Penetration Testing Report

IT430 Ethical Hacking for SysAdmins

By Tasin Naveed

Date: 04/28/2022

**Table of Contents**

**Table of contents**……………………………………………………………. 2

**Introduction**………………………………………………………………… 3

* Project overview

**Technical Report**……………………………………………………………. 4

* Test 1…………………………………………………………………... 4
* Test 2…………………………………………………………………... 5
* Test 3…………………………………………………………………... 6
* Test 4…………………………………………………………………... 7
* Test 5………………………………………………………………….... 8

**Introduction**

**Project overview:** For this project I have conducted an abbreviated pentesting report through which I assess the security vulnerabilities on Windows XP, Windows 7, Linux Ubuntu. In my demonstration, I will show how to exploit some of these operating systems vulnerabilities. The essential tools I will need to conduct these tests are a virtual Kali Linux machine, a virtual instance of each of the three operating systems devices, (Windows XP, Windows 7, and Linux Ubuntu), IP address of each machine. Kali Linux is an operating system (OS) designed for penetration testing with many tools and utilities used for monitoring networks, discovering exploits, and carrying out those exploits. I will also refer to a book written by Georgia Weidman,” Penetration Testing a Hands-On Introduction To Hacking” which details how to simulate some attacks using Kali machine exploit frameworks. I will address the following elements for each test in my technical report:

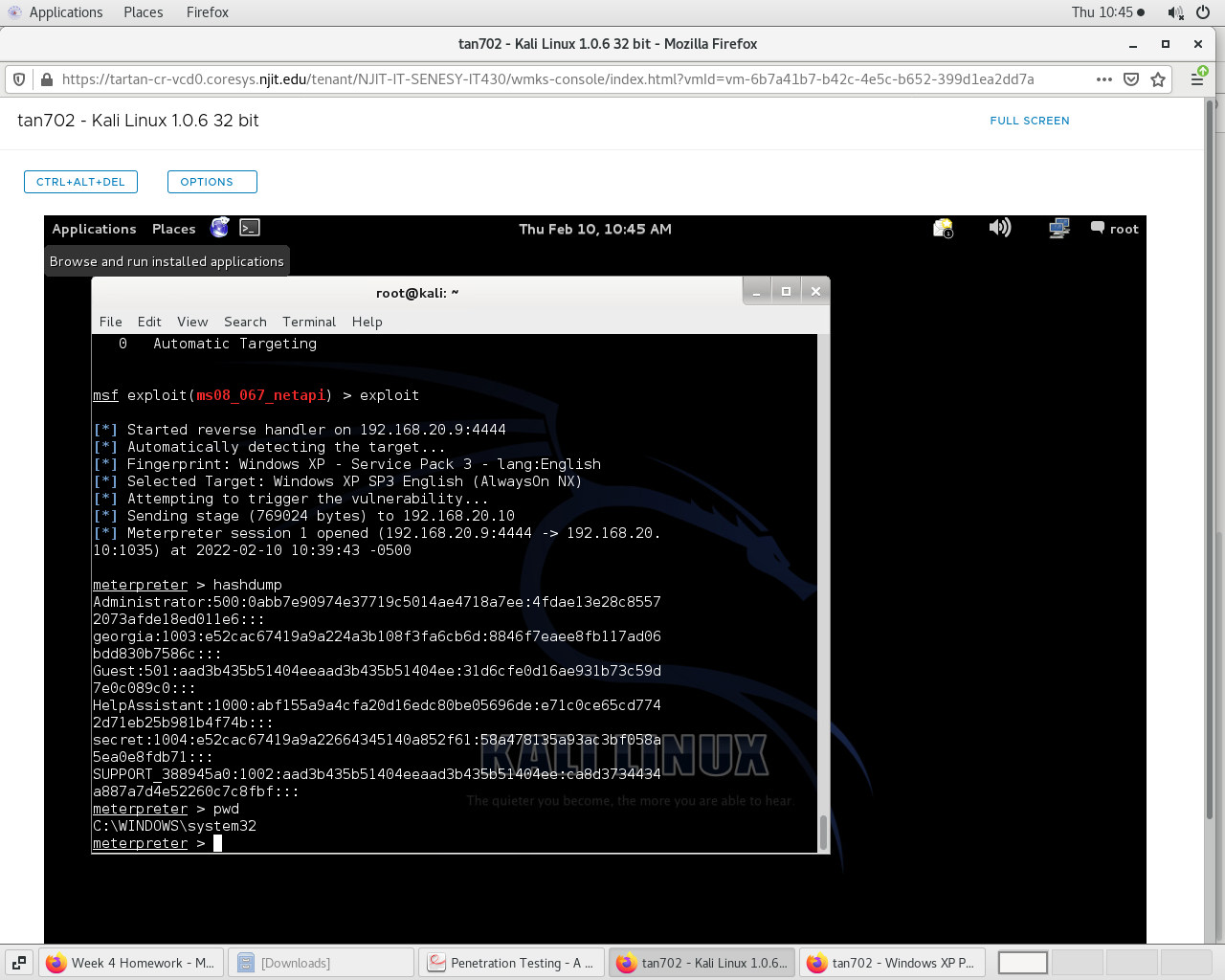
* Vulnerability being tested – to include basic theory of what the vulnerability entails and why it was selected for test.
* Configuration of devices used to perform the test
* Test results
* Recommended mitigation

**Technical Report**

**Test 1: Using Exploit - ms08\_067\_netapi**

ms08\_067\_netapi is one of the most popular remote exploits against Microsoft Windows. It is considered a reliable exploit and allows access as SYSTEM - the highest Windows privilege. I have used the Metasploit framework on kali linux (192.168.20.9) to use this exploit against Windows XP (192.168.20.10). I have set an automatic target and used a meterpreter shell as payload which provided me with an interactive shell from which I could explore the target machine and execute code. I was able to use the “hashdump” command that retrieved the contents of the local SAM database, allowing me to get the local user IDs and the password hashes.

**Proof of successful access gain:**



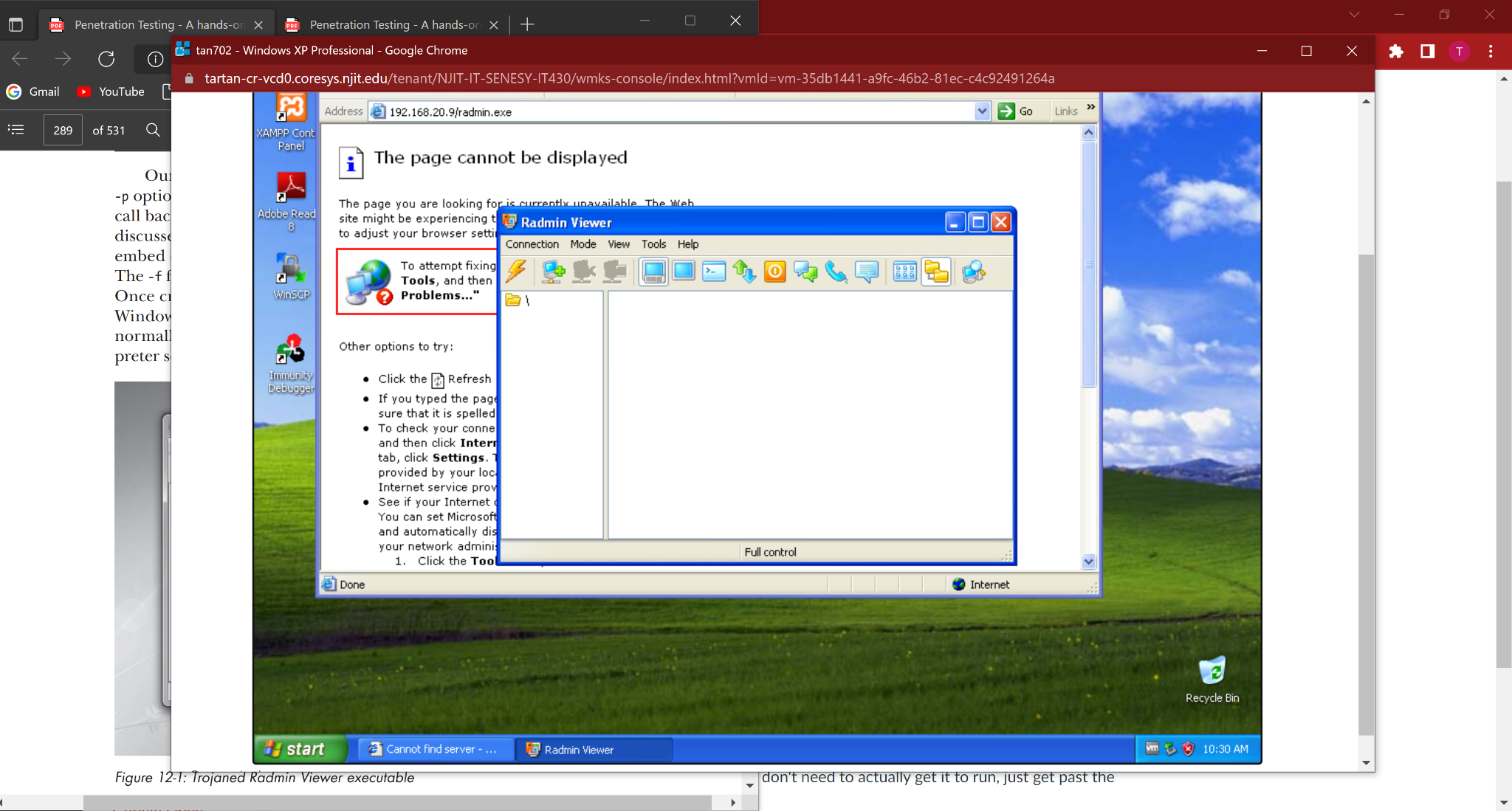
**Mitigation:** Microsoft released an update for patching this vulnerability and it is recommended that users apply the update immediately.

**Test 2: Using Malware to gain access**

I have built a trojaned Windows executable that appeared normal to the targeted machine and sent me back a Meterpreter session to gain the control. For creating this trojan I have used the msfvenom tool in kali linux which embedded a payload inside the executable. I have used the -x and -k flags to build a trojaned Windows executable called radmin.exe that appeared normal to users but which sent me a Meterpreter session in the background. To do so I had to start a reverse handler for listening connection from the targeted machine. As soon as the executable was running in the targeted machine I had the meterpreter shell. And importantly in the targeted machine the executable looked very normal and ran successfully.

**Proof of creating malware and executing in windows machine:**



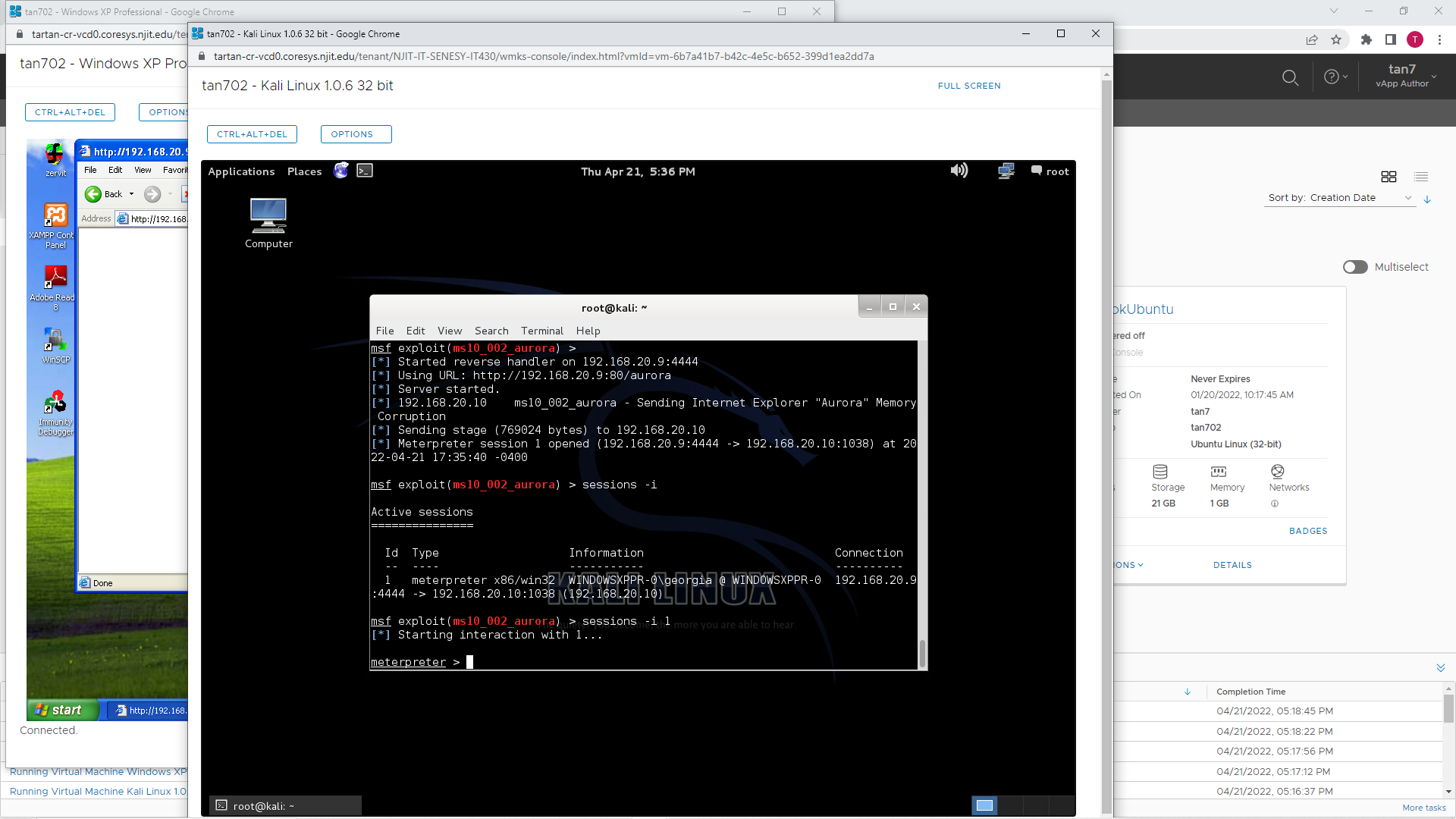


**Mitigation:** Updated antivirus softwares has signatures to detect these kinds of trojans. So it’s best to have an updated antivirus software and scan the executable file before running.

**Test 3: Using Exploit - ms10\_002\_aurora**

During this test I have used the aurora which is a famous vulnerability in Internet Explorer. I used my kali machine and targeted windows xp’s internet explorer for the attack as the browser wasn’t updated. In order to do that I relied on Metasploit to take control of the target machine by attacking the internet explorer using the Aurora Metasploit module, **exploit/windows/ browser/ms10\_002\_aurora**. For this case I have chosen **windows/meterpreter/reverse\_tcp** payload which provided me with a Meterpreter session.

**Proof of the attack:**

****

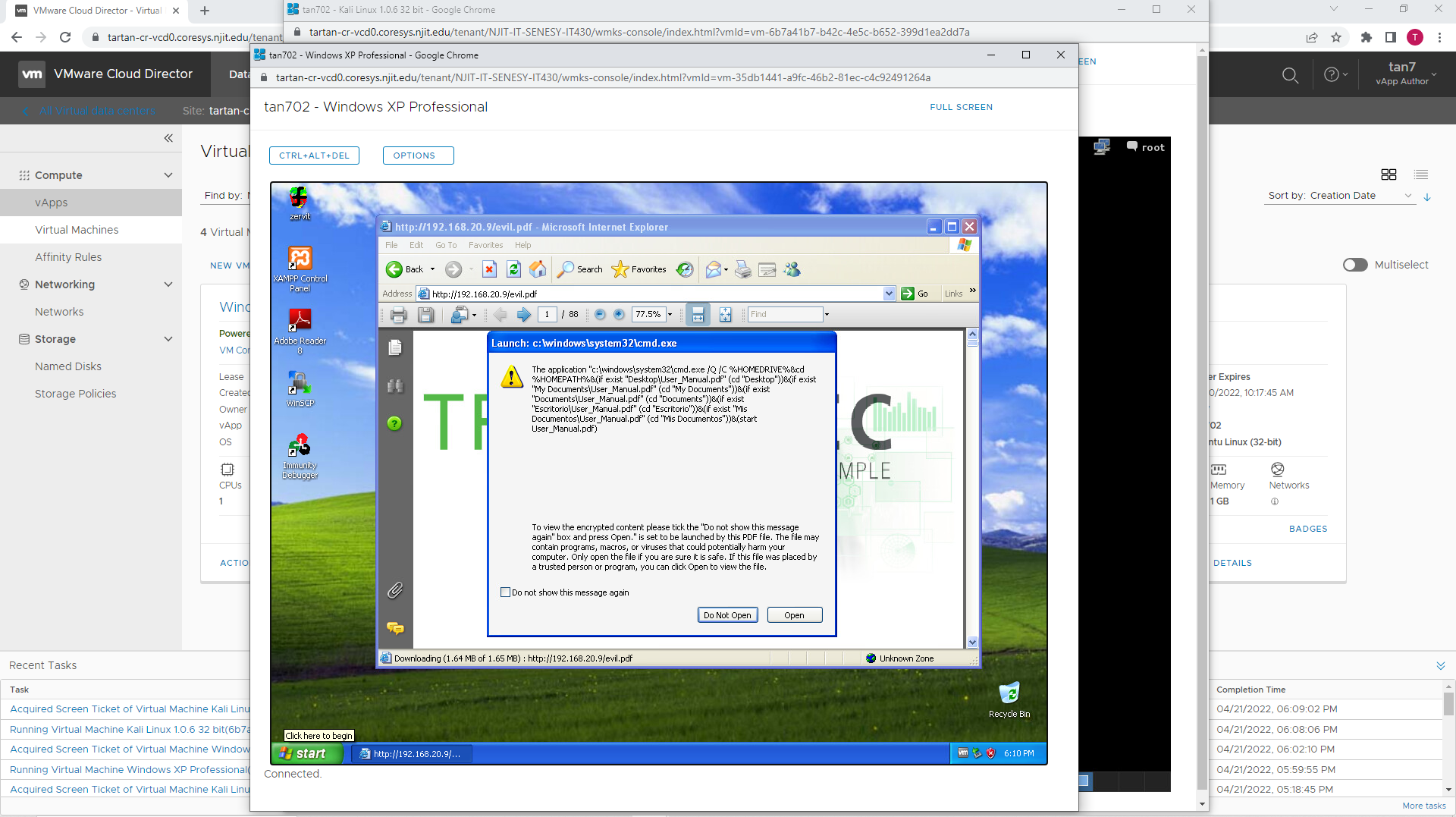
**Mitigation**: The Aurora vulnerability was patched in 2010, but users and organizations are bad at keeping their browsers up to date, so this exploit still finds targets today. So, it's highly recommended to update the browsers to get rid of the Aurora vulnerability.

**Test 4: Using PDF Viewer Exploit**

The Windows XP target has an outdated version of Adobe Reader 8.1.2 installed that is subject to **CVE-2008-2992**, a stack-based buffer overflow. I have used the corresponding Metasploit module **exploit/windows/fileformat/ adobe\_utilprintf** to see the result. This module simply creates a malicious PDF; hosting it for delivery but I had to set up a handler to listen to the upcoming connection. Metasploit used the default payload, **windows/meterpreter/ reverse\_tcp** on port 4444. When we enter exploit, Metasploit generates a PDF that will exploit this vulnerability in a vulnerable version of Adobe Reader on Windows XP SP3 English.

Then I have tried another pdf attack which is a little bit advanced.This time I have embedded a malicious executable inside a PDF. The corresponding Metasploit module is **exploit/windows/ fileformat/adobe\_pdf\_embedded\_exe**. Instead of exploiting the software as soon as the PDF is opened, the generated PDF will prompt the user for permission to run the embedded file. The success of our attack is contingent on the user allowing our executable to run.

**Proof of attack:**

****

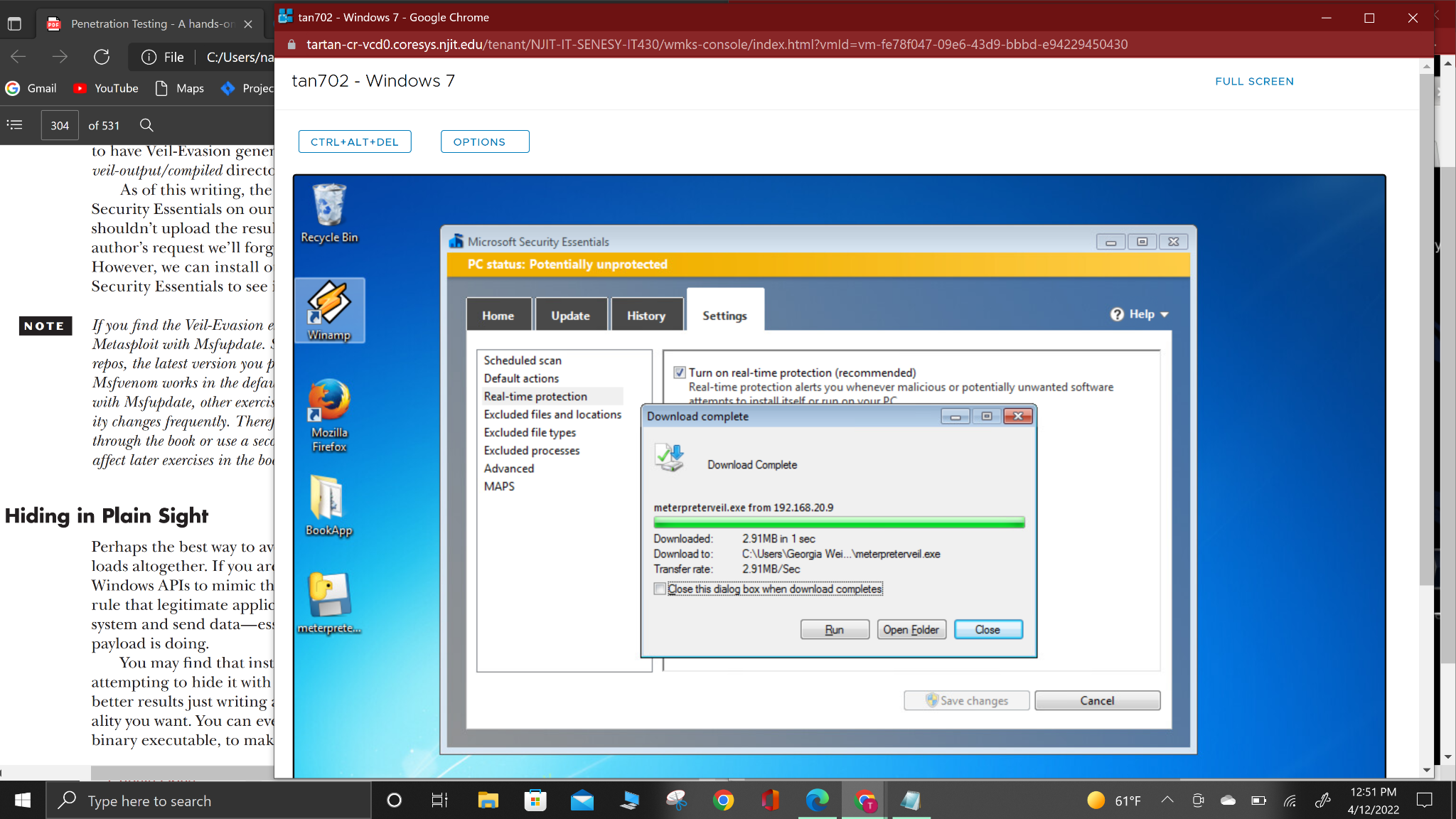
**Mitigation:** The most popular PDF viewer for Windows systems is Adobe Reader. Like browsers, Adobe Reader has a history littered with security holes. Also like browsers, even when a patch-management process is in place, regularly updating the underlying operating system, PDF software is often forgotten, and remains at an older, vulnerable version. So, the best way is updating software on a regular basis.

**Test 5: Evading Antivirus of windows 7 with Veil Evasion**

For this test I have used Windows 7 as a targeted machine and made sure to turn on real time protection in Microsoft Security Essentials. Then I have created a payload in my kali linux. The payload was encoded by Metasploit’s encoder. Even though this process makes it more difficult for antivirus vendors to create signatures for the payload, it was not enough to evade the antivirus. Then I used msfvenom’s encoder named x86/shikata\_ga\_nai which had an excellent rank, but it wasn’t good enough. Then I have created a custom c script and used hyperion to encrypt the executable which was able to evade MSE.

But the easiest option was to use veil-evasion. Veil-Evasion is a Python framework that automates creating antivirus-evading payloads, giving users the choice of multiple techniques. Veil-Evasion prompts to select either Msfvenom to generate the shellcode or to provide custom shellcode. For my purposes, I have chosen Msfvenom. The default payload was **windows/meterpreter/reverse\_tcp**. Then I have set the options for LHOST and LPORT, and named the file meterpreterveil for the generated executable. Finally, Veil-Evasion offers two Python to executable methods. From windows 7 I could easily download the file without getting detected by Antivirus while Microsoft real time protection was on.

**Proof of downloading meterpreterveil.exe:**

****