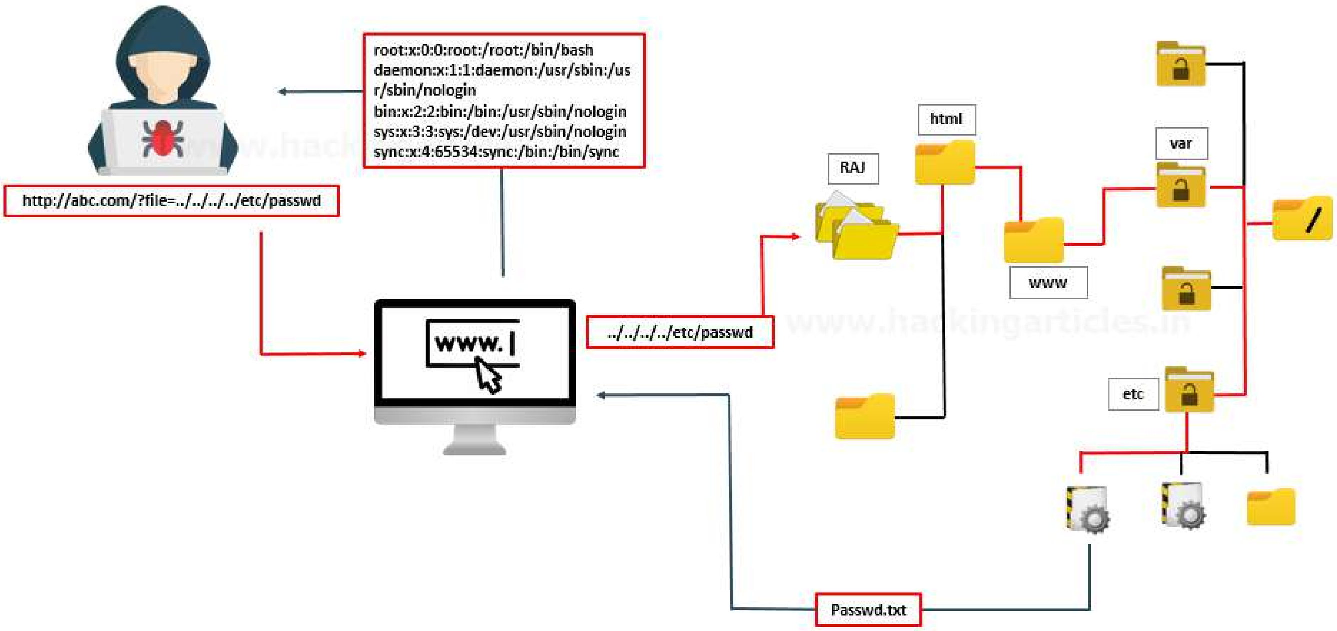
We have used the PHP server in XAMPP environment to exploit all possible directory traversal vulnerabilities that could arise and analysed the consequences of those vulnerabilities. For each of the vulnerabilities assessed, we demonstrated a security measure for mitigation that is fast and effective.

Proposed System

Architecture Diagram



The php website architecture. The vulnerability can expose the files present at these levels to the attacker



```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/us
r/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
```

The directory traversal attack architecture. The attacker can be seen accessing the various files present inside the various folders and directories of the website deployed on the internet**.**

Methodology (Detailed explanation)

For directory traversal analysis, we have created a website, using the PHP server in XAMPP environment. Inside the working directory of the website, we have created directories named, issue, etc and passwd which contain confidential information.

Using a virtual machine, we setup a Kali Linux environment and use dotdotpwn to analyse the vulnerabilities. DotDotPwn is a very flexible intelligent fuzzer to discover traversal directory vulnerabilities in software such as HTTP/FTP/TFTP servers, Web platforms such as CMSs, ERPs, Blogs, etc. In the command prompt of the main operating system, we get the IP address of the website we created and use dotdotpwn in Kali Linux to check if there are any directories in the website that are vulnerable to directory traversal. Since the main OS is Windows, we use the dotdotpwn -o windows command to access the windows boot files. All the vulnerable directories such as etc, passwd, issue are shown as vulnerable by dotdotpwn.

We also demonstrate the Brute force method for directory traversal where, the url of the website is modified by trial and error to look for confidential directories and their information. We even demonstrate how manually removing a file name from the website url could list all existing directories of the website.

With certain dotdotpwn commands such as -f “controllers” and -f “controllers” -d <number>, we can view custom folders and specify the depth of directory traversal respectively.

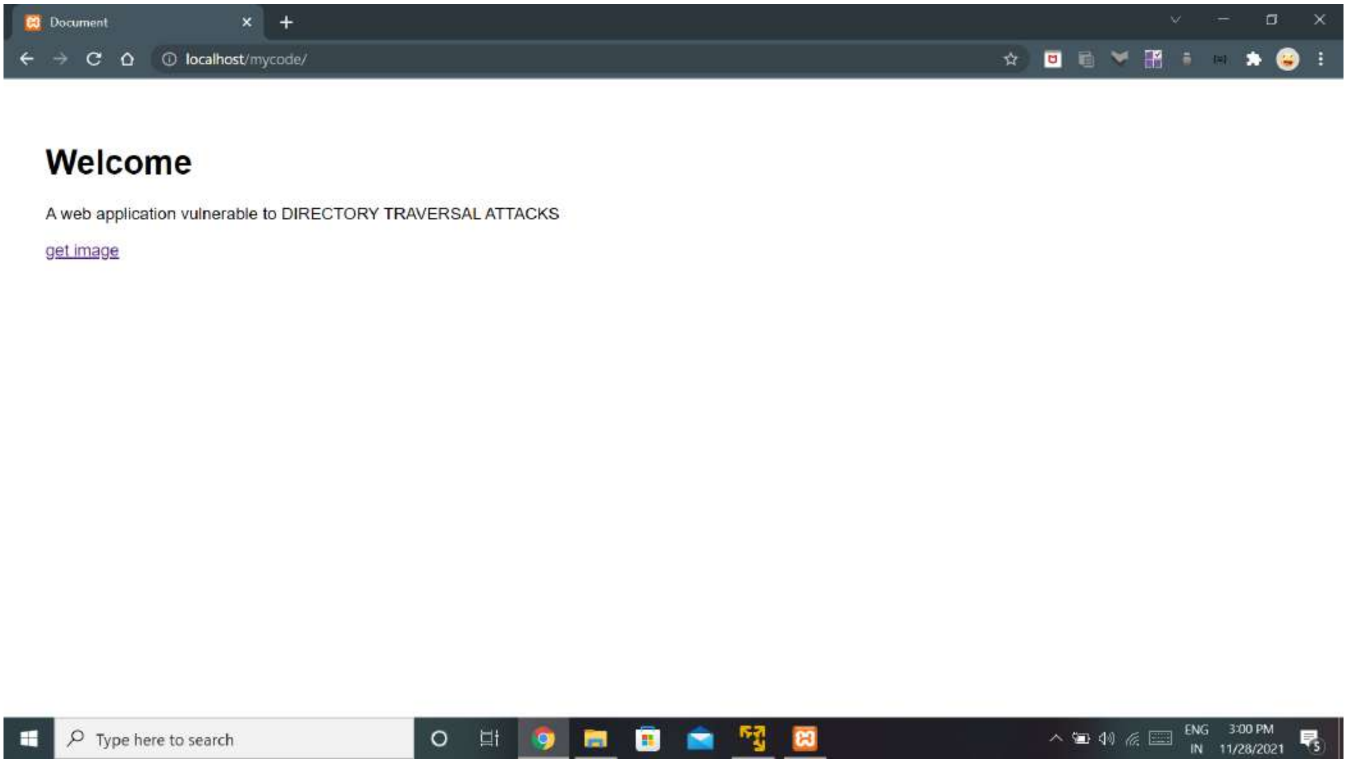
For the mitigation of the vulnerabilities detected, we first propose the technique of storing all Windows/OS based config/admin/boot files in a directory outside htdocs which is specific for PHP servers, since the directories outside htdocs cannot be accessed, no matter the length of the traversal.

Next, we demonstrate a mitigation technique where, inside the working directory of the website, we create a file named htaccess and write “Options -Indexes” inside it. This causes the dotdotpwn to show all directories as 403 which were earlier shown as vulnerable. When we give the path to the directory in the website url, it displays “403: Forbidden”.

Lastly, in the working directory of the website, we create a file named “index.php” where we write an algorithm that creates an empty index.php file in each of the directories and sub directories that exist in the working directory of the website. Now, when we give the path for any directory in the url, it displays a blank screen and all confidential information is secure.

Hence we have shown all possible directory traversal vulnerabilities arising in our website and also shown ways to mitigate and demonstrated the mitigation techniques for each of the vulnerabilities.

Screenshots

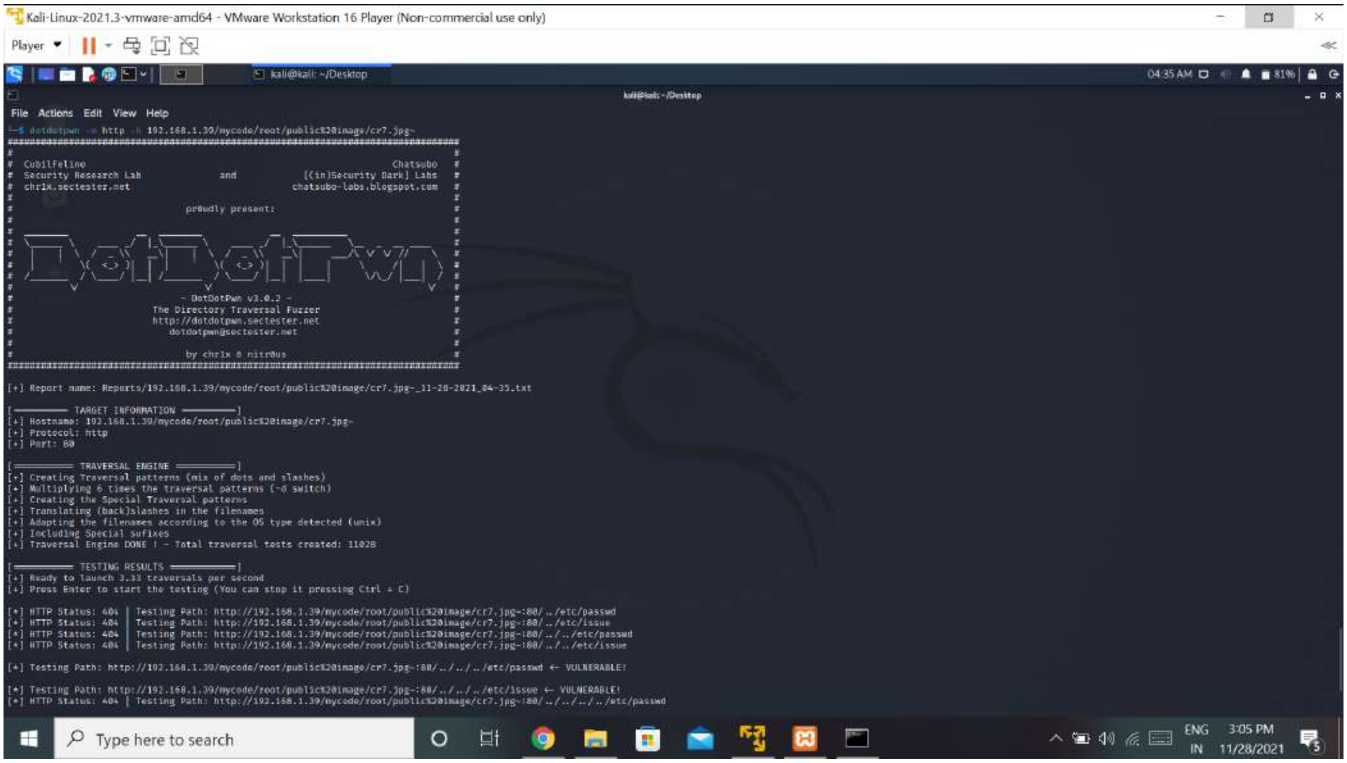


Welcome

A web application vulnerable to DIRECTORY TRAVERSAL ATTACKS

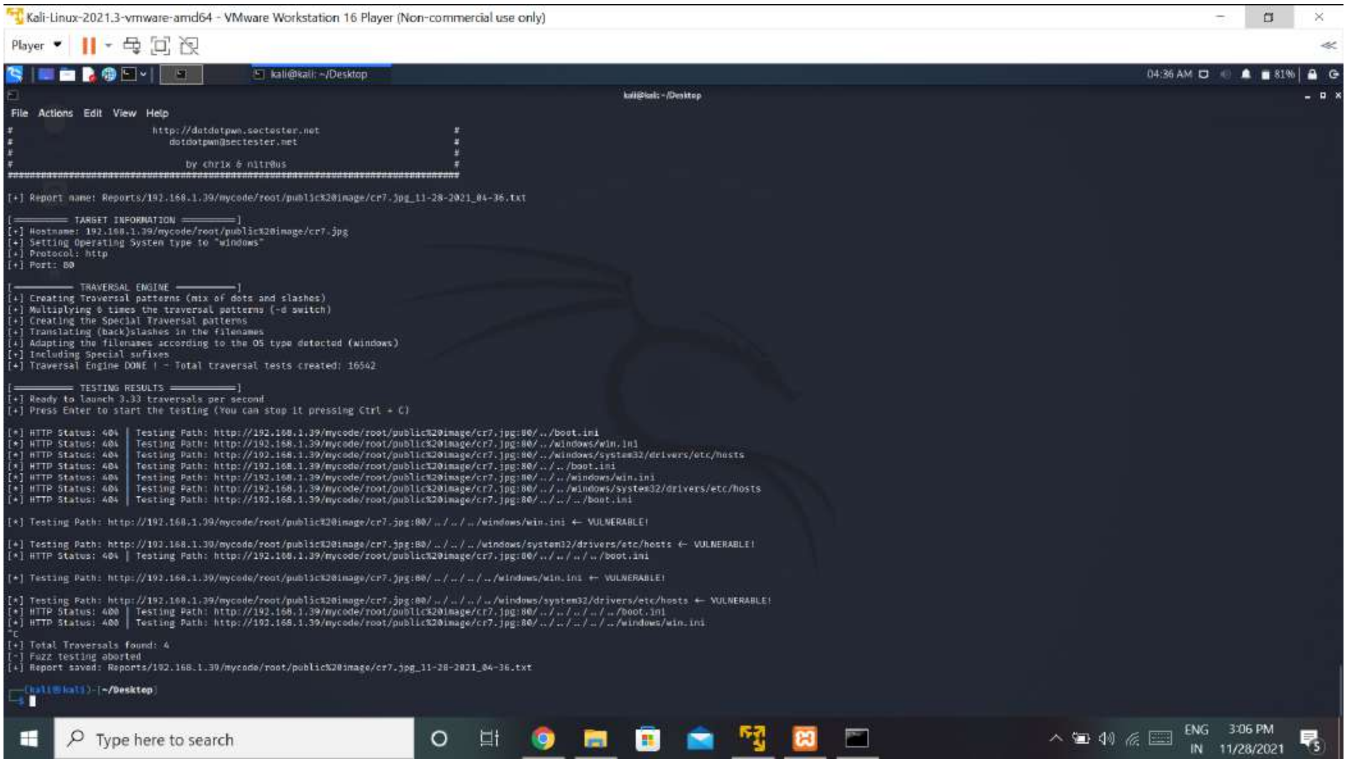
get image

The created website with vulnerabilities present in it.


Kali-Linux-2021.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
kali@kali: ~/Desktop
04:35 AM
81%
File Actions Edit View Help
Type here to search
ENG
IN
3:05 PM
11/28/2021

```
└─$ dotdotpwn -m http -h 192.168.1.39/mycode/root/public%20image/cr7.jpg~
#################################################################################
#                                                                               #
#  CubilFelino                                                      Chatsubo    #
#  Security Research Lab              and          [(in)Security Dark] Labs    #
#  chr1x.sectester.net                             chatsubo-labs.blogspot.com  #
#                                                                               #
#                            pr0udly present:                                   #
#                                                                               #
#                           - DotDotPwn v3.0.2 -                                #
#                      The Directory Traversal Fuzzer                           #
#                      http://dotdotpwn.sectester.net                           #
#                         dotdotpwn@sectester.net                               #
#                                                                               #
#                            by chr1x & nitr0us                                 #
#################################################################################

[+] Report name: Reports/192.168.1.39/mycode/root/public%20image/cr7.jpg~_11-28-2021_04-35.txt

[======== TARGET INFORMATION ========]
[+] Hostname: 192.168.1.39/mycode/root/public%20image/cr7.jpg~
[+] Protocol: http
[+] Port: 80

[========= TRAVERSAL ENGINE =========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (unix)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 11028

[========= TESTING RESULTS =========]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../etc/passwd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../etc/issue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../etc/passwd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../etc/issue

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../../etc/passwd <- VULNERABLE!

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../../etc/issue <- VULNERABLE!
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../../../etc/passwd
```

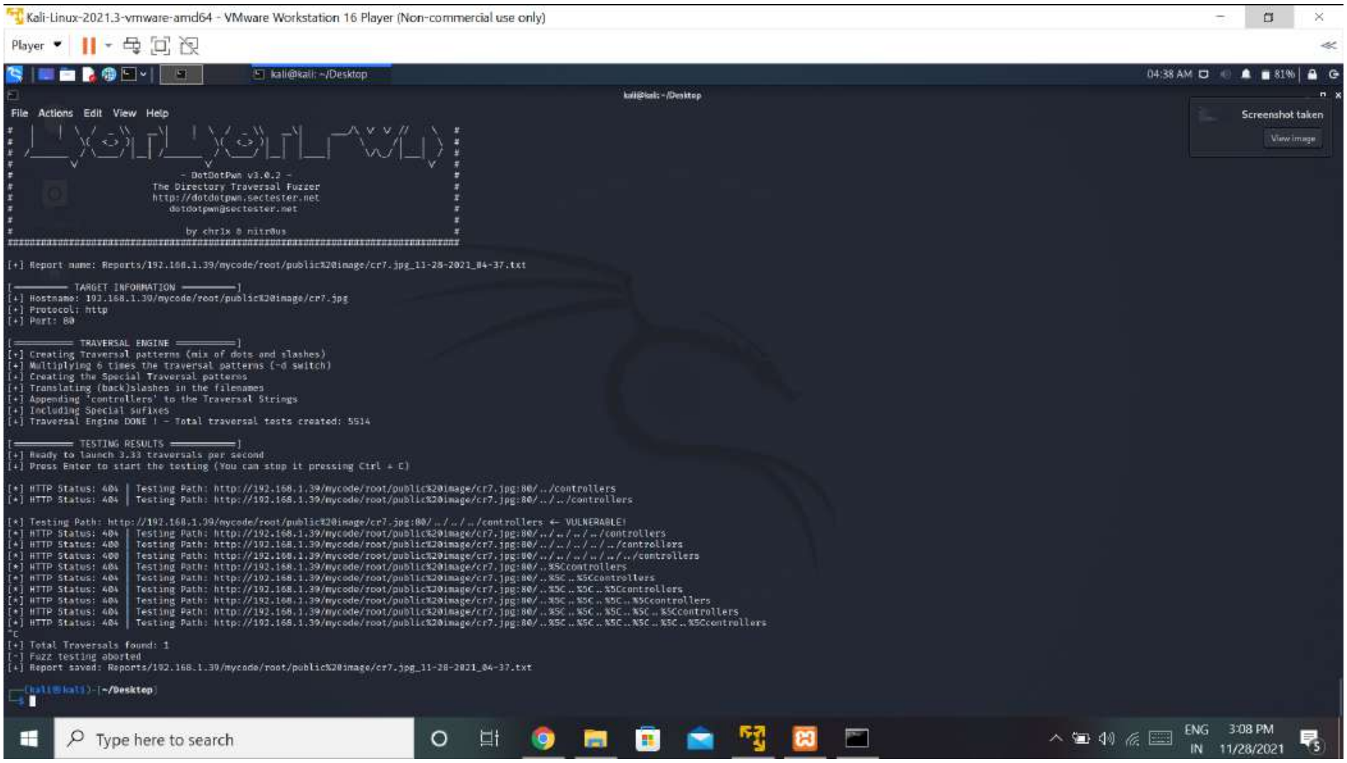
Using dotdotpwn tool on kali linux to attack localhost and check for directory traversal vulnerability


Kali-Linux-2021.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
kali@kali: ~/Desktop
04:36 AM
81%
File Actions Edit View Help
Type here to search
ENG
IN
3:06 PM
11/28/2021

```
#                      http://dotdotpwn.sectester.net                           #
#                         dotdotpwn@sectester.net                               #
#                                                                               #
#                            by chr1x & nitr0us                                 #
#################################################################################

[+] Report name: Reports/192.168.1.39/mycode/root/public%20image/cr7.jpg_11-28-2021_04-36.txt

[======== TARGET INFORMATION ========]
[+] Hostname: 192.168.1.39/mycode/root/public%20image/cr7.jpg
[+] Setting Operating System type to "windows"
[+] Protocol: http
[+] Port: 80

[========= TRAVERSAL ENGINE =========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (windows)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 16542

[========= TESTING RESULTS =========]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../boot.ini
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../windows/win.ini
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../windows/system32/drivers/etc/hosts
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../boot.ini
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../windows/win.ini
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../windows/system32/drivers/etc/hosts
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../boot.ini

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../windows/win.ini <- VULNERABLE!

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../windows/system32/drivers/etc/hosts <- VULNERABLE!
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../boot.ini

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../windows/win.ini <- VULNERABLE!

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../windows/system32/drivers/etc/hosts <- VULNERABLE!
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../../boot.ini
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../../windows/win.ini
^C
[+] Total Traversals found: 4
[-] Fuzz testing aborted
[+] Report saved: Reports/192.168.1.39/mycode/root/public%20image/cr7.jpg_11-28-2021_04-36.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

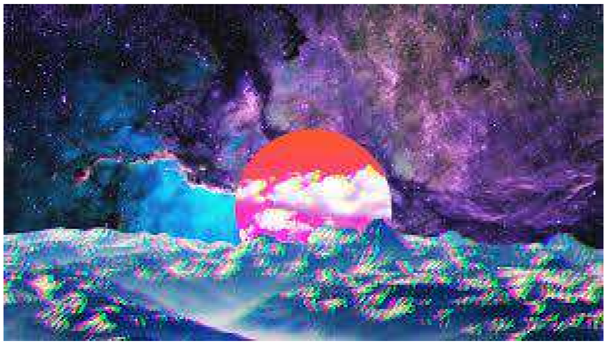
Using dotdotpwn to get windows boot files and other sensitive windows files.

Vulnerability found!


Kali-Linux-2021.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
kali@kali: ~/Desktop
04:38 AM
81%
File Actions Edit View Help
Screenshot taken
View image
Type here to search
ENG
IN
3:08 PM
11/28/2021

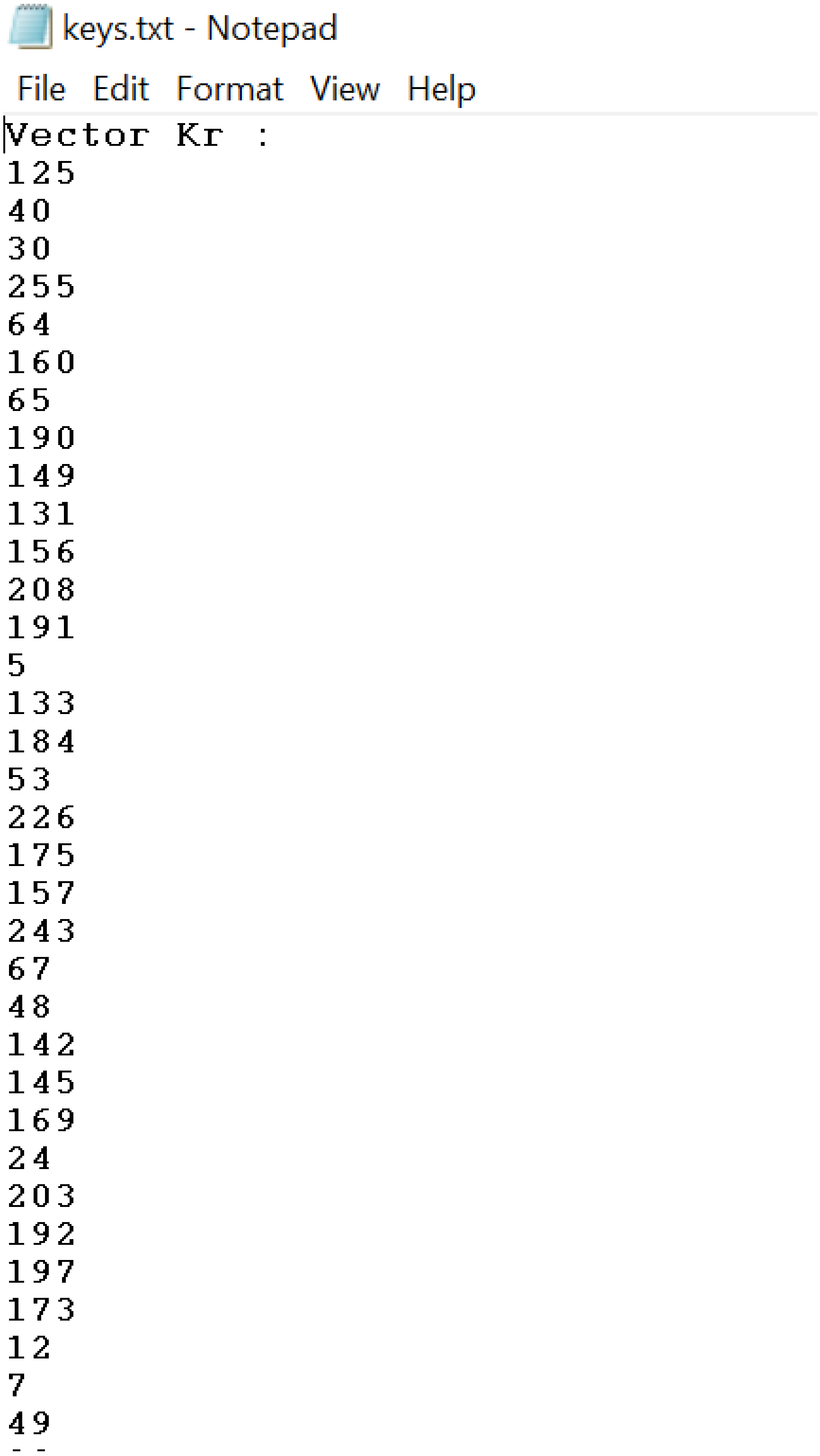
```
#                                                                         #
#                      - DotDotPwn v3.0.2 -                               #
#                  The Directory Traversal Fuzzer                         #
#                  http://dotdotpwn.sectester.net                         #
#                     dotdotpwn@sectester.net                             #
#                                                                         #
#                       by chr1x & nitr0us                                #
###########################################################################

[+] Report name: Reports/192.168.1.39/mycode/root/public%20image/cr7.jpg_11-28-2021_04-37.txt

[========== TARGET INFORMATION ==========]
[+] Hostname: 192.168.1.39/mycode/root/public%20image/cr7.jpg
[+] Protocol: http
[+] Port: 80

[=========== TRAVERSAL ENGINE ===========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Appending 'controllers' to the Traversal Strings
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 5514

[=========== TESTING RESULTS ============]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../controllers
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../controllers

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../controllers <- VULNERABLE!
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../controllers
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../../controllers
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../../../controllers
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/..%5Ccontrollers
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/..%5C..%5Ccontrollers
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/..%5C..%5C..%5Ccontrollers
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5Ccontrollers
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5C..%5Ccontrollers
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5C..%5C..%5Ccontrollers
^C
[+] Total Traversals found: 1
[-] Fuzz testing aborted
[+] Report saved: Reports/192.168.1.39/mycode/root/public%20image/cr7.jpg_11-28-2021_04-37.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

Using dotdotpwn to target the controllers folder and attack the directory with controller files.



Original Image stored. Even if the directory traversal protection system is breached, the additional encryption mechanism can save the file from being read.

Protecting the original image by encryption using the Rubik Cube Algorithm.



```
Vector Kr :
125
40
30
255
64
160
65
190
149
131
156
208
191
5
133
184
53
226
175
157
243
67
48
142
145
169
24
203
192
197
173
12
7
49
```

Keys generated for decrypting the encrypted image.

Conclusion

This project work can successfully identify and detect the possibility if a directory traversal vulnerability in a php based website using the tool dotdotpwn on kalilinux. After detection, the proposed system can also mitigate the attack and create a secure channel free from directory traversal vulnerability. If there is a still an attack which can retrieve sensitive data, the project can successfully follow the encryption algorithm to protect the file. Lastly, the project can be extended and the encryption process can be automated. Also , the system can be upgraded to make the website viewership secure by adding user roles and access constraints.

References

[1]Michael Flanders, A Simple and Intuitive Algorithm for Preventing Directory Traversal Attacks (2019), Cornell Univ.

[2]Hegui Zhu, Lewen Dai, Yating Liu, Lijun Wu, A three-dimensional bit-level image encryption algorithm with Rubik's cube method, Mathematics and Computers in Simulation (2021)

[3]N. Muraleedharan, Anna Thomas, A Traffic Monitoring and Policy Enforcement Framework for HTTP (2020)

[4]Zainab S. Alwan1, Manal F. Younis2, Detection and Prevention of SQL Injection Attack: A Survey (2017)

[5]Matthew Denis, Carlos Zena, Penetration testing: Concepts, attack methods, and defense strategies (2016)

[6] Ionescu, V. M., & Diaconu, A. V., Rubik's cube principle based image encryption algorithm implementation on mobile devices. (2015).

[7] Bai, Y., & Chen, Z., Analysis and Exploit of Directory Traversal Vulnerability on VMware, (2015).

[8]Han, E. E., Detection of Web Application Attacks with Request Length Module and Regex Pattern Analysis (2015)

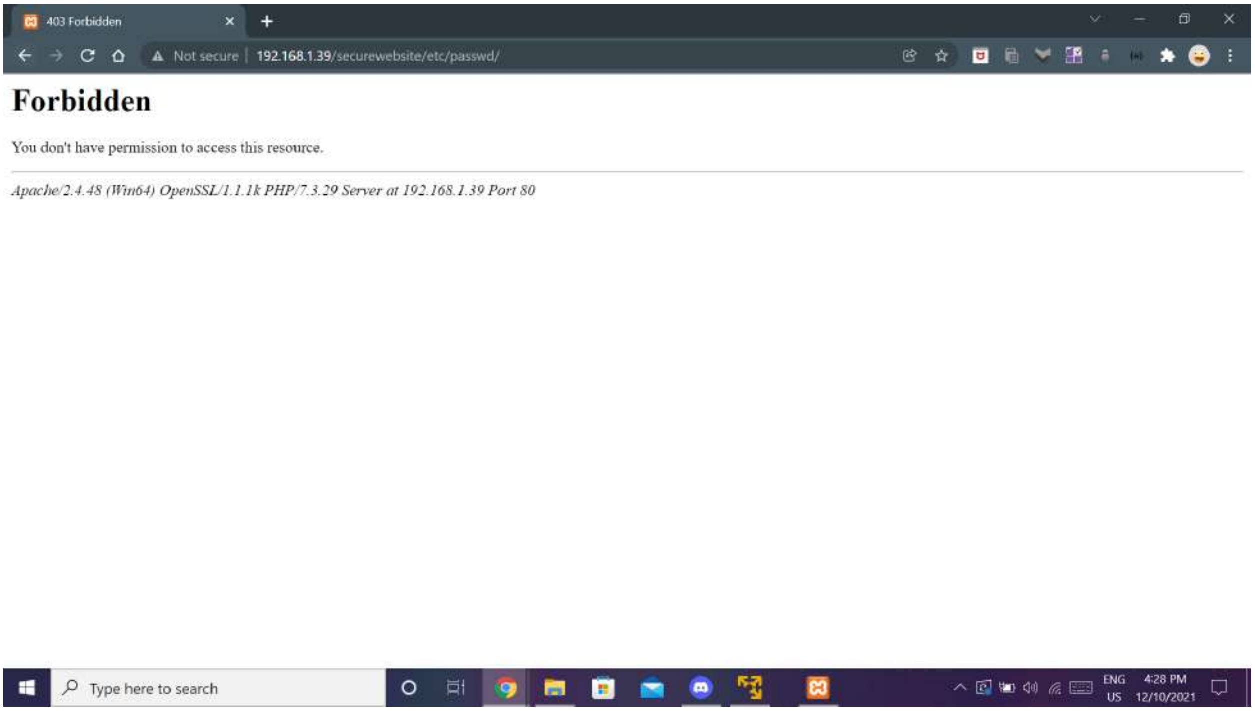
[9]Joshi, C., & Singh, U. K., Information security risks management framework – A step towards mitigating security risks in university network (2017)

[10]Shinde, P. S., & Ardhapurkar, S. B., Cyber security analysis using vulnerability assessment and penetration testing, (2016)

Audit Report

1.Risk Assessment :

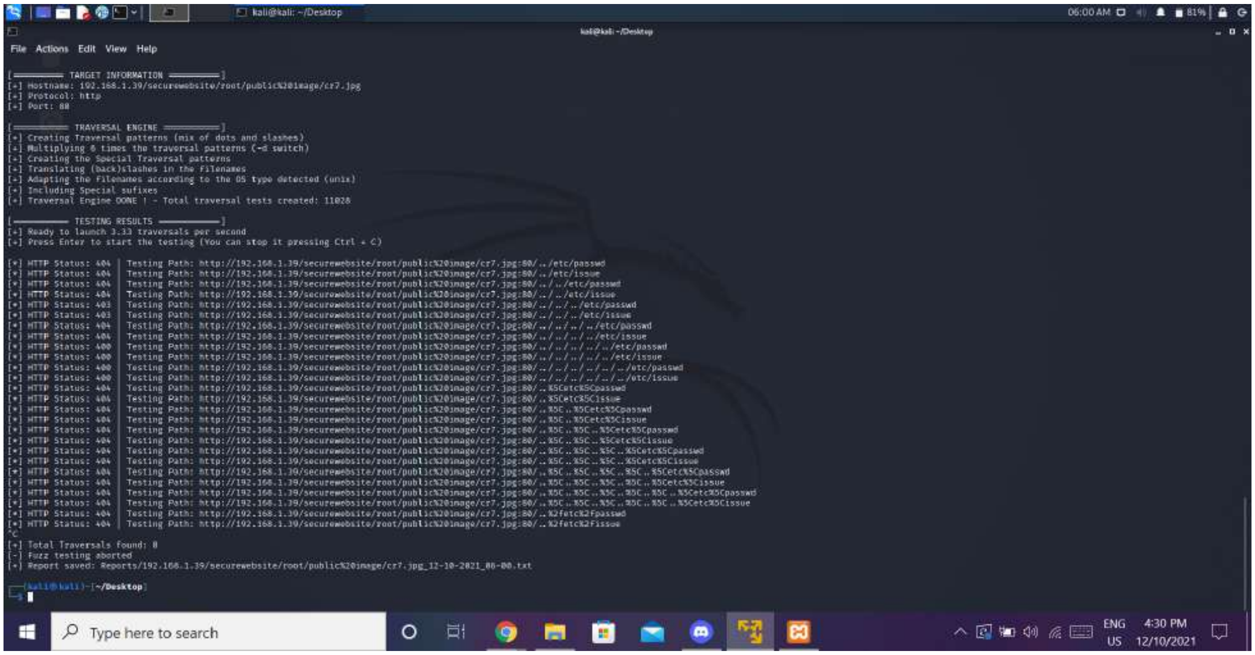
Low Risk
All the vulnerabilities related to directory traversal were mitigated and the website developed is safe to use.



Forbidden

You don't have permission to access this resource.

Apache/2.4.48 (Win64) OpenSSL/1.1.1k PHP/7.3.29 Server at 192.168.1.39 Port 80



kali@kali: ~/Desktop

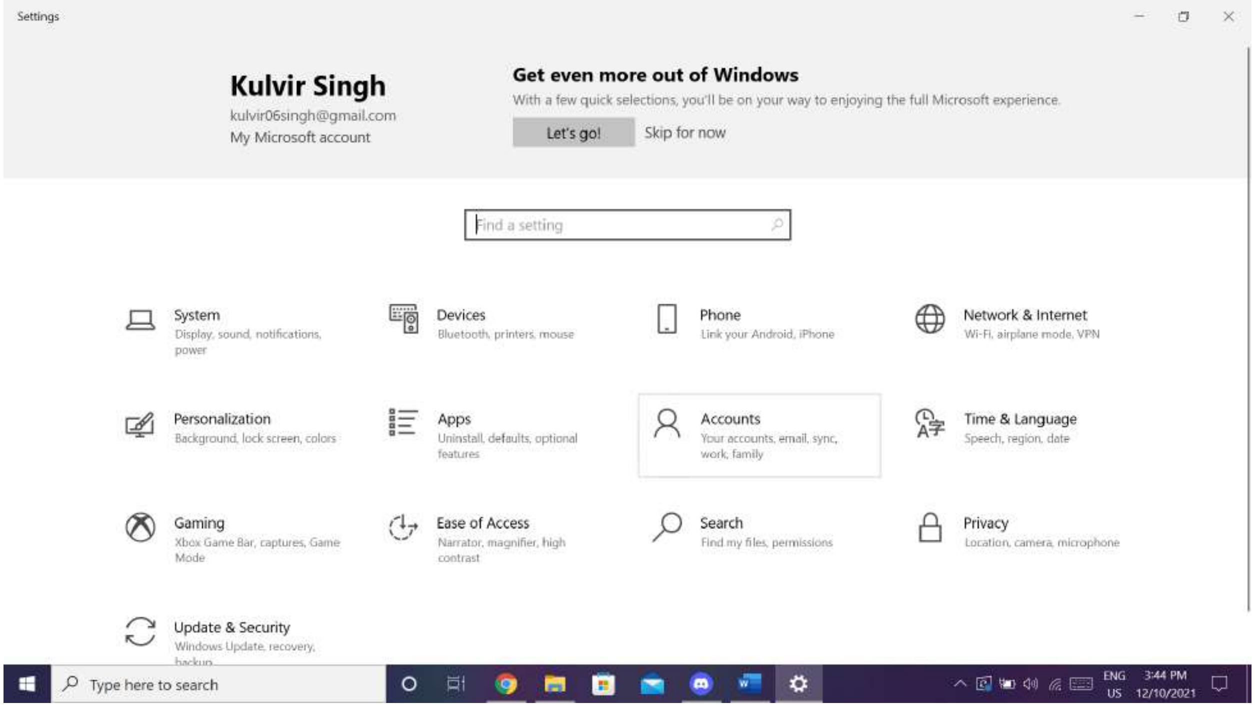
06:00 AM 81%

File Actions Edit View Help

```
[========= TARGET INFORMATION =========]
[+] Hostname: 192.168.1.39/securewebsite/root/public%20image/cr7.jpg
[+] Protocol: http
[+] Port: 80

[========== TRAVERSAL ENGINE ==========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (unix)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 11028

[---------- TESTING RESULTS ----------]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../etc/passwd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../etc/issue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../etc/passwd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../etc/issue
[*] HTTP Status: 403 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../../etc/passwd
[*] HTTP Status: 403 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../../etc/issue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../../../etc/passwd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/../../../../../../etc/issue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5Cetc%5Cissue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5Cetc%5Cissue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5C..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5C..%5Cetc%5Cissue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5Cetc%5Cissue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5C..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5C..%5Cetc%5Cissue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5C..%5C..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%5C..%5C..%5C..%5C..%5C..%5Cetc%5Cissue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%2fetc%2Fpasswd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/securewebsite/root/public%20image/cr7.jpg:80/..%2fetc%2Fissue
^C
[+] Total Traversals found: 0
[-] Fuzz testing aborted
[+] Report saved: Reports/192.168.1.39/securewebsite/root/public%20image/cr7.jpg_12-10-2021_06-00.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

2. Backup Procedures :

- Open Settings.
- Click on Update & Security.
- Click on Backup.
- Under the "Looking for an older backup?" section, click the Go to Backup and Restore option.
- Click the Create a system image option from the left pane.
- Select the On a hard disk option.
- Use the "On a hard disk" drop-down menu and select the location to export the Windows 10 full backup.
- Click the Next button.
- (Optional) Select any additional hard drives to include them in the backup.

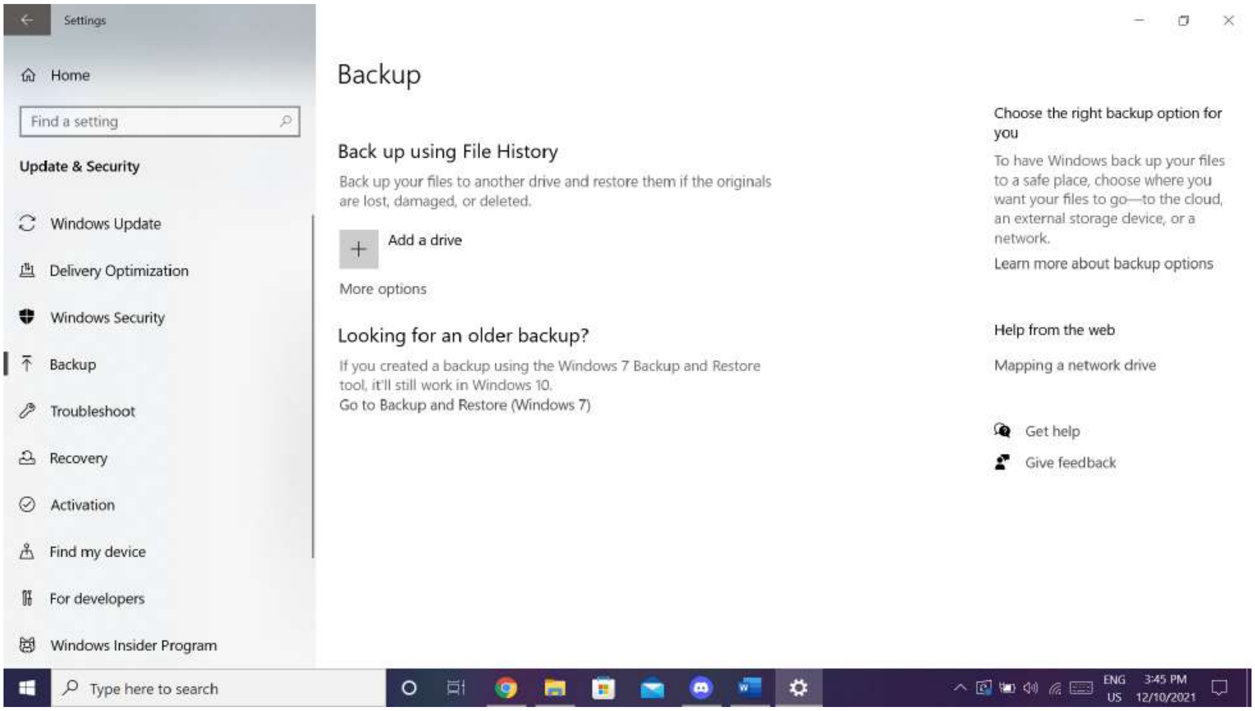


Settings

Kulvir Singh
kulvir06singh@gmail.com
My Microsoft account

Get even more out of Windows
With a few quick selections, you'll be on your way to enjoying the full Microsoft experience.

Let's go!
Skip for now
Find a setting
System
Display, sound, notifications, power
Devices
Bluetooth, printers, mouse
Phone
Link your Android, iPhone
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Apps
Uninstall, defaults, optional features
Accounts
Your accounts, email, sync, work, family
Time & Language
Speech, region, date
Gaming
Xbox Game Bar, captures, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Search
Find my files, permissions
Privacy
Location, camera, microphone
Update & Security
Windows Update, recovery, backup
Type here to search
ENG
US
3:44 PM
12/10/2021



Backup

Back up using File History

Back up your files to another drive and restore them if the originals are lost, damaged, or deleted.

Add a drive

More options

Looking for an older backup?

If you created a backup using the Windows 7 Backup and Restore tool, it'll still work in Windows 10.
Go to Backup and Restore (Windows 7)

Choose the right backup option for you

To have Windows back up your files to a safe place, choose where you want your files to go—to the cloud, an external storage device, or a network.
Learn more about backup options

Help from the web

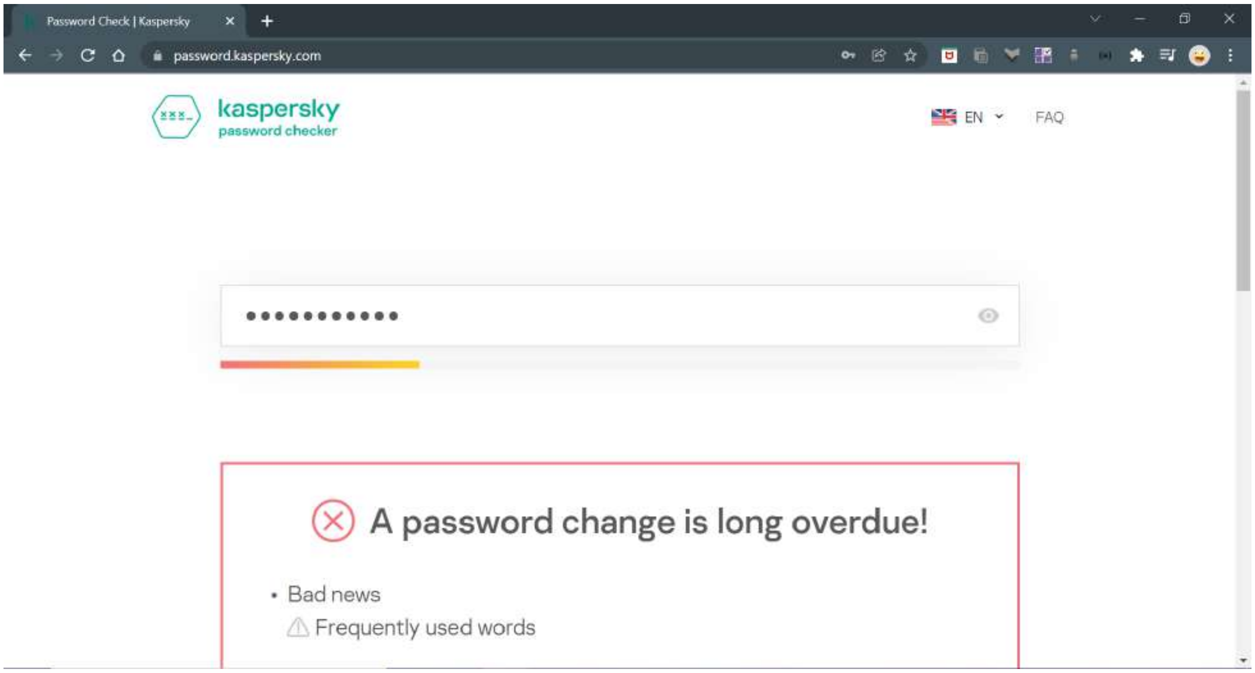
Mapping a network drive

Get help

Give feedback

3. Password Policy

Admin has a password set for root or administrator purposes.
Password Check :

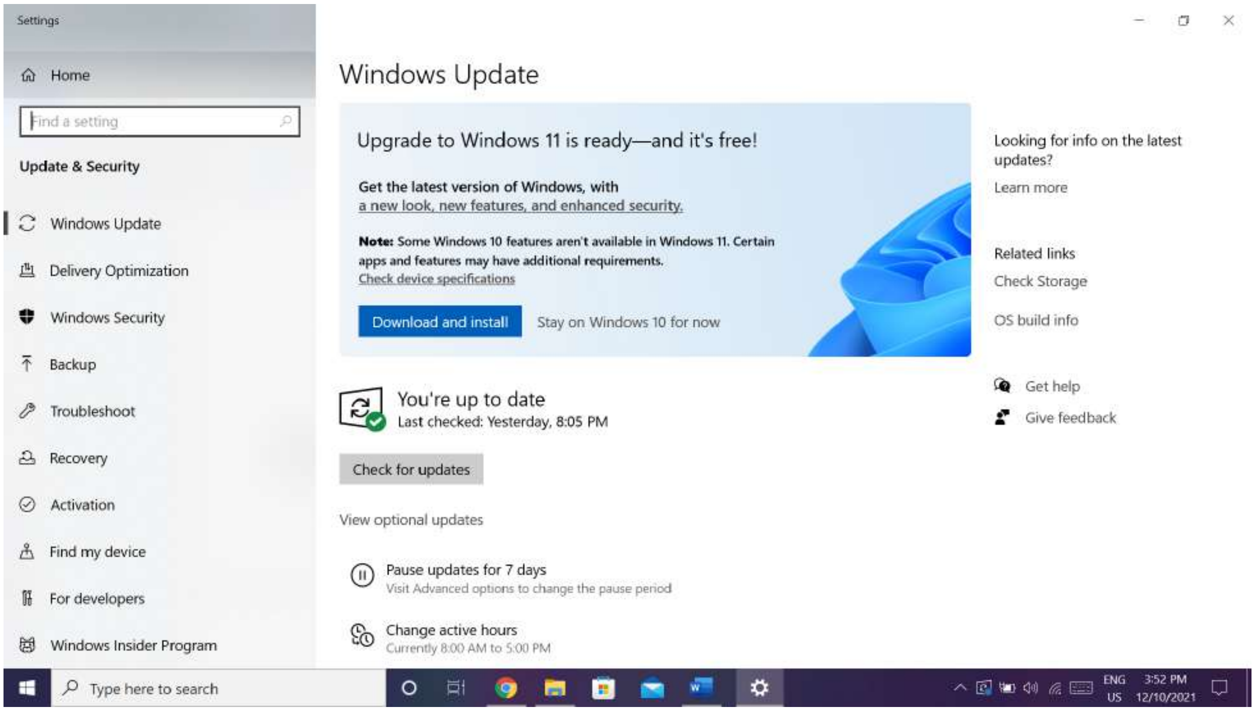


A password change is long overdue!

- Bad news
  ⚠ Frequently used words

4. Change control section :

Update : PC ready to be updated to Windows 11
Time Interval : 2 hours at max to download and install updates
Last Update Time : 11/11/2021
Latest OS version : Windows 10
Safe Zone Software present : No



Windows Update

Upgrade to Windows 11 is ready—and it's free!

Get the latest version of Windows, with a new look, new features, and enhanced security.

Note: Some Windows 10 features aren't available in Windows 11. Certain apps and features may have additional requirements. Check device specifications

You're up to date
Last checked: Yesterday, 8:05 PM

Pause updates for 7 days
Visit Advanced options to change the pause period

Change active hours
Currently 8:00 AM to 5:00 PM

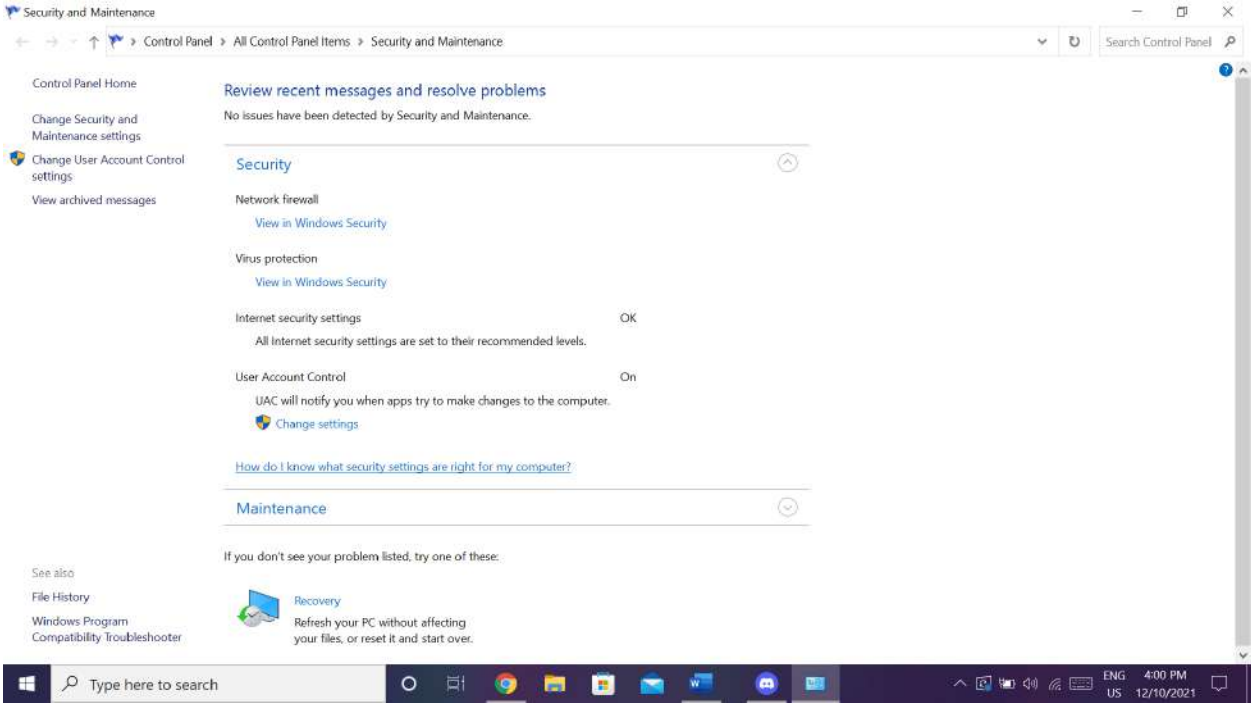
Looking for info on the latest updates?

5. Incident Response Section

As per the guidelines issued by Microsoft Windows 10 system based incident response process, we can follow the following methodology.
Incident response is the practice of investigating and remediating active attack campaigns on your organization. This is part of the security operations discipline and is primarily reactive in nature. Incident response has the largest direct influence on the overall mean time to acknowledge (MTTA) and mean time to remediate (MTTR) that measure how well security operations are able to reduce organizational risk. Incident response teams heavily rely on good working relationships between threat hunting, intelligence, and incident management teams (if present) to actually reduce risk. See SecOps metrics for more information.

6. Security Posture

Security posture refers to an organization's overall state of cybersecurity readiness. An enterprise's security posture takes into account: Visibility into the security status of software and hardware assets, networks, services, and information.



Security and Maintenance
Control Panel > All Control Panel Items > Security and Maintenance
Search Control Panel
Control Panel Home
Change Security and Maintenance settings
Change User Account Control settings
View archived messages
See also
File History
Windows Program Compatibility Troubleshooter
Type here to search
ENG US 4:00 PM 12/10/2021

Review recent messages and resolve problems

No issues have been detected by Security and Maintenance.

Security

Network firewall
View in Windows Security

Virus protection
View in Windows Security

Internet security settings — OK
All Internet security settings are set to their recommended levels.

User Account Control — On
UAC will notify you when apps try to make changes to the computer.
Change settings

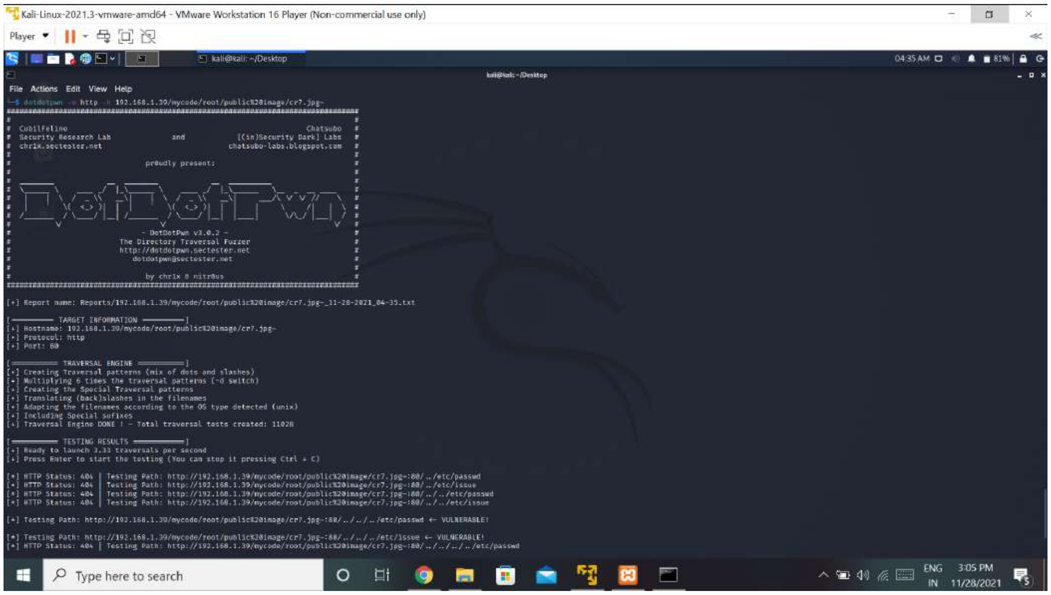
How do I know what security settings are right for my computer?

Maintenance

If you don't see your problem listed, try one of these:

Recovery
Refresh your PC without affecting your files, or reset it and start over.

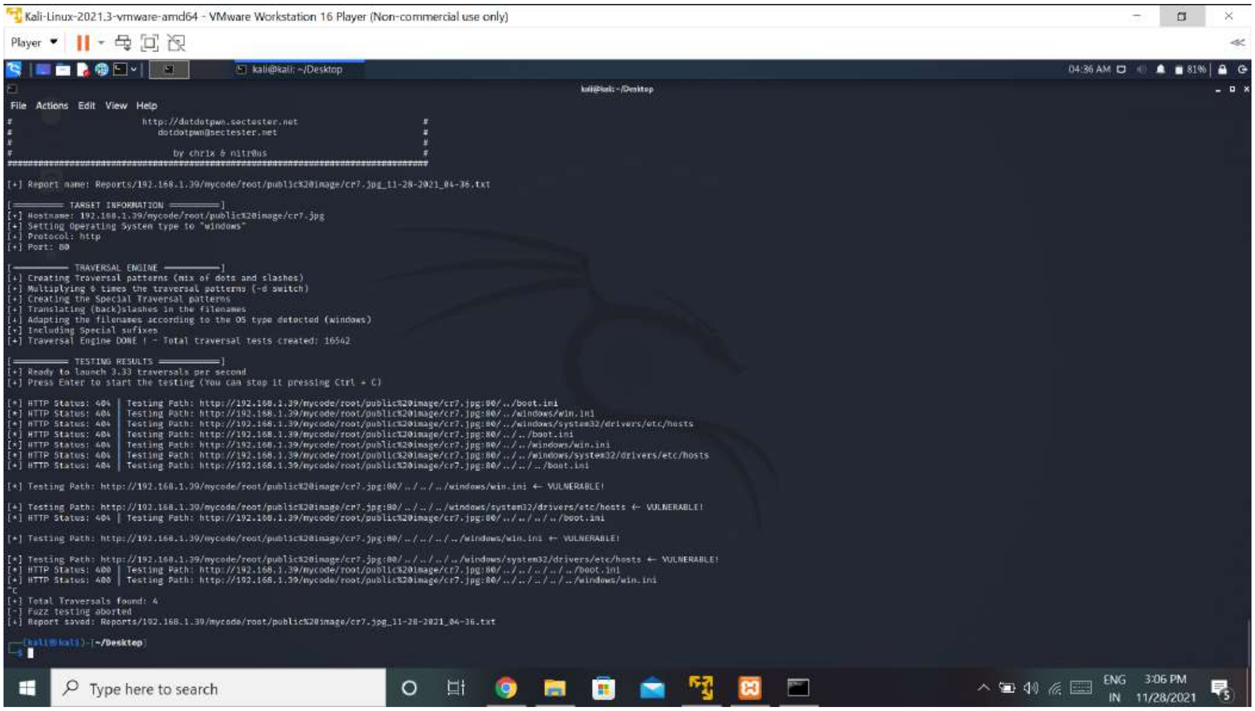
7. Report



Kali-Linux-2021.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
kali@kali: ~/Desktop
File Actions Edit View Help
Type here to search
ENG IN 3:05 PM 11/28/2021

```
└─$ dotdotpwn -m http -h 192.168.1.39/mycode/root/public%20image/cr7.jpg~
#################################################################################
#                                                                               #
#  CubilFelino                                                        Chatsubo  #
#  Security Research Lab              and          [(in)Security Dark] Labs    #
#  chr1x.sectester.net                             chatsubo-labs.blogspot.com  #
#                                                                               #
#                               pr0udly present:                                #
#                                                                               #
#                              - DotDotPwn v3.0.2 -                             #
#                          The Directory Traversal Fuzzer                       #
#                          http://dotdotpwn.sectester.net                       #
#                            dotdotpwn@sectester.net                            #
#                                                                               #
#                               by chr1x & nitr0us                              #
#################################################################################

[+] Report name: Reports/192.168.1.39/mycode/root/public%20image/cr7.jpg~_11-28-2021_04-35.txt

[========== TARGET INFORMATION ==========]
[+] Hostname: 192.168.1.39/mycode/root/public%20image/cr7.jpg~
[+] Protocol: http
[+] Port: 80

[=========== TRAVERSAL ENGINE ===========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (unix)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 11028

[=========== TESTING RESULTS ===========]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../etc/passwd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../etc/issue
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../etc/passwd
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../etc/issue

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../../etc/passwd <- VULNERABLE!

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../../etc/issue <- VULNERABLE!
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg~:80/../../../../etc/passwd
```

Using dotdotpwn tool on kali linux to attack localhost and check for directory traversal vulnerability

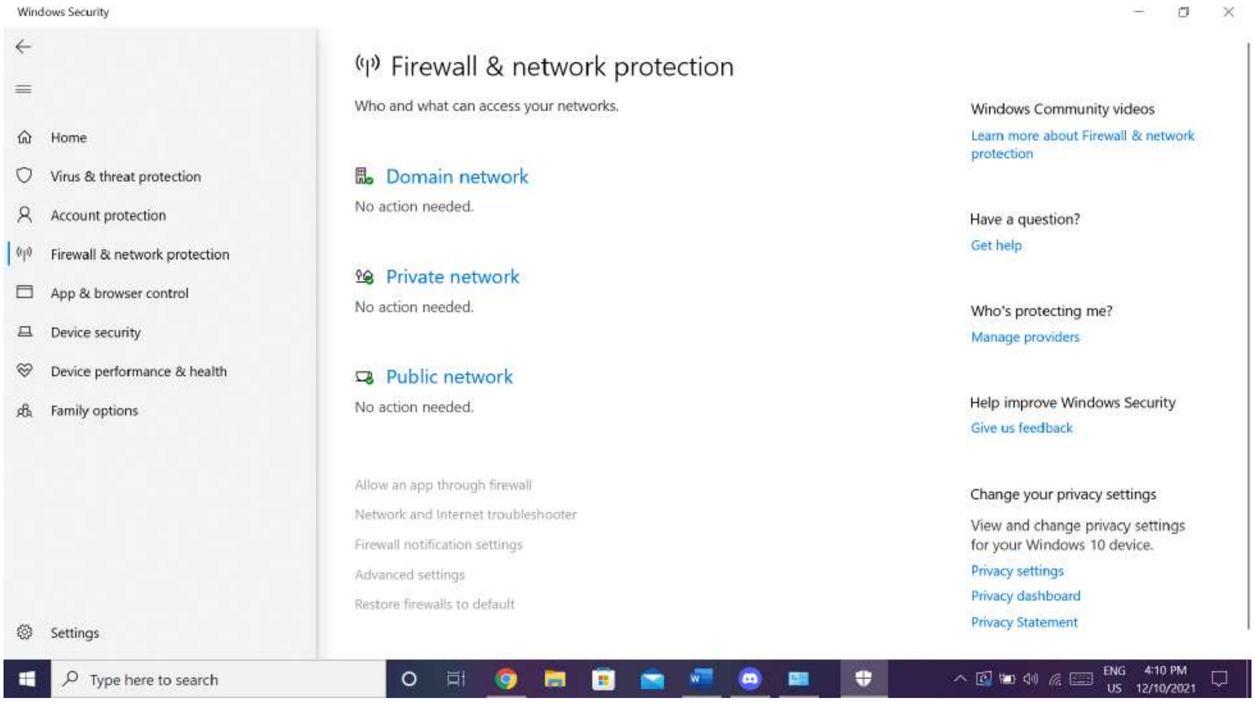

Kali-Linux-2021.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
kali@kali: ~/Desktop
File Actions Edit View Help
04:36 AM
81%
Type here to search
ENG
IN
3:06 PM
11/28/2021

```
#                         http://dotdotpwn.sectester.net                       #
#                            dotdotpwn@sectester.net                           #
#                                                                              #
#                               by chr1x & nitr0us                             #
################################################################################

[+] Report name: Reports/192.168.1.39/mycode/root/public%20image/cr7.jpg_11-28-2021_04-36.txt

[========== TARGET INFORMATION ==========]
[+] Hostname: 192.168.1.39/mycode/root/public%20image/cr7.jpg
[+] Setting Operating System type to "windows"
[+] Protocol: http
[+] Port: 80

[=========== TRAVERSAL ENGINE ===========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (windows)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 16542

[=========== TESTING RESULTS ===========]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../boot.ini
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../windows/win.ini
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../windows/system32/drivers/etc/hosts
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../boot.ini
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../windows/win.ini
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../windows/system32/drivers/etc/hosts
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../boot.ini

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../windows/win.ini <- VULNERABLE!

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../windows/system32/drivers/etc/hosts <- VULNERABLE!
[*] HTTP Status: 404 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../boot.ini

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../windows/win.ini <- VULNERABLE!

[*] Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../windows/system32/drivers/etc/hosts <- VULNERABLE!
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../../boot.ini
[*] HTTP Status: 400 | Testing Path: http://192.168.1.39/mycode/root/public%20image/cr7.jpg:80/../../../../../windows/win.ini
^C
[+] Total Traversals found: 4
[-] Fuzz testing aborted
[+] Report saved: Reports/192.168.1.39/mycode/root/public%20image/cr7.jpg_11-28-2021_04-36.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

Using dotdotpwn to get windows boot files and other sensitive windows files.

a) Network Security Audit

Firewall : Enabled



Firewall & network protection

Who and what can access your networks.

Domain network

No action needed.

Private network

No action needed.

Public network

No action needed.

Allow an app through firewall

Network and Internet troubleshooter

Firewall notification settings

Advanced settings

Restore firewalls to default

Windows Community videos

Learn more about Firewall & network protection

Have a question?

Get help

Who's protecting me?

Manage providers

Help improve Windows Security

Give us feedback

Change your privacy settings

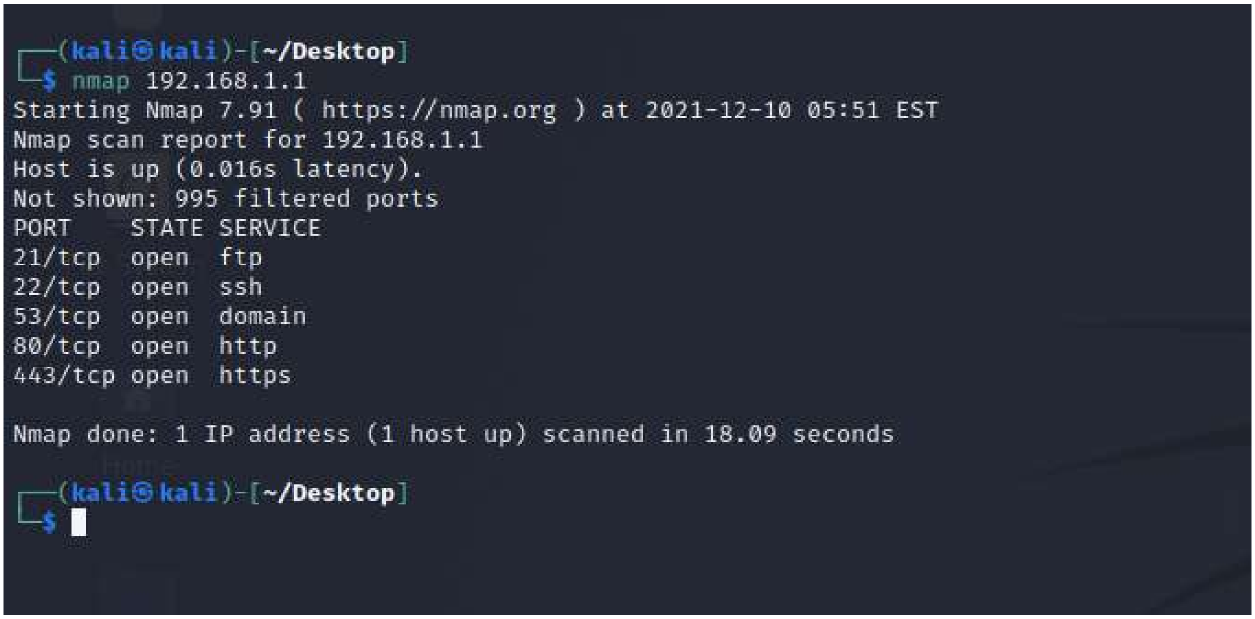
View and change privacy settings for your Windows 10 device.

Privacy settings

Privacy dashboard

Privacy Statement

Nmap on network IP :



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap 192.168.1.1
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 05:51 EST
Nmap scan report for 192.168.1.1
Host is up (0.016s latency).
Not shown: 995 filtered ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 18.09 seconds

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

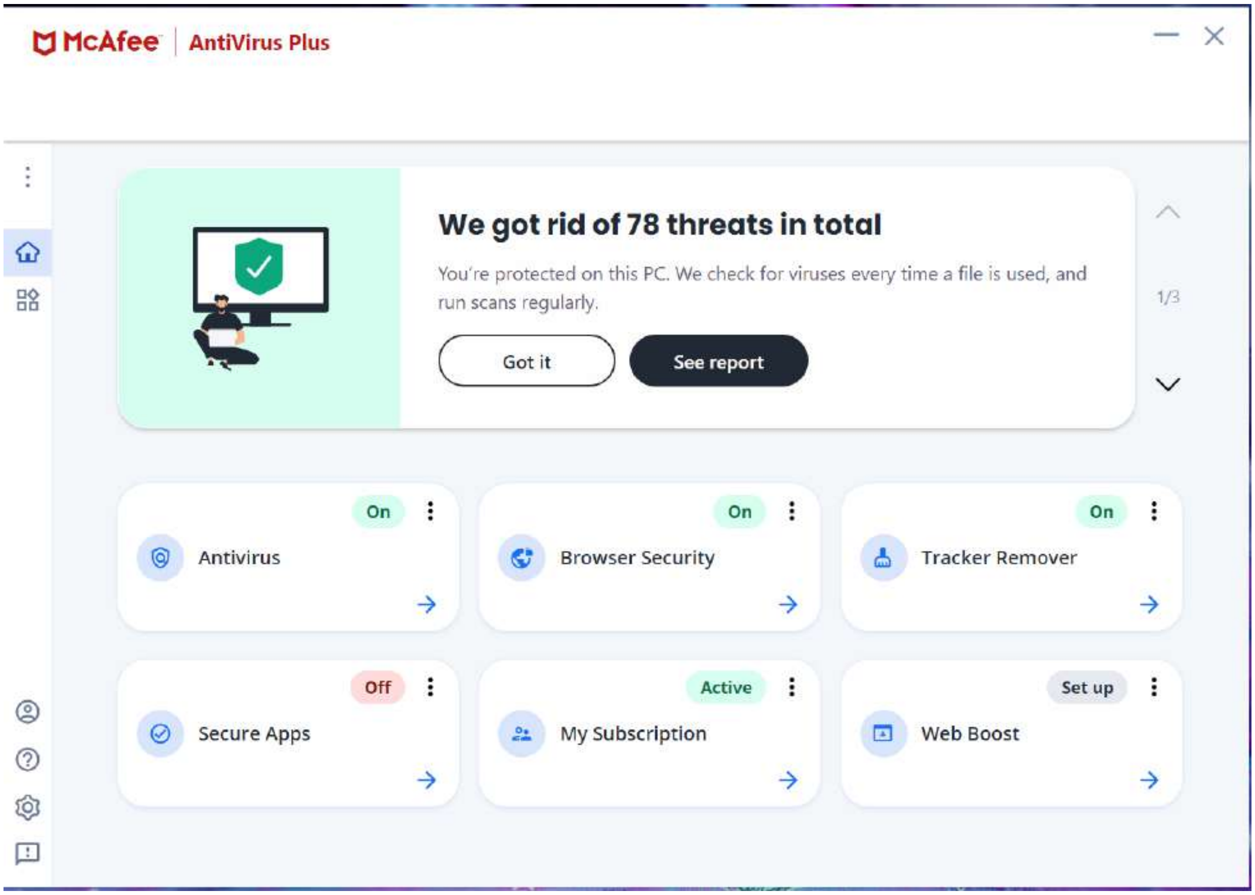
b) Cyber Security Audit

Antivirus →

Name : McAfee

Version : 16.0

Status :



We got rid of 78 threats in total

You're protected on this PC. We check for viruses every time a file is used, and run scans regularly.

Antivirus On

Browser Security On

Tracker Remover On

Secure Apps Off

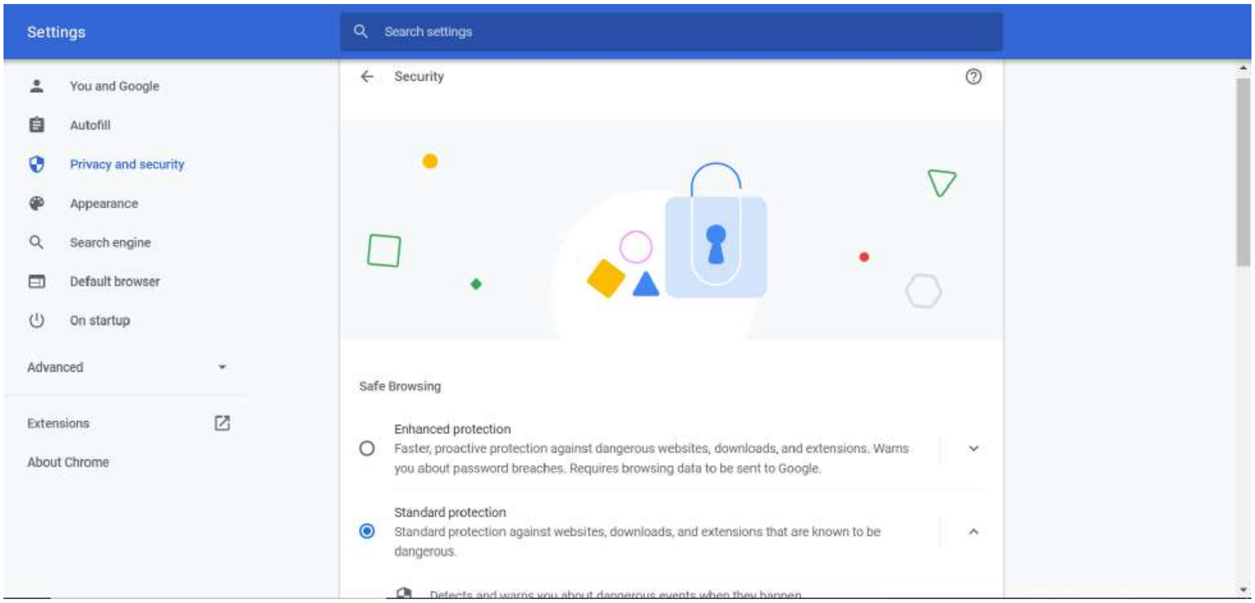
My Subscription Active

Web Boost Set up

Rank :

McAfee
Together is power.
McAfee Reviews
in Endpoint Protection Platforms
4.5
1478 Reviews

c) Web Application Security
Browser Security : Standard or Mild level



Safe Browsing

○ Enhanced protection
Faster, proactive protection against dangerous websites, downloads, and extensions. Warns you about password breaches. Requires browsing data to be sent to Google.

● Standard protection
Standard protection against websites, downloads, and extensions that are known to be dangerous.

Detects and warns you about dangerous events when they happen

d) Compliance Audit
https://www.asus.com/in/Terms_of_Use_Notice_Privacy_Policy/Official-Site

1) Your environment under protection of Disaster Management : YES
2) People involved are educated : YES
3) All components involved are following Indian standards : YES
4) All components are made in India : YES
5) Regular Internal Audit was covered : YES

VERDICT : It's safe to run the application in my place.