Web Vulnerability Scanning and Exploitation From Two Operating Systems

K. Hunter, R. Prasad

December 4th, 2022

Table of Contents

Abstract: 2

Introduction: 2

Background: 3

Methods: 4

a. Testing lab deployment 5

b. Vulnerability Scanning and exploitation- Black Box 6

c. Vulnerability exploitation- “White Box” 6

Results 7

Black Box Results 7

Nessus Scan 7

N-map Scan 8

OpenVAS and Nikto Scan 8

White Box Results 9

Exploiting Server Side Include Injection 9

Exploiting HTML Injection 10

Exploiting iFrame Injection 10

Exploiting Insecure FTP 10

Exploiting Insecure SNMP 11

Exploiting Insecure OpenSSL 11

Recommendations 11

Conclusion 12

References: 13

Appendix 15

Abstract:

Our start up penetration testing company is deciding which operating system we will adapt for our employees and systems. We have narrowed down our options to either Windows 10 or Kali Linux, both of which have tools and systems commonly used for penetration testing. In order to decide which one to adapt, we propose a comprehensive analysis of penetration testing against a vulnerable web application target running both Linux and Windows operating systems, in order to compare the efficacy of the different operating systems, their applicable tools and scripts, and their ability to identify and exploit vulnerabilities on our target machine. The main purpose of the analysis is to create a report on the usage of web application-based vulnerability tools, payloads, injection points or manipulation attacks from two different operating systems, in order to compare their efficacy. We also will compare both black box and white box penetration testing activities in order to determine the differences, strengths, and weaknesses of each approach. Ultimately, we hope to choose the best operating system and the best penetration testing strategy for our organization moving forward.

Introduction:

According to Positive Technologies cybersecurity research, More than 91% of web applications in 2021 experienced major vulnerabilities like SQL Injection, XSS, Content spoofing and Broken access control, predictable resource location etc. Additionally, the average web application has 15 vulnerabilities per site, two of which can be considered high severity [1].Web applications are exceptionally susceptible to attack when compared to other IT assets since they are connected to the Internet. Web forms and APIs are frequently used as attack vectors against web applications in order to manipulate user and machine inputs. Web applications generally require authentication and must allow traffic over several ports. Since websites must permit network traffic to and fro, hackers frequently target the most used ports and exploit them [9].

Thousands of companies and millions of users every year invest in penetration testing, where vulnerabilities are identified and fixed by professionals who mirror the actions of an attacker. Pentesters solve the most common vulnerabilities by automated scans and tests. Automated Pentesting has a variety of advantages, including increased speed and efficiency. On the other hand, manual Pentesting helps in identifying the minute

vulnerabilities. Since the number of web applications is constantly increasing, our company has secured funding and is setting up our systems in order to address the growing need for penetration testing.[2]

We propose setting up a virtual penetration testing lab, where we can compare and contrast penetration testing from two different attack machines-- one Windows 10 machine, and one Kali Linux machine. Both Windows 10 and Kali Linux have similar tools and systems available for penetration testing, but there are differences in ease of use, time, efficacy, and cost of penetration testing from either machine. By attacking an identical, vulnerable target web application in a virtual environment, we plan on evaluating these two different operating systems, in order to report our findings and make a final decision regarding which one to adopt. We are using bWAPP as our vulnerable target application; bWAPP is a PHP and mySQL based web application that runs on a linux server, and has more than 100 web vulnerabilities [6]. We use tools such as Nessus, N-map to perform scans and analyze network traffic for vulnerabilities like configuration problems or injection attacks.

Background:

Penetration testing can be considered as an attempt to evaluate the security of a computer system or IT infrastructure in a safe environment. In order to systematically compromise servers, endpoints, online applications, wireless networks, network devices and other possible sources of exposure, penetration testing is often carried out utilizing manual or automated techniques. Penetration testing typically simulates a variety of attacks that can threaten an organization. It can ensure that your system is robust enough to withstand attacks from authenticated and unauthenticated locations and various system roles. Furthermore, Pen testing helps to find weakness in the system, robustness of controls and support compliance; It helps to perform various tests which include Reconnaissance, Scanning, Analysis etc..,[3]. A basic overview of key differences between Kali and Windows is explained below. Kali Linux is a Debian-based Linux distribution aimed at advanced Pen-testing and Security auditing; whereas Windows 10 is the most secure Operating system ever built with Fast and Efficient Performance., Kali is designed primarily for "offensive security" whereas Windows serve many purposes. Kali runs fast even on older hardware, but Windows is slower than Linux. Kali provides centralized space for various functionalities like downloading and installing, searching etc.., In windows, it can be done at user convenience [5].

The main reason for web application based attacks is insecure coding. A critical vulnerability allows hackers to exploit data from various parts of the network. We use Buggy web application(bWAPP) as our Vulnerable application target. bWAPP is a shaky open-source web application that uses PHP and MySQL databases. It is intended to enhance the aptitudes of students, designers or individuals intrigued by IT security to find and anticipate web vulnerabilities; bWAPP has hundreds of vulnerabilities like SQL injection, Configuration problems, XXE, SSRF etc.., [6]

Organizations need web application scanning tools to prevent malicious hackers from gaining unauthorized access to confidential data and information. Although organizations make every attempt to install well-known network security and antivirus solutions, web apps have shown to be the bottleneck in overall enterprise security [11]. A network vulnerability scanner scans your computer across your network for open ports and services. It checks for services like vulnerabilities, configuration errors, and other information. Most of the vulnerability scanners perform web application scanning beyond the login interface (i.e.,) Scanning is done from the perspective of a malicious user.

In the current day, most of the security Professionals build their own Pen testing labs using Virtual machines to hone security skills in a safe environment. Virtual machines provide high scalability by hosting multiple pentesting machines with snapshot functionality. A virtual lab has the additional benefit of isolating from real-world systems and even your host machine. Penetration testing reveals vulnerabilities that allow attackers to access users, systems, networks, or applications, and is critical to an organization's ability to fine-tune its security standards [4]. Because our company is a penetration testing company, our main service will be to penetration test our client’s systems. Before we purchase and set up our computers and systems, we need to decide whether or not we are going to use Kali Linux or Windows in order to best reach our business objectives.[10]

Methods:

Initially, we will use Oracle’s VirtualBox, along with its “host-only adapter network” feature, to set up our virtual pen testing lab. One team member will be tasked with black box penetration testing, whereby they scan the vulnerable machine without any prior knowledge or credentials, as an attacker would. They will detail which vulnerabilities they were able to identify, and which ones they were able to exploit from both the Windows and Linux machines. Meanwhile, the other team member with an identical setup will perform “white box” penetration testing, whereby they have credentials,

information, and knowledge of known vulnerabilities in the target machines. (This is a variation on the typical white box penetration testing, where there isn't a list of known vulnerabilities). They will attempt to exploit each one of these vulnerabilities from both the Windows and Linux machines, and compare how successful they were on each. See diagram below regarding black box and white box penetration testing.[8]

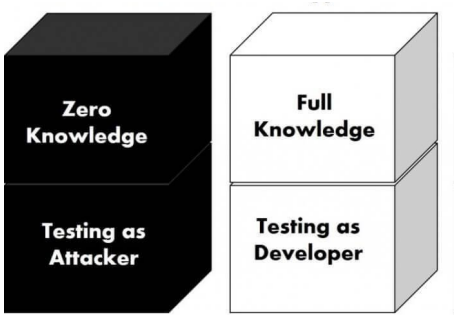

Figure A: Black box versus white box penetration testing

a. Testing lab deployment

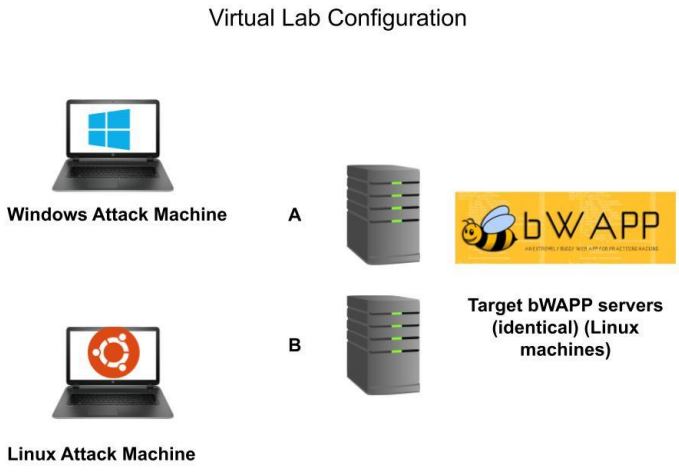

Figure B: Our Configuration

Our virtual machine configuration includes Windows 10, Kali Linux, and two bWAPP Linux vulnerable web servers. Each team member will use both machines to perform

penetration testing actions on the bWAPP VMs, where bWAPP server A will be probed only by the Windows machine while bWAPP server B will be probed only by the Kali Linux machine.

b. Vulnerability Scanning and exploitation- Black Box

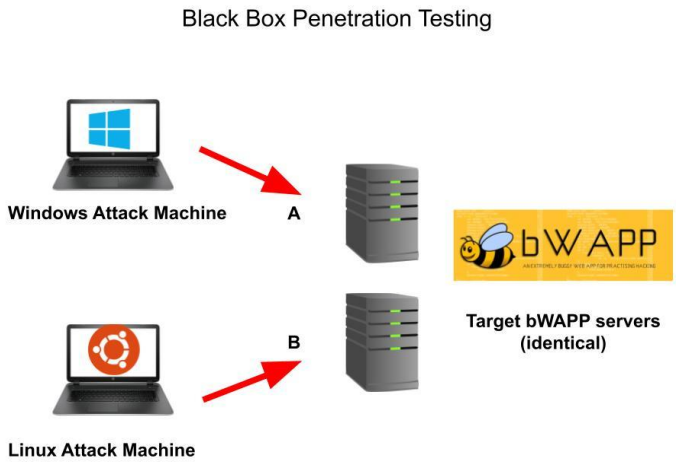

Figure C: Black box scanning and exploiting

Our first method will be blind penetration testing-- in order to maintain consistency, team member 1 will use first the Linux and then the Windows machine to follow identical steps of footprinting, analyzing network traffic, and performing scans. The tools team member 1 will use for this portion are Nessus, nmap, and SQLMap for both windows and linux (different versions of these tools are available on both operating systems).[7][8]

c. Vulnerability exploitation- “White Box”

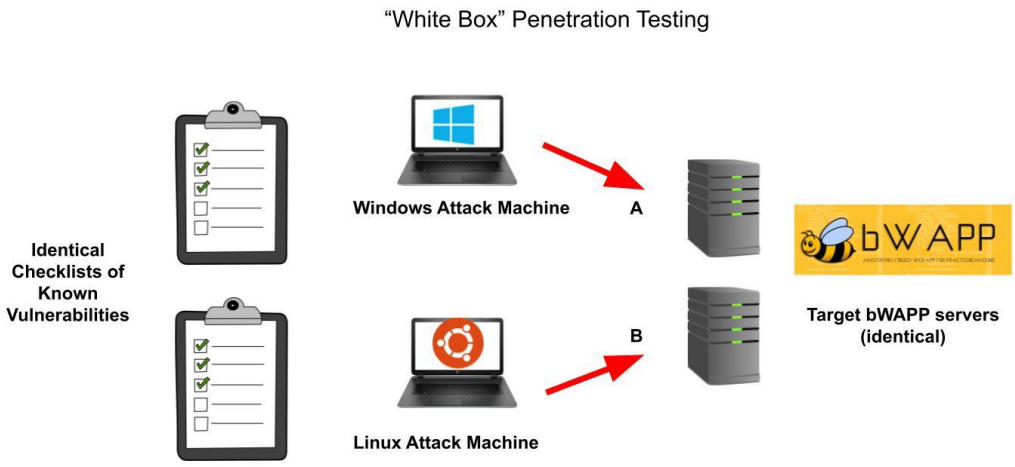

Figure D: "White box" exploits

Our second method will be choosing 6 of the known vulnerabilities in the web app, 3 injection and 3 misconfiguration: Server Side Includes Injection, HTML Injection, and iFrame Injection, and insecure FTP, insecure SNMP, and insecure OpenSSL. Team member 2 will attempt to exploit these vulnerabilities using the penetration testing tools on both Windows and Linux attack machines. The tools they will use include MetaSploit, netcat, and wireshark [12]. Team member 2 will compare which operating system is quickest at exploiting each vulnerability.

Results

Note: See Appendix for Figures

Black Box Results

Team member 1 performed vulnerability scanning with help of scanners like nmap, Nessus, OpenVAS and Nikto. The main motive of using various scanners is to find out the efficacy of scanners and to get a clear overview regarding vulnerability scanners.

Nessus Scan

Nessus Scanner discovered vulnerabilities with respect to severity and provided the steps of possible remediation. When Team member 1 scanned the Vulnerable target

bWAPP , it gave output of the scan ranging from low to critical (**Figure E**). Nessus scan results had different vulnerabilities such as SQL injection, HTML Injection, XSS, CSRF, Web services issues etc. Nessus scanner usually scans over 65000 ports and possibly searches for all vulnerabilities in the target host. While performing the Nessus scan in Kali, the time taken for the scan is lower than that of the Windows operating system. The vulnerability list had few changes when the scan was performed in the Windows operating system. There were few additional vulnerabilities which were encountered in the Kali operating system. As per the Nessus scan most of the vulnerabilities can be exploited easily. Kali Linux was easier to perform scans as the software is pre-installed, Team member 1 faced issues while running scans in Windows 10 operating system.

N-map Scan

To get a better understanding of discovering vulnerabilities, Team member 1 performed the n-map scan on the same vulnerable target host to check the efficiency of the scan. N-map scan discovered the open ports and the services provided by them. Performing N-map has been easier in both operating systems. The main advantage of running n-map has been checking the status of each IP, the user can check the security by exploiting the required IP. Team member 1 performed an N-map scan on bWAPP and metasploitable2, both the vulnerable targets finished the scans around the same time. From these scans, we can see the status of ports and services of respective hosts. (**Figures F and G**)

OpenVAS and Nikto Scan

Team member 1 performed scans like OpenVAS, Nikto scans which ran for around 20 to 40 minutes. The scan time was comparatively longer than the Nessus and n-map scans but the output results which had the vulnerabilities list is almost identical to the Nessus. OpenVAS scan was easier to perform in the Kali operating system, whereas to perform a similar scan in Windows operating system is quite a complex process which includes the use of a hypervisor. Nikto is helpful in finding information of specific directories and checks syntax of the database to find errors. (**Figure H**)

The majority of vulnerabilities that were found during scanning from two operating systems were almost identical. Efficient and wide variety of tools were available in Kali compared to Windows.

White Box Results

Team member 2 looked at the bWAPP list of known vulnerabilities and chose three injection attacks: Server Side Include Injection, HTML Injection, and iFrame Injection. After conducting research on each of these exploits, she timed herself executing each attack on both the Linux and Windows machines. She then chose three misconfiguration vulnerabilities: insecure FTP, insecure SNMP, and insecure openssl and attempted to exploit them. See table 1 below for the time taken to complete these exploits, rounded to the nearest 5 minutes:

	Windows Attack Machine Time	Linux Attack Machine Time
Server Side Include Injection	20 minutes	15 minutes
HTML Injection	10 minutes	10 minutes
iFrame Injection	10 minutes	10 minutes
Insecure FTP	25 minutes	15 minutes
Insecure SNMP	20 minutes	15 minutes
Insecure OpenSSL	NA (attempt of 2 hours)	65 minutes

Table 1- Time for each exploit on both Windows and Linux

Exploiting Server Side Include Injection

A server side include injection takes advantage of something called a server side include (SSI), which is used to dynamically load custom content into web pages. For example, SSI can load information about the user's local time zone into the website content for a customized experience [14]. If input is not properly validated or sanitized on web applications using SSI, it is possible to inject SSI directives into the server, and execute a malicious payload upon reloading the site. In order to take advantage of this vulnerability on bwapp, team member 2 used SSI on the program portion of the page (**Figure I**) in order to extract information about the web server, including the passwd file

(**Figure J**), before ultimately opening a reverse shell using a combination of an SSI attack and netcat in order to gain direct access to the backend of the web application (**Figure K**). These steps were nearly identical on both the Windows and Linux machines, although the Linux machine had netcat preinstalled while the Windows machine required installation, adding time.
Note: See Appendix for Figures

Exploiting HTML Injection

HTML Injection is a type of injection where the payload is a piece of HTML code, which is inserted into a vulnerable area of the website via user input where there is insufficient input sanitization. This can change the appearance of the website, or even run a malicious script. Team member 2 exploited this vulnerability first by altering the appearance of the website by appending text. Then, she demonstrated that a malicious script could be injected into the website by prompting a browser alert. This attack was nearly identical on both the Windows and Linux machines, as it only involved direct interaction with the web application via a browser. [13] (**Figures L and M**)

Exploiting iFrame Injection

The bWAPP web application has a vulnerability where the iFrame parameters are exposed in the URL, making it extremely easy for any individual to simply inject their own content into the bWAPP website and distribute a misleading link to others, a combined social engineering and technical attack. Team Member 2 demonstrated this by attempting to display her own personal website inside the bWAPP web application. There was an issue loading the website since the Host-Only Adapter Network was not directly connected to the internet, but the content on the site was still successfully altered. Since this injection involved only direct interaction with the web application via a browser, the Windows and Linux processes were identical and took a similar amount of time. (**Figure N**)

Exploiting Insecure FTP

Since FTP is insecure, it is possible to use a simple packet sniffer to extract credentials and information from any FTP transfer on the web application. To demonstrate this vulnerability, team member 2 used wireshark during an FTP session, and was able to see the credentials used to transfer the file. The way to fix this would be to use FTPS,

which encrypts traffic so that any MITM or packet sniffers cannot see information in plain text, but bWAPP only uses FTP which is an insecure version. The difference between Linux and Windows systems in this case was more significant-- wireshark had to be installed, as well as ftp, but both of these programs were already installed in Kali. Aside from this, the process took longer to complete on the Windows OS. (**Figure O**)

Exploiting Insecure SNMP

SNMP stands for Simple Network Management Protocol, and it allows different devices on a network to communicate with each other. Using Metasploit as the penetration testing tool, team member 2 ran the snmp-login scan, which attempts to login to various SNMP devices using the default credentials in case any of them are misconfigured. The exploit was successful at gaining access to read/write privileges for the SNMP device enabled on the web application, which gives us the power to completely alter or attack the SNMP device. Since the Metasploit console had to be downloaded into Windows, it took longer than the Linux machine. Additionally, running the exploit took longer on Windows. (**Figure P**)

Exploiting Insecure OpenSSL

Since bWAPP uses a version of OpenSSL that is deprecated, it is vulnerable to the heartbleed attack, where an attacker can steal important information that SSL should have encrypted. First, to determine that it was, team member 2 ran an nmap script called openssl-heartbleed and determined that the web application is, indeed, vulnerable to the heartbleed attack. Then, she downloaded a malicious python script from the internet and ran it on port 8443 on the web application from the attack machine. The response from the server was login credential information that was “leaked” due to this vulnerability. An attacker could use this information to escalate privileges and take control of the web server. This attack was hard enough to complete on the Windows machine that it was only completed on the Linux machine. (**Figure Q**)

Recommendations

In order to finalize the operating system for our organization, we have performed both black box and white box penetration testing activities and identified the differences between two different operating systems, Windows and Linux. However, we in no way

explored all of the vulnerabilities in bWAPP comprehensively, and for that reason, in the future, we would like to explore in-depth some of the other common web application vulnerabilities, such as XSS and CSRF.

We also realized that most attacks on web applications are already known to the security community, and common misconfigurations and vulnerabilities can be found just by using simple scanning tools. This means we can prevent the majority of attacks simply by routinely scanning our system in order to correct those known vulnerabilities and avoid the most common attacks.

In setting up our penetration testing company, we need to ensure that the most common and important security tools are pre-installed and available for our penetration testers to use. This would save a lot of time and allow them to focus on helping our clients identify the vulnerabilities on their systems. We also will constantly be updating all of our software and operating systems as new security flaws are identified and corrected.

Conclusion

Our team has determined based on qualitative experience on the different operating systems and quantitative evidence of efficiency that we will invest in a Kali Linux setup in our penetration testing company. In most cases, Windows and Linux operating systems follow very similar steps, especially in the case of running vulnerability scanners like nmap and Nessus. Additionally, the output of these scanners is almost identical, with a few exceptions. Therefore, the advantages that Kali Linux has, like pre-installed security features and extensive command line functionality, weigh heavily in our decision, as they make navigation and the use of the tools easy and intuitive. Windows, on the other hand, has a relatively clunky GUI interface with a more limited command line, and we ran into some issues with our scans and exploits on the Windows machine because of its more complicated environment.

When it comes to white box versus black box penetration testing, there are advantages and drawbacks to both: black box penetration testing allows the tester to see what an attacker would see, and take actions that an attacker would take, thus more accurately representing an attack situation in the real world. On the other hand, because the black box tester has no inside knowledge of the system or its intricacies, they may miss some

of the deeper vulnerabilities that may be exploited by inside threats. White box penetration testers have the perspective of a developer, which means they know exactly what is going on with their system and how information moves and flows. This saves time and money, and allows them to execute many different attack vectors. However, white box penetration testers may need an outside perspective to be able to identify vulnerabilities that weren't on their radar, and won't be able to take on the perspective of a malicious attacker since they have so much inside knowledge.

Because of these different strengths and weaknesses, we recommend a combined approach-- using black box penetration testing methods to constantly identify the vulnerabilities that an attacker may find in probing the network, while also using white box penetration testing methods to pinpoint deeper vulnerabilities and identify how to solve them. In leveraging both approaches, our company can ensure that our own systems and those of our clients are secure from insider and outsider attacks.

References:

[1] Positive Technologies- "Threats and vulnerabilities in web applications 2020–2021", https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020-2021/ , Jun 14, 2022 .

[2] Medium - "Web Statistics", https://medium.com/@aplextorlab/web-statistics-69493eebbd01 , Dec 18, 2018

[3] Getastra - " Security Audit" https://www.getastra.com/blog/security-audit/web-application-penetration-testing/, Sep 26, 2022

[4] Purplesec - "Web Application Penetration Testing", https://purplesec.us/web-application-penetration-testing/, Nov 10, 2021

[5] Hackr - " Linux vs Windows", https://hackr.io/blog/linux-vs-windows, Aug 15, 2022

[6] HackRoyale - " How To Practice Hacking With bWAPP ? ", https://www.hackingroyale.com/hacking-with-bwapp/, Jun 16, 2020

[7] ItsFoss- “Best Kali linux tools”, https://itsfoss.com/best-kali-linux-tools/, Mar 4, 2020

[8] Devcount- “12 Best Windows Pentesting tools in 2022”, (https://devcount.com/windows-pentesting-tools/), Jan 4, 2022

[9] Acunetix- “Web Application attack”, https://www.acunetix.com/websitesecurity/web-application-attack/, Feb 20, 2022

[10] Thomas Wilhelm- “Professional Penetration Testing” , https://www.sciencedirect.com/book/9781597499934/professional-penetration-testing”, Mar 4, 2013

[11] Design Rush - “Vulnerability Scanning: Definition, Strategies & Types of Vulnerability Scanners”, https://www.designrush.com/agency/cybersecurity/trends/vulnerability-scanning, Aug 31, 2022

[12] J.M. Porup and Josh Fruhlinger - “11 penetration testing tools the pros use” “ https://www.csoonline.com/article/2943524/11-penetration-testing-tools-the-pros-use.html, Dec 23, 2021

[13] Imperva-“HTML Injection”, https://www.imperva.com/learn/application-security/html-injection/#:~:text=Hypertext%20Markup%20Language%20(HTML)%20injection,a%20previous%20interaction%20with%20users.

[14] Marc Dahan- Compariatech’s “SSI injection attacks and how to avoid them” https://www.comparitech.com/blog/information-security/ssi-injection-attacks/#:~:text=SSI%20(Server%2d Side%20 Include),%2C%20locally%2C%20by%20the%20web server, Jun 14, 2022

[15] Redscan- “Types of Pen Testing: Black Box, White Box, and Grey Box” https://www.redscan.com/news/types-of-pen-testing-white-box-black-box-and-everything-in-between/#:~:text=A%20white%20box%20penetration%20test,many%20attack%20vectors%20as%20possible.&text=In%20a%20black%20box%20penetration,to%20the%20tester%20at%20all, July 26, 2022

Appendix

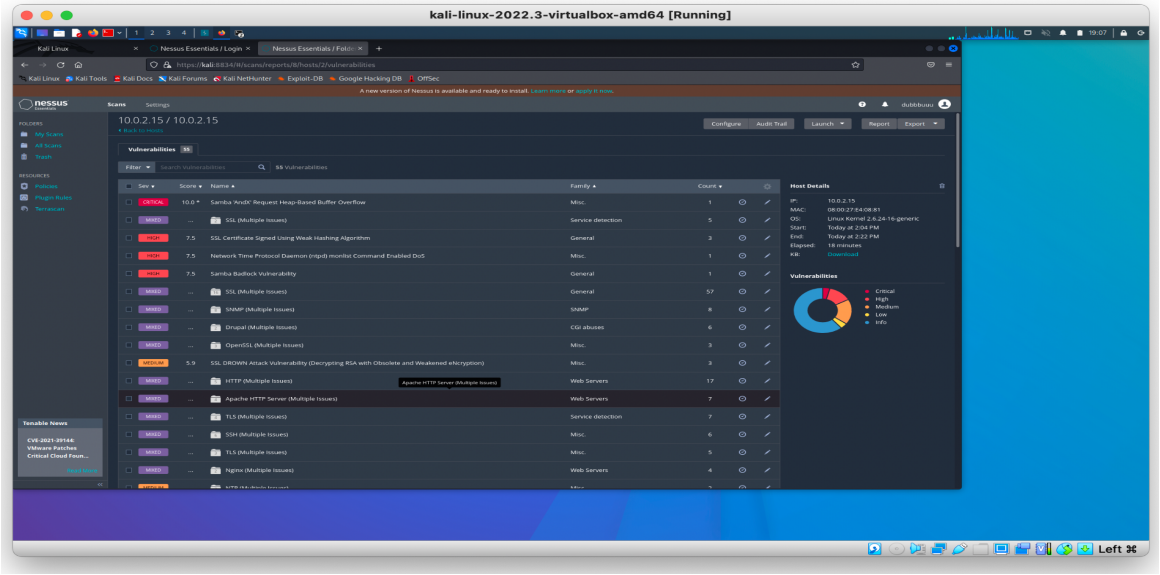

Figure E: Results of Nessus Scan

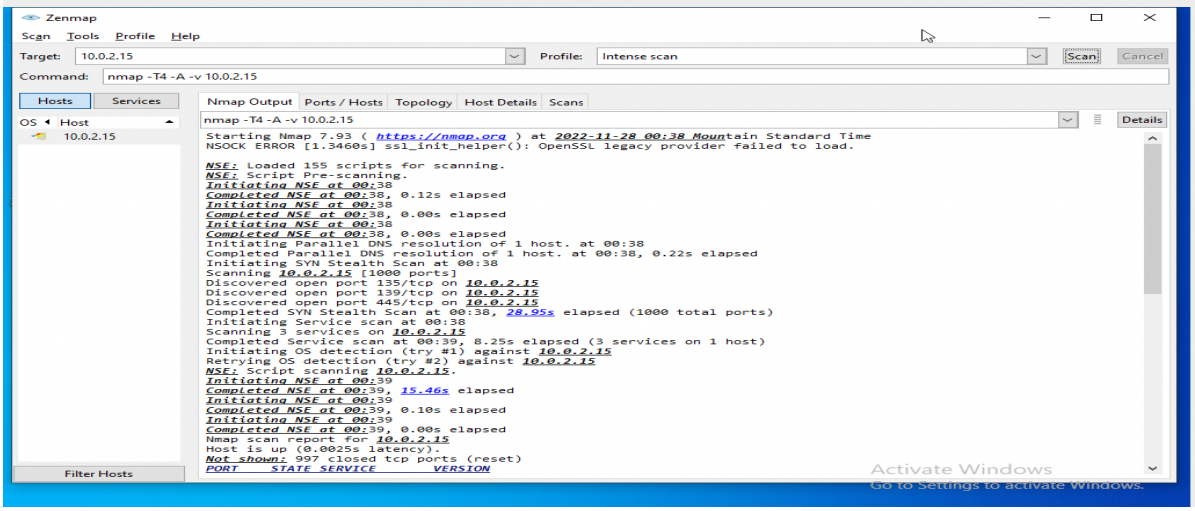

Figure F: N-map Scan 1.1

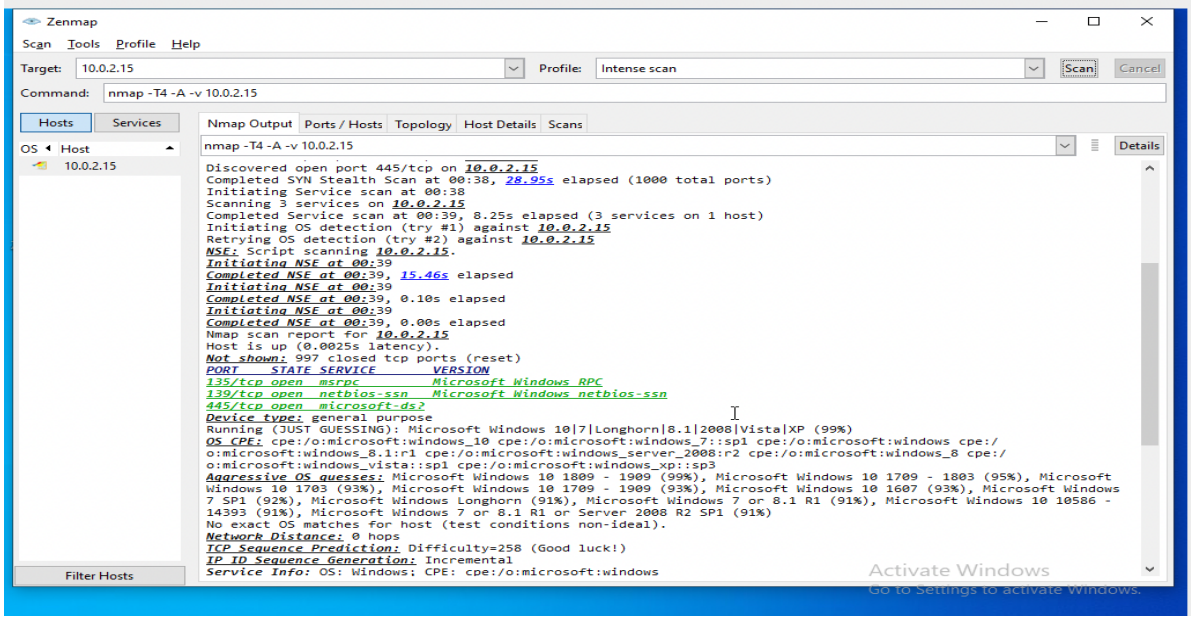

Figure G: N-map Scan 1.2

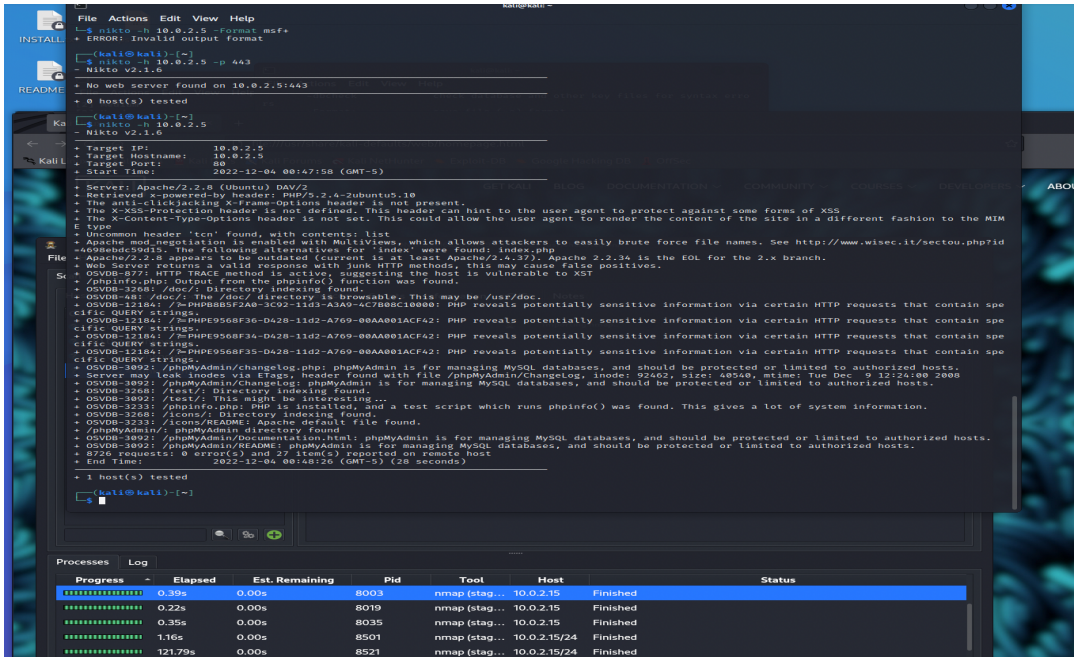

Figure H: Nikto and OpenVAS scanning

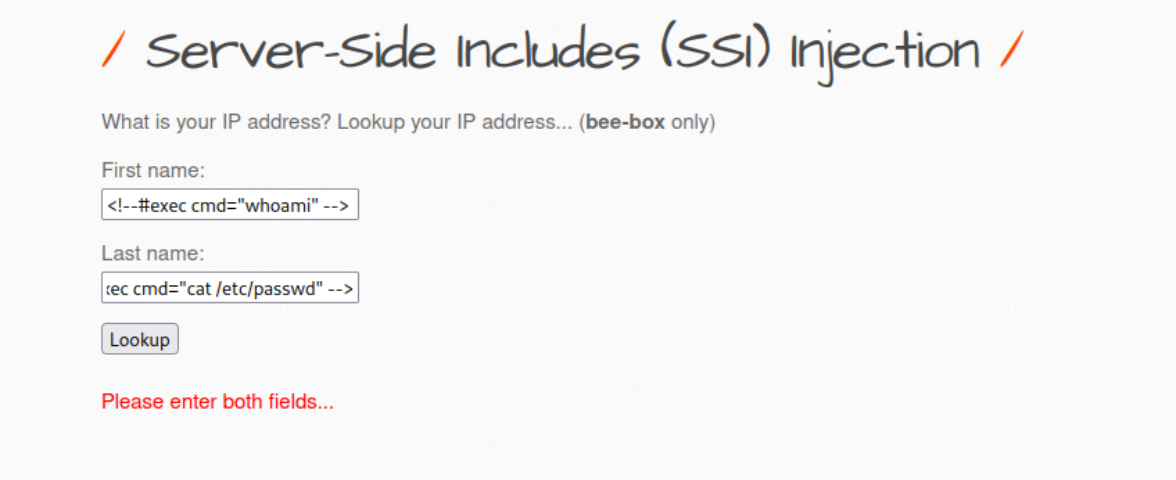

Figure I: Server-Side Includes Injection payload

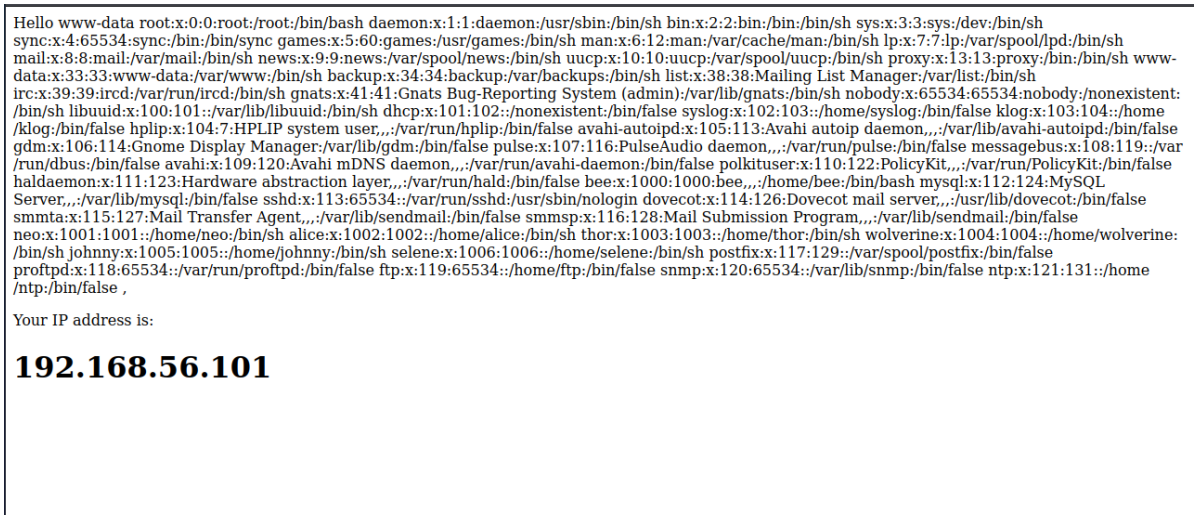

Figure J: Server-Side Includes Injection Result

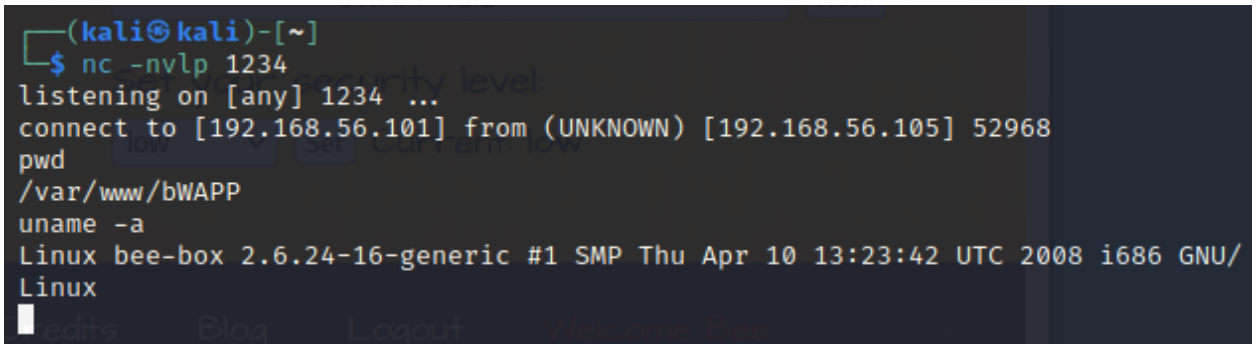

Figure K: Server-Side Includes Reverse Shell

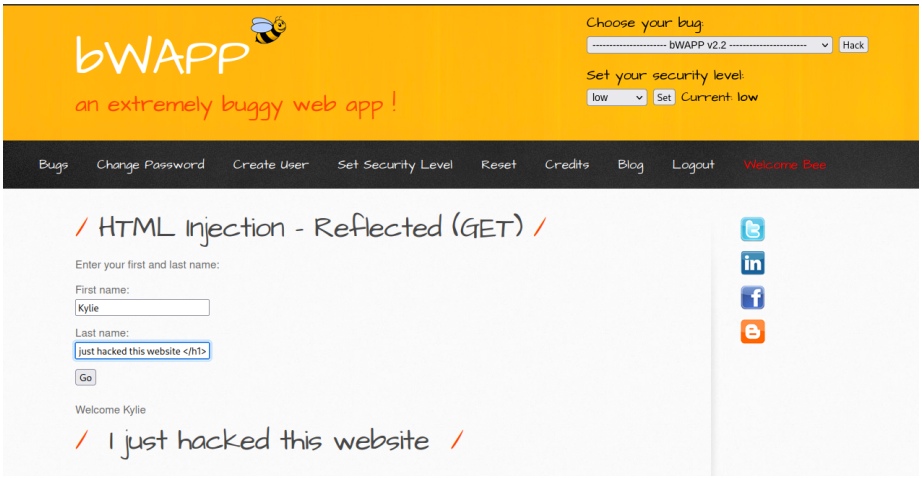

Figure L: HTML Injection Appearance Modification

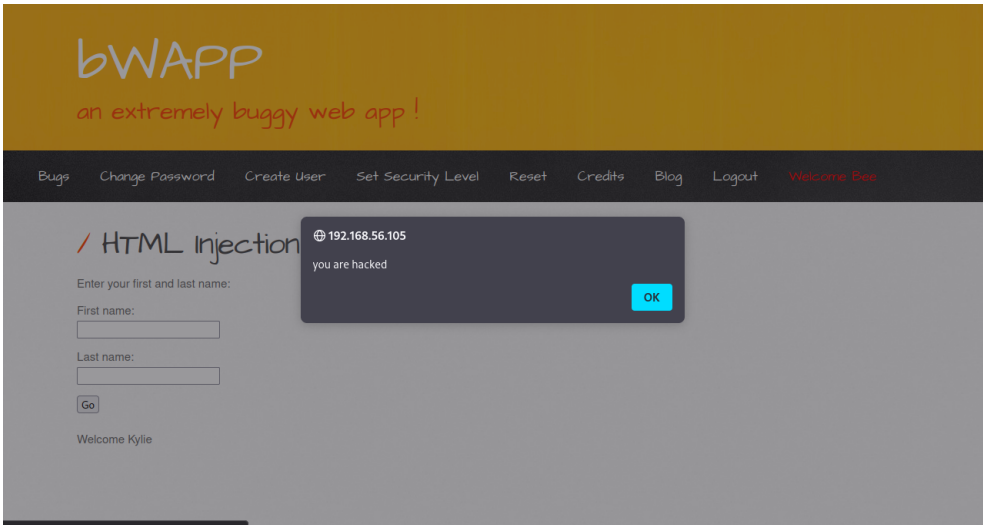

Figure M: HTML script injection

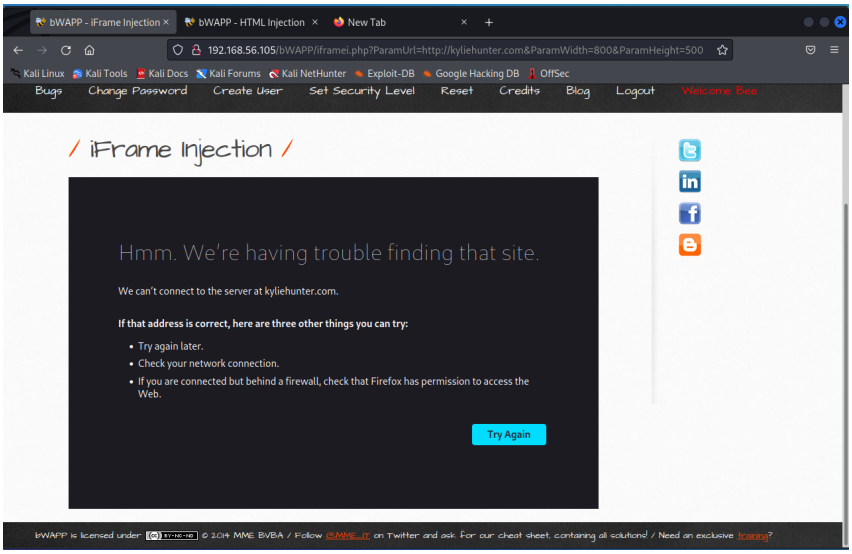

Figure N: iFrame Injection Result

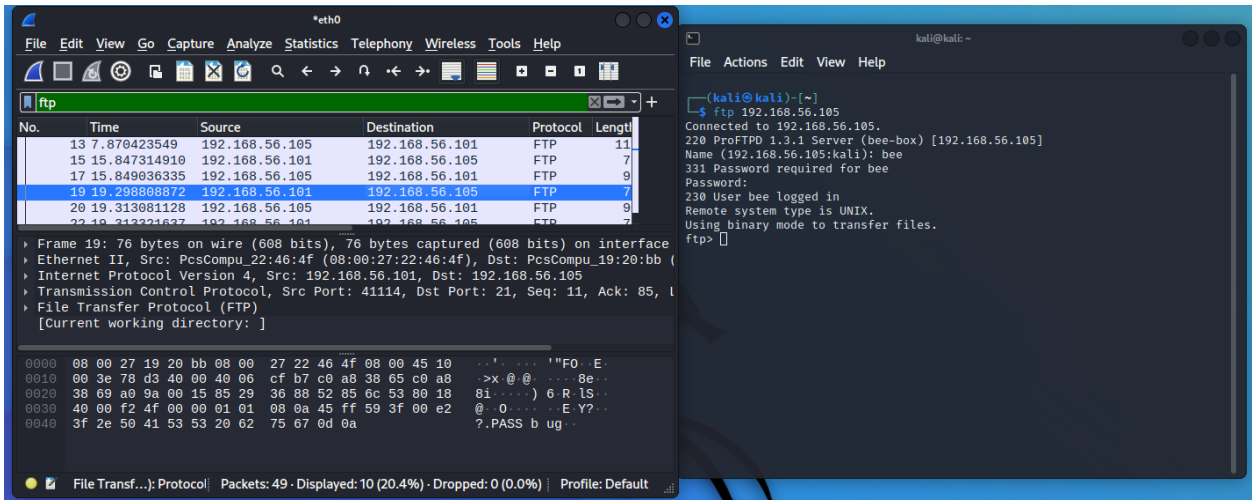

Figure O: Wireshark intercepts password from FTP

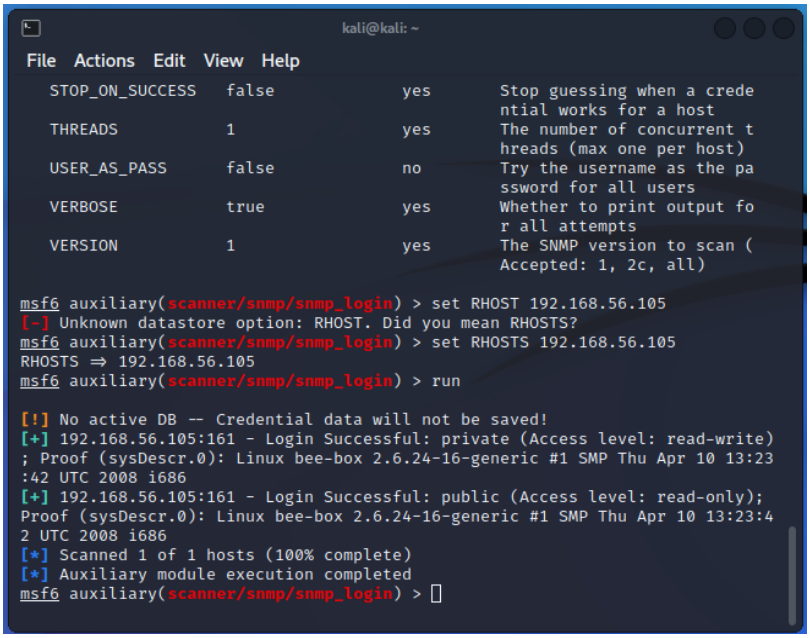

Figure P: msfconsole snmp login attack

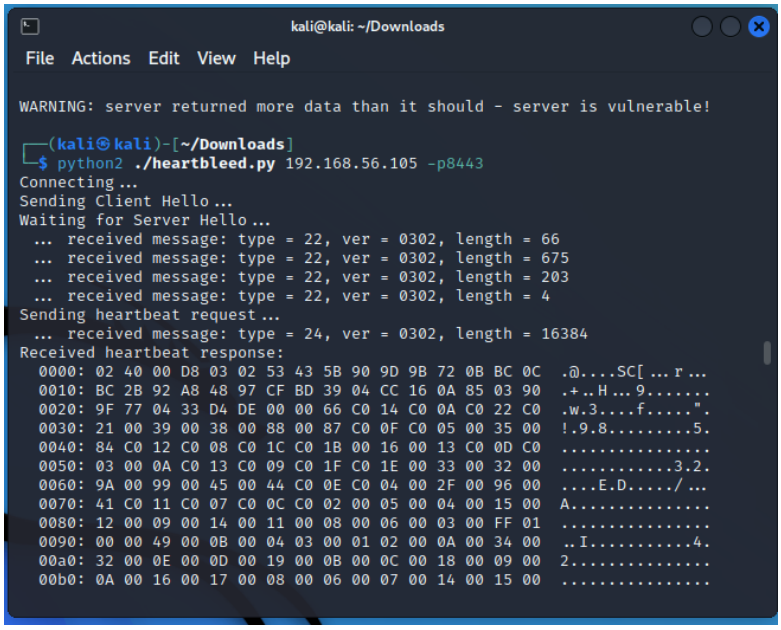

Figure Q: Results of the heartbleed python script