Introduction/Overview:

Throughout the course of the semester, we performed numerous exploits that addressed a variety of topics related to pen testing and general hacking perspectives. Since all of the exploit setups differ from one another, we had to configure different factors in relation to varying configurations in order to execute the following exploits in this report. In order to conduct the tests in this lab, we are using four virtual machines: BookUbuntu, Kali Linux, Windows 7 Professional, and Windows XP Professional. To carry out the bulk of our exploits, we will mainly use Kali Linux as our testing platform. This is thanks to Kali Linux's built-in tools, which can be accessed using msfconsole to obtain access to a target machine. After that, we must launch a meterpreter shell and monitor the target device remotely. We are using an internal private network for our network setups, with identical IP addresses ranging from 192.168.20.9 to 192.168.20.12 allocated to each of the four machines. While none of the virtual machines had internet connectivity, this made the testing phase of the attacks ten times easier. To sum up, the types of tests we conducted this semester involved four virtual machines, with Kali Linux serving as our main operating system and Windows 7, Ubuntu, or Windows XP Professional serving as the target computers. Using these methods to comprehend and carry out penetration test exploits.

Exploit 1: **Creating standalone Payloads with msfvenom**

The first vulnerability I choose is one of the earliest building blocks for many of the exploits that will be discussed in the paper. This segment includes performing a reverse tcp exploit, which helps one to use Msfvenom to generate a standalone payload. To provide any background, this exploit is intended to expose a loophole on the target device and gain control of it, but it does so by exploiting a common security flaw which is the users. In order to do so, we must set up a connection that will be sent to the target machine using "*windows/meterpreter/reverse_tcp*" and then ultimately allow us, for this experiment, to serve a file payload. The setup for this exploit involves the use of Kali and Windows XP machines. Assuming that the two VMs can communicate with one another through pinging, we begin by running the "*msfvenom -p windows/meterpreter/reverse_tcp -o*" command to bind to and send the payload to the target machine, which is Windows XP. LHOST and LPORT are two variables we'll set to tell the target machine where to link back to our target machine (image1). After that, we convert the file to a specific format and then copy the payload to the Apache directory (image2). The exploit is then run by first starting the PostgreSQL database and then the Metasploit server, from which we generate a URL that the target machine can access, and then being able to run the exploit (image3) and open the file on the target machine by doing the "*set PAYLOAD windows/meterpreter/reverse_tcp*" command (image4). Finally, as the URL is entered, the file is accessed on the target computer, and depending on what the file contains, it runs whatever it was designed to run when clicked on. The best way to prevent this is to install or create an IPS and software control signatures that allow you to distinguish packet types as they move through your machine. When you set this up, you'll be able to choose whether to automatically block, watch, authorize, or quarantine packets that fit the signature. The sensor may then be added to

a firewall policy. When a packet with your custom signature is accepted by the firewall policy, the application will perform the action you specify with the packet.

Image1:

Image2:

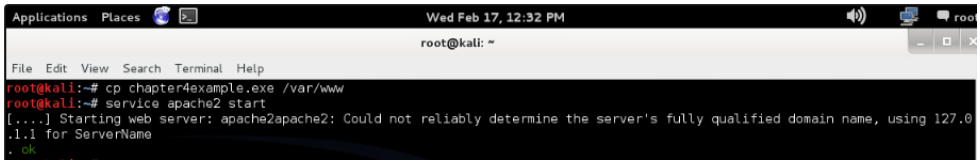


Image3:

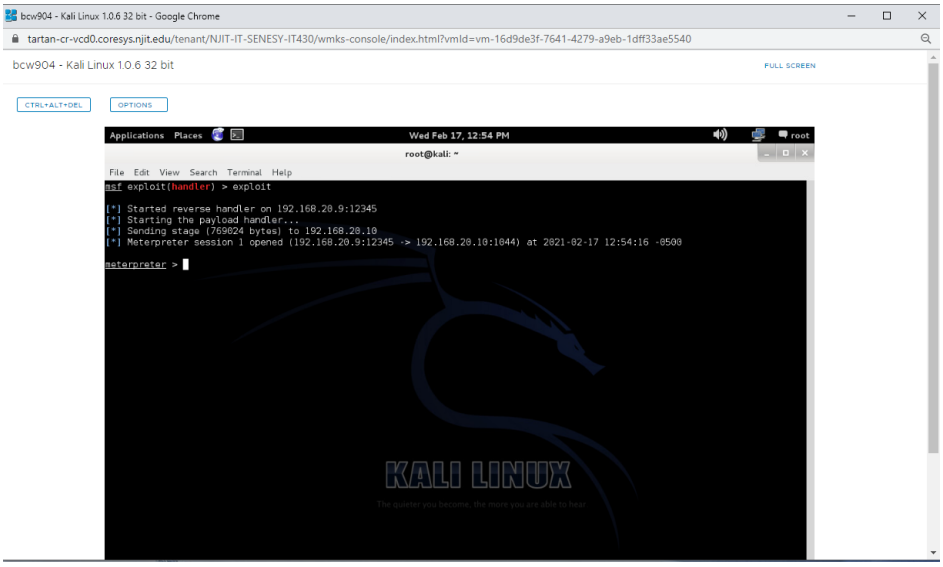
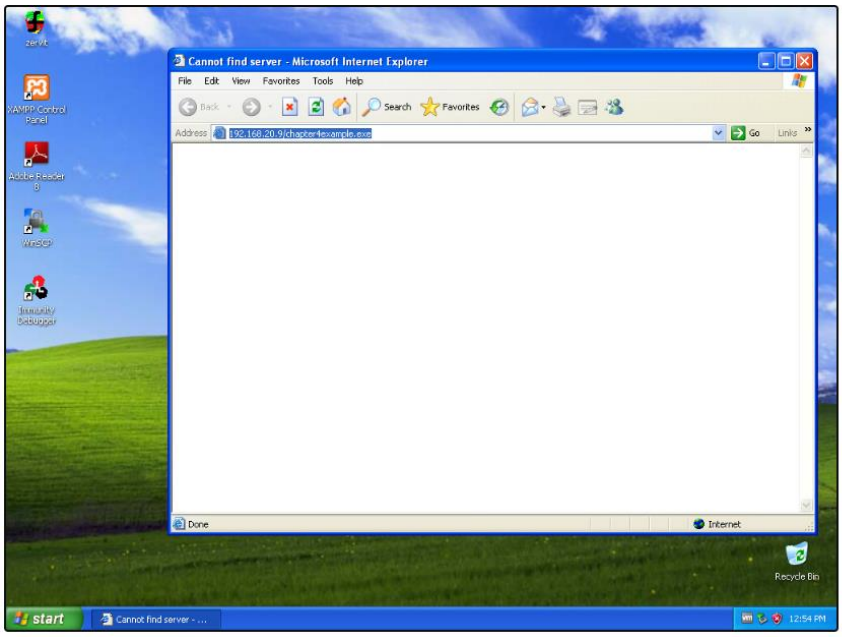


Image4:



Exploit 2: **Exploiting webdaV default Credentials**

The second vulnerability is a follow-up to Chapter 6, in which we looked at problems that could lead to failure during the exploitation process. This exploit utilizes default login credentials for the WebDAV folder used to upload files to the web server and uses XAMPP installation on our Windows XP target. This weakness helps us to use Cadaver, a command-line WebDAV client, to upload our own pages to the server. Kali and Windows XP computers are included in the configuration for this exploit. Assuming that the two VMs will connect by pinging each other. Although we can send different kinds of files we are only sending a text file for this exploit. In order to perform this exploit, we must start the metasploit default payload in order to perform what we call a post exploitation(Image1). The second step involves creating a test file that would then be sent to the target machine Windows XP. From there we will use Cadaver using the command line *"**cadaver** http://192.168.20.10/webdav"* and credentials *"**wampp:xampp**"* to authenticate with WebDAV (Image2). From there we are able to send our testing file to the target machine. On our target machine, we will be able to see the file we sent via browser by entering the ip address and the directory "*/webdav/test.txt*" (Image3). To avoid this exploit, upgrade to the most recent version of XAMPP, update the WebDAV username/password, or use a separate hosting solution. In general, if a device has default passwords, it is wise to modify them as soon as possible before anyone abuses that vulnerability.

Image1:

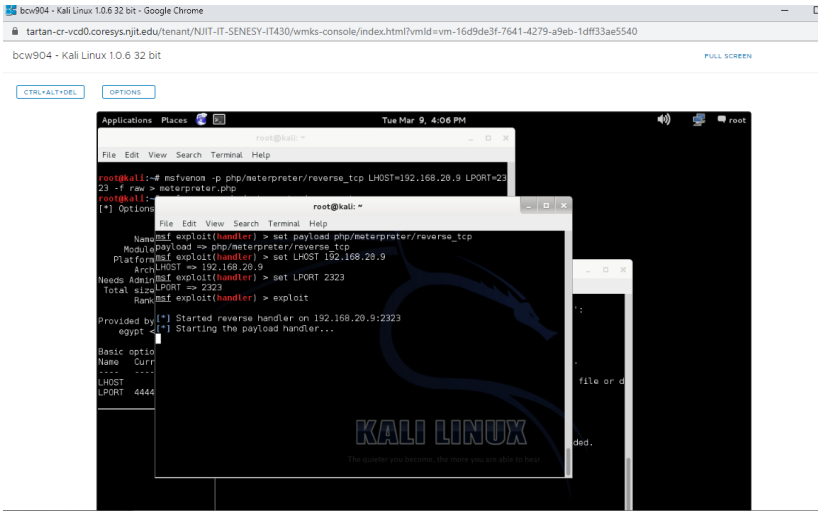


Image2:

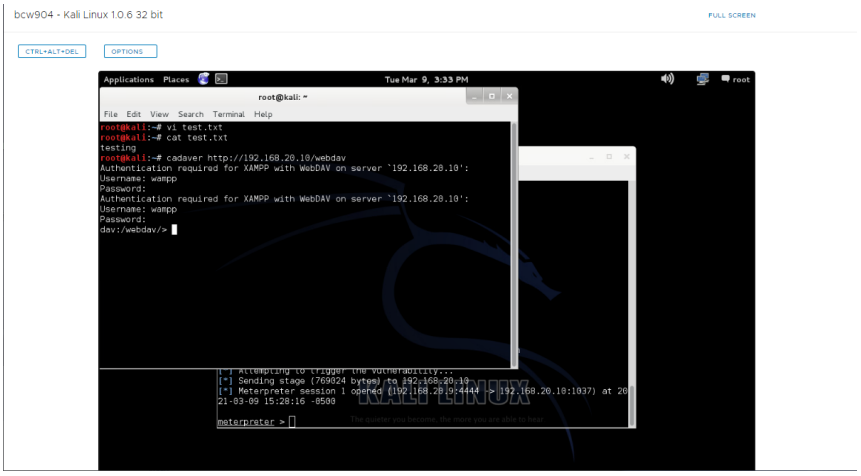
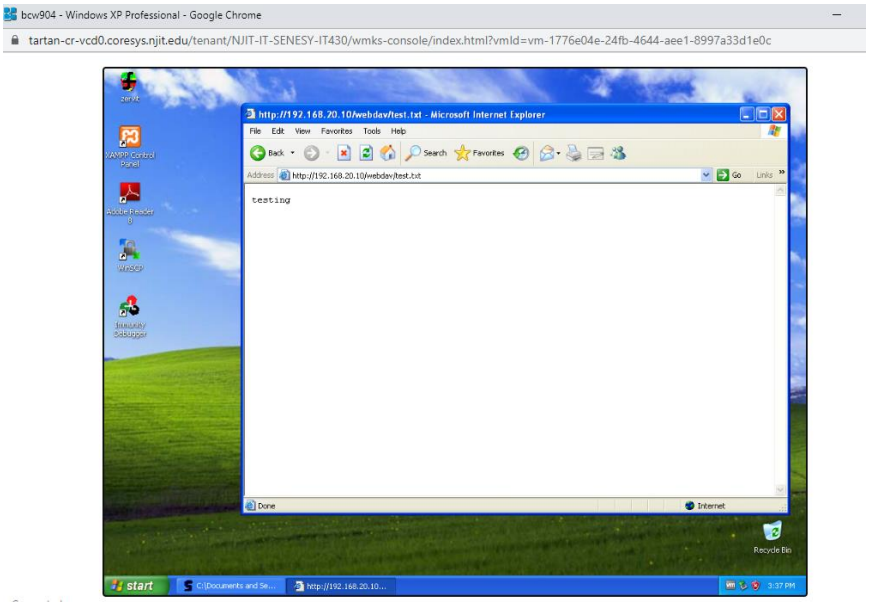


Image3:



Exploit 3: **Social Engineering: Web Attack**

The next exploit is a social engineering attack, something that is quite common and one of my favorite exploits to perform which is a Web Attack. Web Attacks are attacks that are basically done through the web. Many web attacks vary depending on the attacker(s) motive or what they're looking for from the target machine. This exploit is targeted towards what we consider a vulnerability that can never be patched which is the user. This exploit gives user(s) what appears to be a web page that they are familiar with, where they insert their personal login information on a page. However, this illusion in all reality is to gain all the user(s) personal login information. This exploit requires two Vms, the Kali machine where we will be producing the source attack and Windows XP being the target machine. On the Kali machine terminal enter on the commandline "*setoolkit*" where it displays different types of exploits you could use (Image1). In this section, we will configure Metasploit to send a Web Attack to an email. The sort of template we will use to send to the target is then determined. Following that, we configured the metasploit listener to capture our payload if someone opens the email attachment (Image2). The attack is then submitted after we set the attack webpage prototype as a gmail tab (Image3). The user will then obtain the page that we configured (Image4), and whether or when they enter their details on Kali linux, we will check the set page to see the credentials (Image5), which is a modified page from Image3. Now, mitigating this is relatively straightforward but far from simple. If your business has an IT department, it is wise to teach staff to stop clicking on links that seem to provide a lot but provide none at all. Don't click on anything that is sent to you as a general customer. The golden rule for me is that if it sounds too good to be true, it definitely is, so don't click on the link. You can even spot certain emails based on pronunciation, although this is not assured.

Image1:

bcw904 - Kali Linux 1.0.6 32 bit

FULL SCREEN

CTRL+ALT+DEL OPTIONS

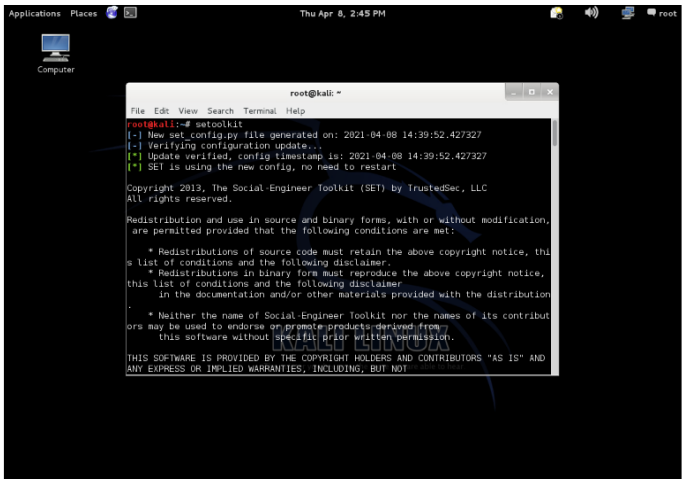


Image2:

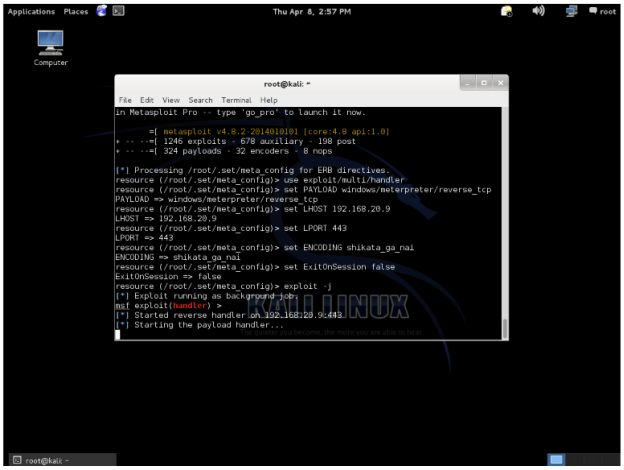


Image3:

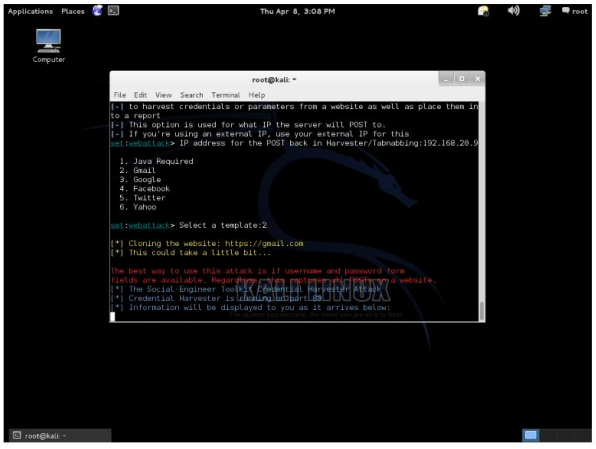


Image4:

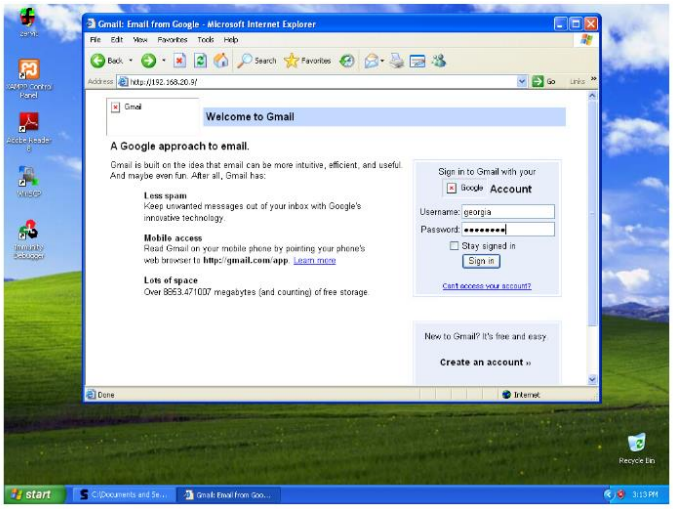
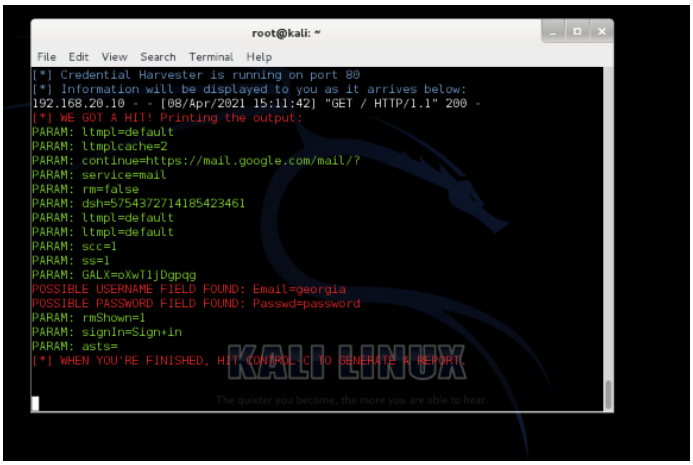


Image5:



Exploit 4: **Exploiting Open NFS Shares**

This exploit involves using Network File System (NFS) which is a distributed file system protocol that allows a user on a client computer to access files over a computer network in the same way that local storage is accessed. We may access the user's private SSH keys as well as keys used for SSH authentication using NFS under the georgia administrator account on our target computer. To get on the target computer, we must adjust the read or write sensitive file accessibility and generate our own key in order for this exploit to function or be effective. Allowing the intruder to install or add malicious apps or files to the target computer. To begin, we must mount the NFS share on the Kali Linux machine that will be used to initiate the attack (1). We must then navigate to the directory *"/tmp/mount/.ssh,"* where we can use the *"ls"* command to find the SSH keys. This is done to ensure that the keys are in the correct file directory. We can read or even modify these values after finding the keys, and we can write to the SSH file that contains a list of SSH public keys that are allowed to log in as the user.

Following the mounting and recovery of the keys, we have two options for logging into the target computer. For the first, we will use the details we have to create our own key, which will enable us to log into the Ubuntu target machine without having to enter a password (2). We can obtain access using the second alternative by copying Georgia's key to the Kali computer. On the primary machine Kali where we launch the exploit, we first copy Georgia's private and public keys to the root's.ssh directory. The identity can then be added to the authentication agent with the ssh-add command. Both methods successfully penetrate the target machine, as seen in images 3 and 4 for method 1 and method 2, respectively. You may put reasonable constraints on all NFS shares or ban NFS from outside access to protect yourself from this exploit. Furthermore, do not use the service in its default configuration. This might weaken the whole machine. An attacker with root privileges on the compromised computer might use it as a jumping off point for a network attack, resulting in a large vulnerability.

Image1:

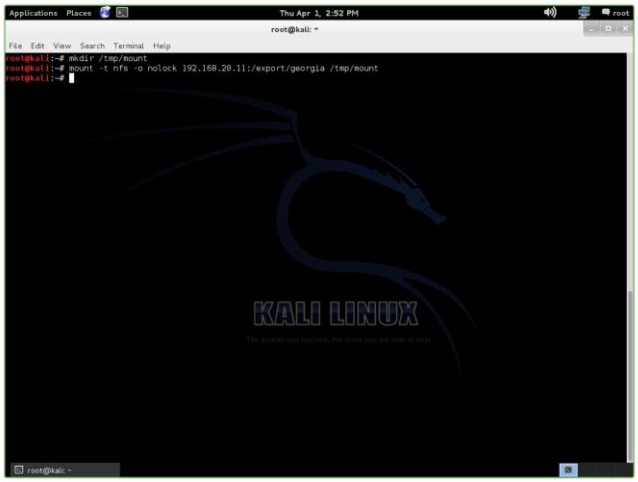


Image2:

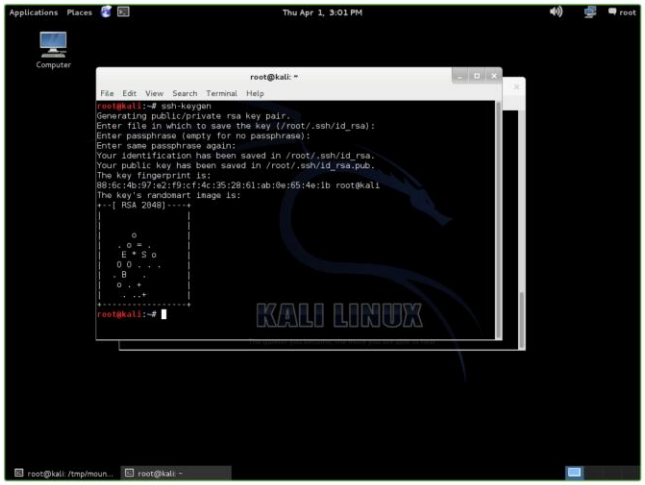


Image3:

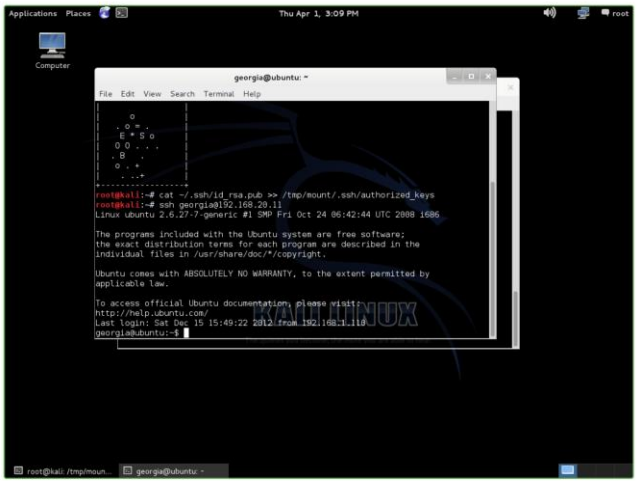


Image4:

Exploit 5: **Encrypting Executables with Hyperion**

The aim of this exploit is to get our payload past some anti-virus software, such as Windows Security Essentials. A standard payload can be quickly detected by protection tools, so one way to take advantage of this condition is to encrypt the payload such that it bypasses all security policies and enters the filesystem directly. The drawback of the protection software is that it fails to recognise the program as a poisoned payload and therefore allows it to run. Hyperion is a platform that enables us to do so, and it has a high performance rate due to the use of AES encryption. The payload is created with msfvenom to begin the lab. The msfvenom command will output the module, the computer that is carrying out the attack, and LHOST to a register. Then we'll go to the Hyperion directory and run Hyperion with the wine program from there. We can now encrypt the file and use it as a poison payload. Now that the poison payload has been developed, we must upload it to Windows 7, which we do by copying the “bypasshyperion.exe” file to our *“/var/www”* directory and restarting the apache operation. Then, on the Windows 7

desktop, make sure the Windows Security Essentials program is running before attempting to import the *“bypasshyperion.exe”* file from *"http://192.168.20.9/bypasshyperion.exe."* (Image1). The user would be able to download the poisoned payload without activating Windows Security Essentials as a result of this test (Image2). You should use a separate antivirus program that gets modified virus definitions on a daily basis to protect yourself against this exploit, which will potentially prevent you from installing something onto your phone. Users may also protect themselves against this form of attack by not uploading suspicious files that they may or may not know.

Image1:

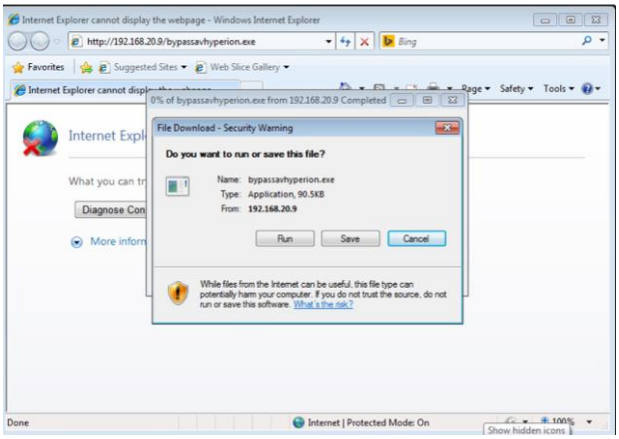


Image2: