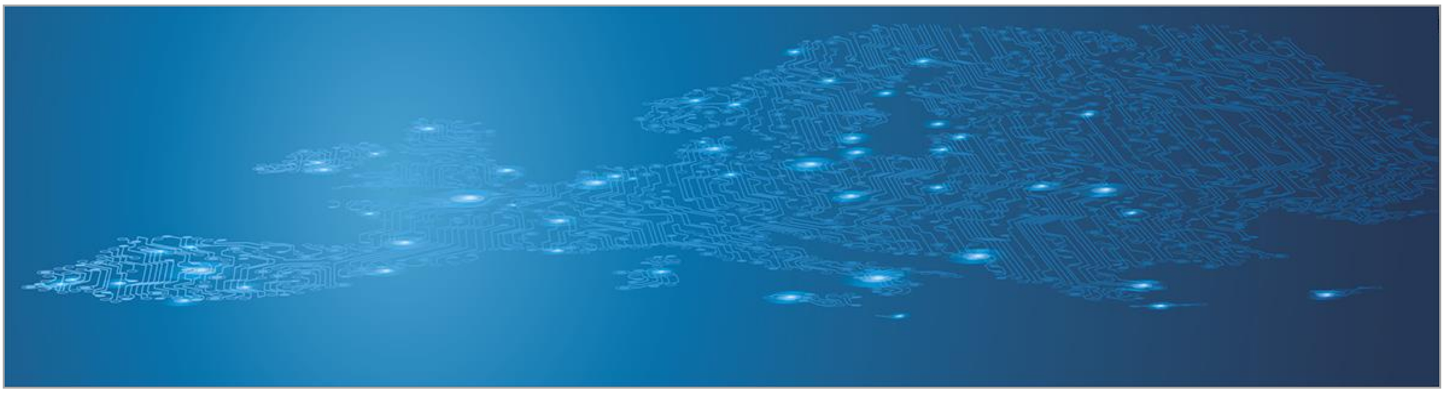


Building artifact handling and analysis environment

Artifact analysis training material

November 2014

European Union Agency for Network and Information Security www.enisa.europa.eu

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Authors

This document was created by Lauri Palkmets, Cosmin Ciobanu, Yonas Leguesse, and Christos Sidiropoulos in consultation with DFN-CERT Services[1] (Germany), ComCERT[2] (Poland), and S-CURE[3] (The Netherlands).

Contact

For contacting the authors please use cert-relations@enisa.europa.eu

For media enquires about this paper, please use press@enisa.europa.eu

Acknowledgements

ENISA wants to thank all institutions and persons who contributed to this document. A special 'Thank You' goes to Todor Dragostinov from ESMIS, Bulgaria.

[1] Klaus Möller, and Mirko Wollenberg

[2] Mirosław Maj, Tomasz Chlebowski, Krystian Kochanowski, Dawid Osojca, Paweł Weżgowiec, and Adam Ziaja

[3] Michael Potter, Alan Robinson, and Don Stikvoort

Legal notice

Notice must be taken that this publication represents the views and interpretations of the authors and editors, unless stated otherwise. This publication should not be construed to be a legal action of ENISA or the ENISA bodies unless adopted pursuant to the Regulation (EU) No 526/2013. This publication does not necessarily represent state-of the-art and ENISA may update it from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources including external websites referenced in this publication.

This publication is intended for information purposes only. It must be accessible free of charge. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

Copyright Notice

Table of Contents

Main Objective	The main objective of this exercise is to teach students how to create a safe and useful malware laboratory based on best practices for the analysis of suspicious files. This exercise presents practical aspects of configuring artifact analysis environment, which will be used throughout the artifact handling training course during the next few days. Participants will become familiar with the threats posed by artifact analysis, as well as learn good practices to set up an environment. In the practical part of the exercise, the participants will set up, configure and deploy an environment in a virtualized environment consisting of: **Gateway "Styx" (Ubuntu 14.04 server)** –traffic from other virtual machines will be routed through this server. The gateway will provide services and connectivity to the machines where artifact analysis is performed. **Analysis environment "Winbox" (Windows 7)** – artifacts may be run/executed and their activity monitored on this machine.	
Targeted Audience	CERT staff involved in the process of incident handling, especially those responsible for detection of new threats related directly to the CERT customers.	
Total duration	6-7 hours	
Time Schedule	**Introduction to the exercise**	0.5 hour
	Task 1: VirtualBox configuration	0.5 hour
	Task 2: Winbox configuration	1.5 hours
	Task 3: Styx configuration	3.5 hours
	Task 4: Summary	0.5 hour
Frequency	Every time a new member joins the team.	

1 Introduction to the exercise

Artifact analysis is considered by many the most interesting experience a CERT/CSIRT member can have at work. It might be however one of the most boring ones when repeatedly analysing yet another malware family variant. The task can be also one of the most time consuming when advanced malicious code is under analysis. The excitement of discovering new malicious behaviour can easily be followed by despair after analysing artifacts without any progress. There are several aspects to consider before constructing an artifact analysis environment.

1.1 Safe and secure operations

Safety of operation is of paramount importance for any artifact analysis laboratory. In such an environment, handling malicious code is routine, so the lab design and practices must ensure that malicious code cannot escape to the outside world, doing harm to the rest of organisation. It should be considered carefully how the artifact acquisition process is handled, how malware execution and testing are carried out, and also how malware storage is designed and maintained. The environment should be made as error resistant as possible. An ideal solution would be a completely disconnected site with all artifacts stored only for the time needed for analysis and kept in an encrypted form, so that no accidental execution is possible. While it's not completely guaranteed that such an approach can guarantee safety, it is obvious that this approach isn't practical. Samples are stored in repositories for further analyses and comparison. Internet connectivity is also useful as malware would in many cases download updates or its actual payload only when it is connected to its command and control server. It is always important to note that malicious code could possibly be able to self-replicate or propagate through some unpatched holes, such as zero day vulnerabilities.

The best approach in terms of safety is to be completely isolated from the outside world by default but retain the possibility of going online. However, going online brings the risk of being noticed. Malware authors would know that someone is analysing their code (or at least running it) if the code had and used a call-home function. In most cases it is advised to spoof the analysis lab identity. It is advised to use anonymisation techniques such as tunnelling all the traffic from the lab through the Tor network or VPN connections. While this wouldn't prevent the malware authors from knowing someone is running their code they wouldn't know exactly who it is. This increases the lab security and prevents a possible "strike back".

Malware analysis data should be treated as confidential information. There is a very popular site, virustotal.com[4] used by many security researchers to see if a sample is malicious, which antivirus engines detect it and what malware it is. However by submitting the sample to analysis at Virustotal, the information is released that a sample with a certain cryptographic hash was submitted for analysis. If the authors are watching VT, they immediately know that someone has their code and is analysing it. This service is given as an example as it is very popular, but there are many similar services utilised by the security community.

Last but not least, the security of the environment must be ensured to prevent any leak of malicious code, the techniques and tools used by security researchers, and in many cases, the researchers' personal details.

[4] https://www.virustotal.com/

1.2 Architectural considerations: Physical or Virtual

Virtualisation technologies provide a great deal of flexibility when analysing malware. They give the ability to run multiple operating systems, multiple versions and patch sets as well as different combinations of third party software installed. The ability to create virtual machine snapshots allows the malware analyst to replay test scenarios with same configuration settings as well as conduct offline memory analyses when the snapshot is created containing a memory dump. There are software packages that will be described later in this document which take full advantage of virtual machine operations to create an automatic analysis environment.

However, malware authors are fully aware of possibilities and malware often contains code that can detect virtualisation environments and debuggers and then refrain from performing malicious tasks when run under such conditions[5]. There are techniques to hide the fact of running code in virtual environment as well as techniques of emulating user interaction, but this is an arms race and it should expected that in some cases when dealing with new and advanced artifacts a physical machine may be necessary. Lab design should allow putting a physical machine in the infrastructure if needed for proper analysis.

In this exercise a basic virtualised version of an artifact analysis environment is described, that could be considered secure and flexible.

[5] See one of the problem descriptions: http://www.darkreading.com/analytics/security-monitoring/attackers-toolbox-makes-malware-detection-more-difficult/d/d-id/1140283?

2 Introduction to the analysis environment

2.1 Architecture overview

The analysis environment is a system consisting of virtual machines and an isolated virtual network. Its role is to allow users to perform artifacts analysis and then to examine collected results. The analysis environment consists of two virtual machines: Styx and Winbox, which are used throughout the exercises. The analysis environment is built in such a way as to allow it to be extended with additional virtual machines in the future.

Styx is an Ubuntu Server 14.04 (32-bit) virtual machine. Its main role is to be a lab gateway between the Winbox machine and the Internet. In normal operation, all network traffic from the Winbox machine is blocked from going to the Internet and instead is redirected to network simulator (INetSim). In a second operational mode, all network traffic is redirected through a virtual private network (VPN) or the onion router (Tor) network. Styx also serves as a Cuckoo Sandbox server, which is used in automatic analyses. During the analyses, all network traffic is captured and checked against Snort intrusion detection system (IDS) signatures.

Winbox is a Windows 7 (32-bit) virtual machine (VM) where the actual artifact analyses take place. This VM would contain two snapshots. One is used for automatic analyses with Cuckoo Sandbox and consists only of a minimal toolset. The second snapshot is used in manual analysis (static and dynamic) and consists of various tools used in artifact analysis (debuggers, disassemblers, hex editors, portable executable (PE) viewers, etc.).

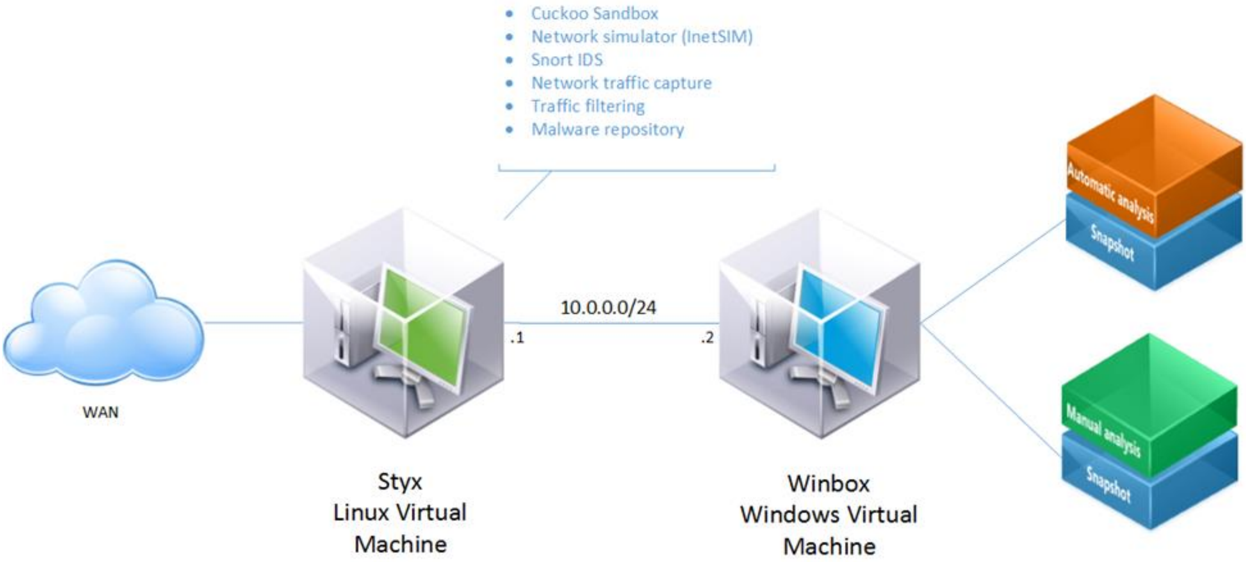


Figure 1: Artifact analysis environment build-up

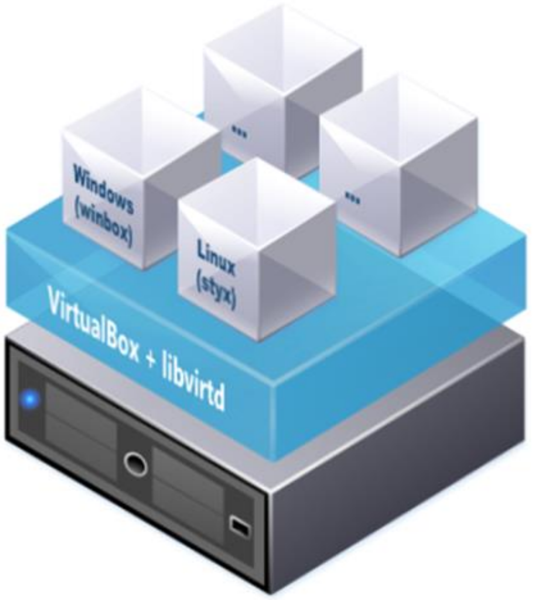


Figure 2: Artifact analysis environment logical structure

3 Preparing virtual images

3.1 Import Virtual Machines

In this step, users import the provided Styx and Winbox virtual machine images, which are bare installations of Ubuntu Server 14.04 and Windows 7. Before importing virtual machines, start VirtualBox **the *root* user**.

3.2 VirtualBox network configuration

Imported virtual machines don't have a proper network configuration in VirtualBox. Users must add virtual interfaces to both Styx and Winbox according to the network graph – making Winbox operate in the isolated lab network and Styx be a lab gateway.

At the beginning, the user must create an additional network – vboxnet0. Vboxnet0 will be used as a host-only connection between the Host machine and the Styx virtual machine.

To create the Host-only network, select File->Preferences (CTRL+G), select Network and click on the Host-only Networks tab. Next click on the icon on the right side to create a new network called *vboxnet0*.

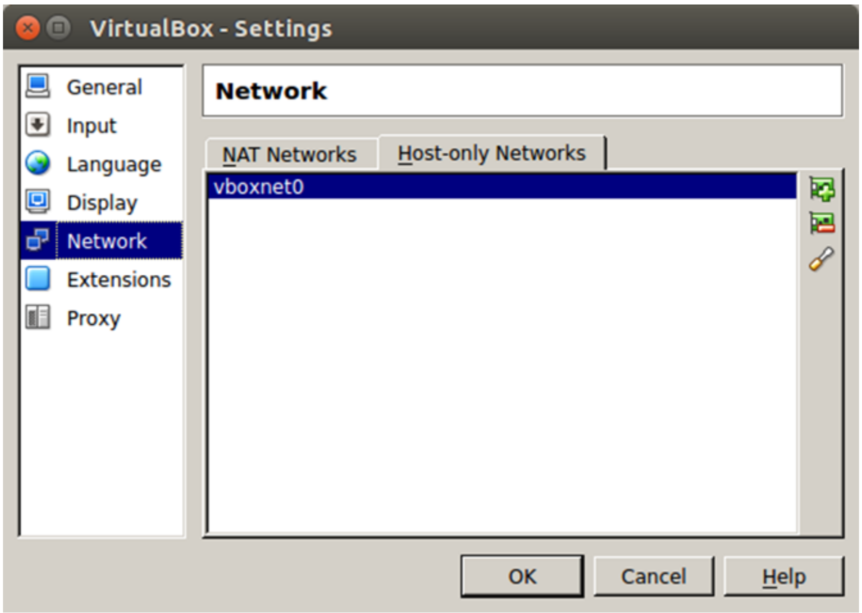


Figure 3: Host-only Networks

In the next step, participants must configure the network interfaces for the *Styx* Virtual Machine. The *Styx* machine requires three network interfaces: Bridged Adapter, Internal network and Host-only Adapter. The Windows machine (Winbox) will be used with only one network interface for direct connection to the *Styx* virtual machine. *Styx* will be used as gateway for *Winbox*.

To configure the network interfaces, right click on the *Styx* machine and choose *Settings* from context menu.

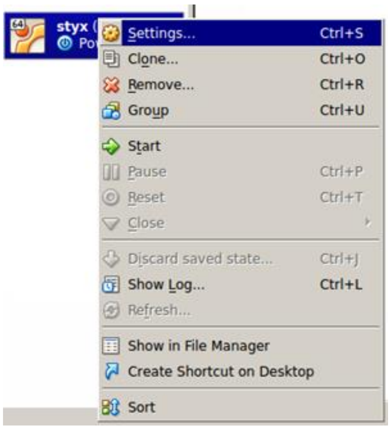
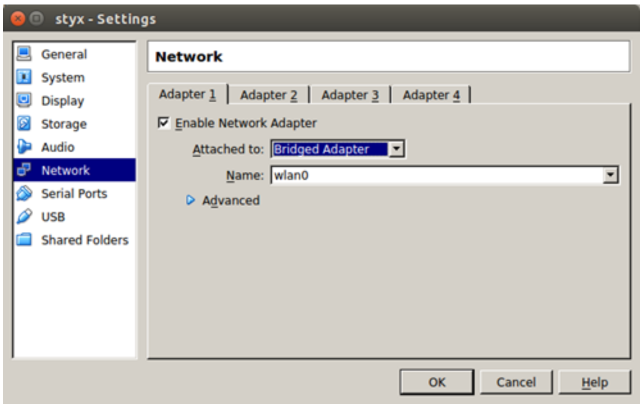


Figure 4: Screenshot Styx context menu

Adapter 1 must be set as the Bridged Adapter (Figure 5) which will be used for the Internet connection, Adapter 2 as *intnet* (Figure 6) and the third adapter as the Host-only adapter (Figure 7).



Network

☑ Enable Network Adapter

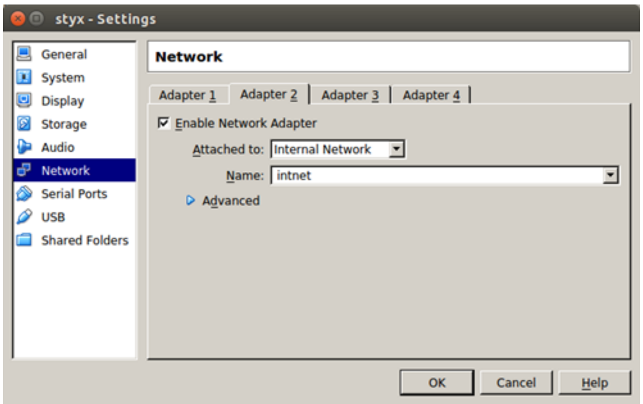
Attached to: Bridged Adapter

Name: wlan0

Advanced

Figure 5: Adapter 1 configuration

Adapter 1 will be used for Internet network connectivity for the Styx virtual machine.



Network

☑ Enable Network Adapter

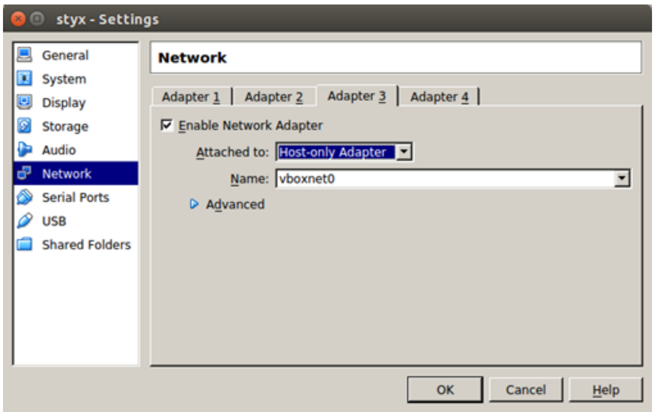
Attached to: Internal Network

Name: intnet

Advanced

Figure 6: Adapter 2 configuration

The second adapter will be used only for direct connection between the *Styx* and *Winbox* machines. All network traffic from Styx will be directed through this interface. This interface should have IP 10.0.0.1, which will be set later to this interface.



Network

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: vboxnet0

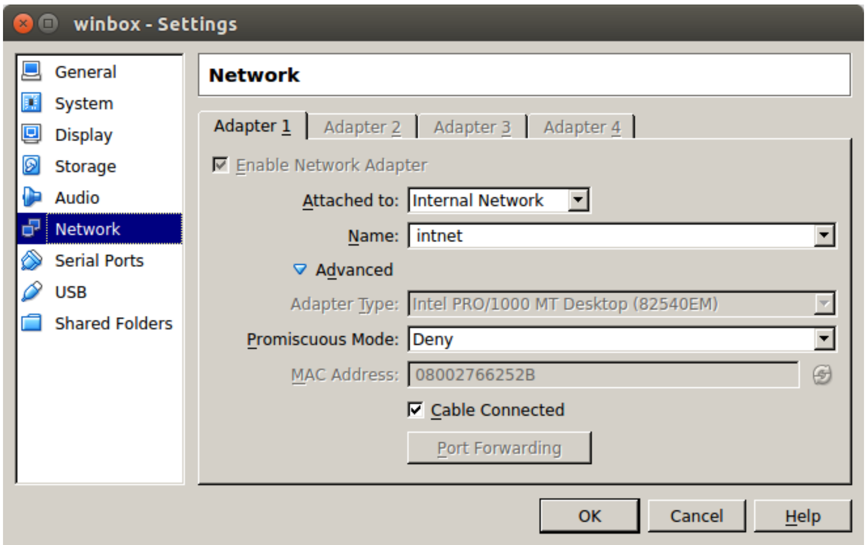
Advanced

Figure 7: Adapter 3 configuration

The last adapter (Adapter 3) will be used for the Cuckoo sandbox software. It will be used for traffic between the Cuckoo server (Styx) and the Cuckoo agent (Winbox). This interface should have IP 192.168.56.1, which will be set in the next steps.

In the next step, participants must configure the network interface for the Windows (Winbox) machine. As in the previous example of Styx configuration, right click on the Winbox VM in the

VirtualBox application and choose Setting from context menu. Choose the *Network* page and set *Adapter 1* as *Internal Network* (Figure 8). This will be the only interface for network access.



☑ Enable Network Adapter
Attached to: Internal Network
Name: intnet
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Deny
MAC Address: 08002766252B
☑ Cable Connected

Figure 8. Winbox network adapters in VirtualBox.

3.3 Role of Snapshots in malware analysis

Snapshots can be created or restored at any time. To restore a snapshot, select a specific machine and click the Snapshot button on the right corner of the VirtualBox window (Figure 9). On the main panel, a list of all available snapshots is displayed (Figure 10).

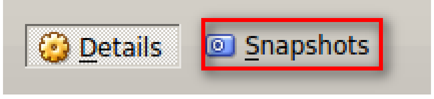


Figure 9. Snapshot button in VirtualBox.

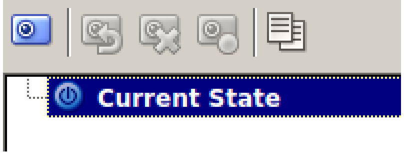


Figure 10. Lists of snapshots.

To restore a snapshot, select snapshot from the list, right click and select “Restore Snapshot” from the context menu (Figure 11).

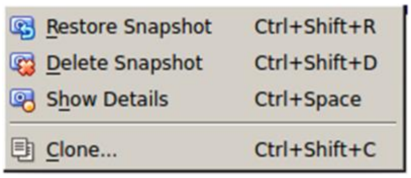


Figure 11: Snapshot context menu

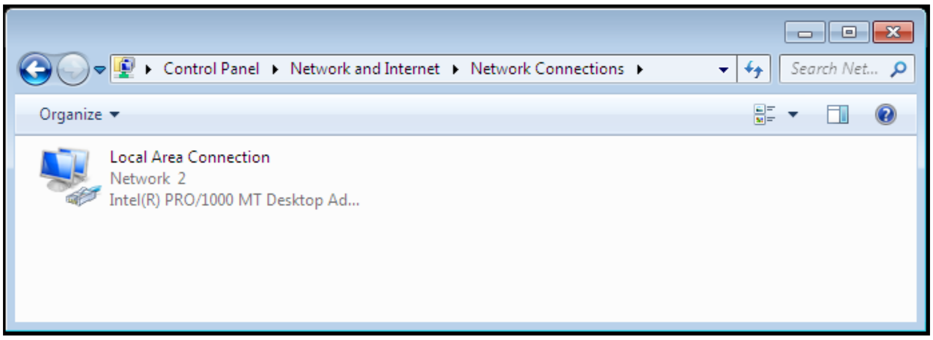
To create a snapshot of current machine, press the icon from the toolbar.

4 Configuring Winbox virtual machine

In this step, participants perform the initial configuration of the Windows analysis machine.

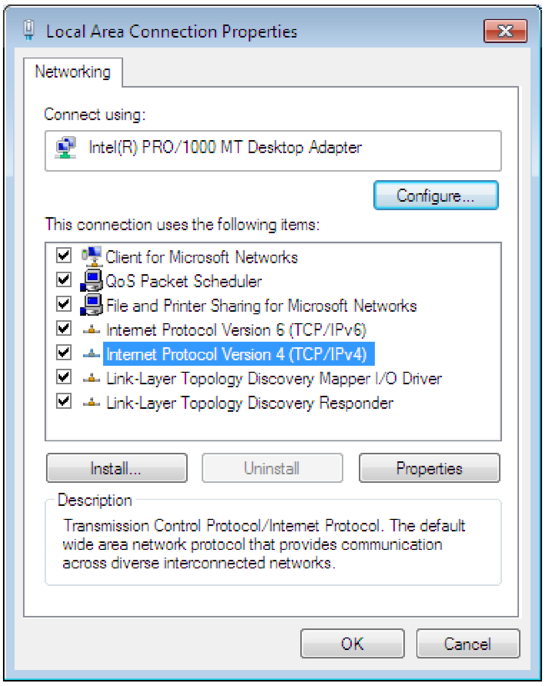
4.1 Initial configuration

Start the Winbox machine in VirtualBox. First, participants must set an IP address. The IP address for the Winbox machine must be set manually. To set a static IP address, click Start, Control Panel, View network status, and under the Network and Internet category click Change adapter settings. Only one network interface should be listed (Figure 12). To configure it, right click on it and select Properties from the context menu. Choose Internet Protocol Version 4 (TCP/IPv4) and click the Properties button (Figure 13).


Control Panel ▸ Network and Internet ▸ Network Connections ▸
Search Net...
Organize ▾

Local Area Connection
Network 2
Intel(R) PRO/1000 MT Desktop Ad...

Figure 12. Network interfaces in Winbox machine



Connect using:
Intel(R) PRO/1000 MT Desktop Adapter

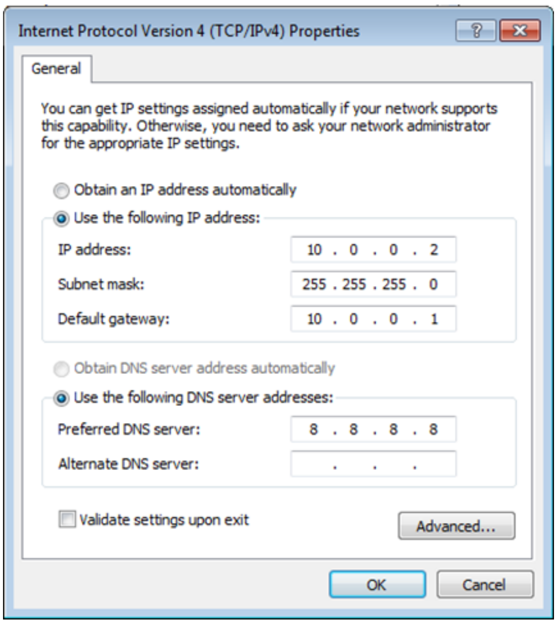
This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Figure 13. Network interface protocols.

The network interface in the Winbox machine should have a static IP address. For communication with the Styx VM, Winbox should have an IP address of 10.0.0.2, 255.255.255.0 subnet mask and 10.0.0.1 as the gateway (Figure 14). The DNS server should be set to 8.8.8.8.



Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 10 . 0 . 0 . 2
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 10 . 0 . 0 . 1

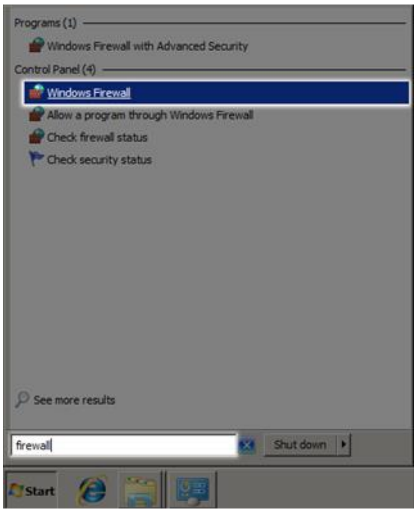
○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: 8 . 8 . 8 . 8
Alternate DNS server: . . .

□ Validate settings upon exit

Figure 14: Winbox machine Network Configuration

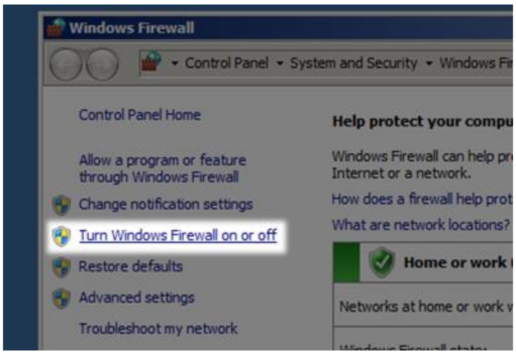
Winbox must have non-blocked access to the Internet, so the operating system firewall should be disabled. To disable the firewall, click the Start menu and type firewall in the search box at the bottom of the menu. In the results shown, click on Windows Firewall (Figure 23).



firewall

Figure 15: Windows Firewall in Start menu

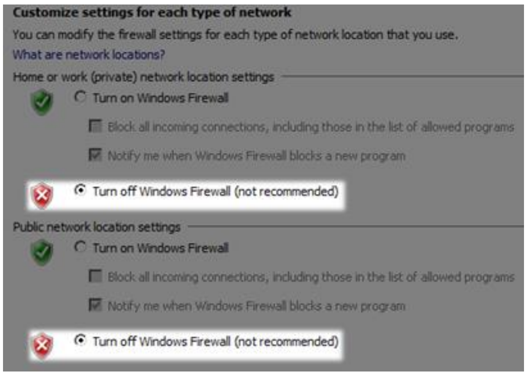
Next, click on “Turn Windows Firewall on or off” on the left side of the window (Figure 24).



Windows Firewall
Control Panel
System and Security
Windows Fi
Control Panel Home
Allow a program or feature through Windows Firewall
Change notification settings
Turn Windows Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network
Help protect your compu
Windows Firewall can help pr
Internet or a network.
How does a firewall help prot
What are network locations?
Home or work
Networks at home or work v
Windows Firewall state:

Figure 16: Turn the Windows Firewall on or off

In the last step, select “Turn off Windows Firewall (not recommended)” for both network types (Home/work and public network) (Figure 24).



Customize settings for each type of network

You can modify the firewall settings for each type of network location that you use.

What are network locations?

Home or work (private) network location settings

○ Turn on Windows Firewall

□ Block all incoming connections, including those in the list of allowed programs

☑ Notify me when Windows Firewall blocks a new program

● Turn off Windows Firewall (not recommended)

Public network location settings

○ Turn on Windows Firewall

□ Block all incoming connections, including those in the list of allowed programs

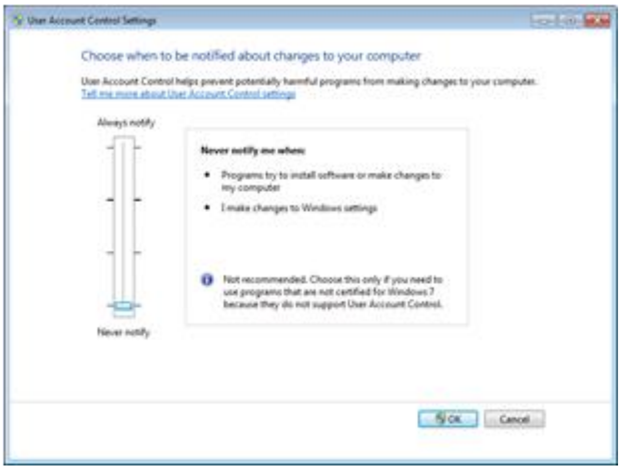
☑ Notify me when Windows Firewall blocks a new program

● Turn off Windows Firewall (not recommended)

Figure 17: Disabling Firewall for Home/work and Public network

4.1.1 Disable User Account Control

User Account Control (UAC)[6] is a feature in Windows that can help you stay in control of your computer by informing you when a program wants to make a change requiring administrator-level permission. In the Winbox VM machine, participants should disable UAC. In Control Panel, choose “Change User Account Control settings” and set slider to “Never notify” (Figure 18). After making these changes, shutdown the Winbox machine.



User Account Control Settings

Choose when to be notified about changes to your computer

User Account Control helps prevent potentially harmful programs from making changes to your computer.

Tell me more about User Account Control settings

Never notify me when:

- Programs try to install software or make changes to my computer
- I make changes to Windows settings

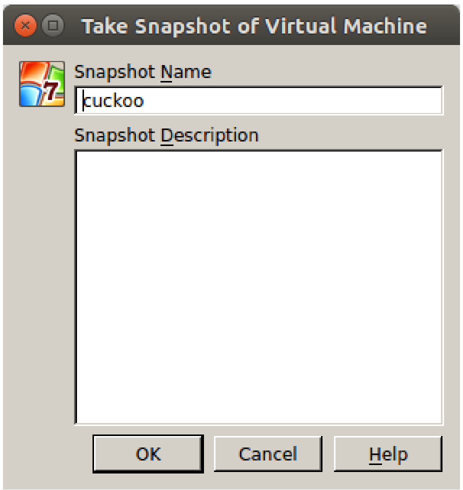
Not recommended. Choose this only if you need to use programs that are not certified for Windows 7 because they do not support User Account Control.

Figure 18: Disabling User Account Control in Winbox

4.2 Create snapshots

Creating snapshots of the virtual machine is important in this phase. Participants must now **create two snapshots of the Winbox machine** – the first one will be used for automatic malware analysis with Cuckoo and the second one will be used for static and dynamic analysis by participants. Name the first snapshot for automatic malware analysis “cuckoo”. Name the second snapshot “winbox-clean” (Figure 19).

[6] http://windows.microsoft.com/en-us/windows/what-is-user-account-control#1TC=windows-7

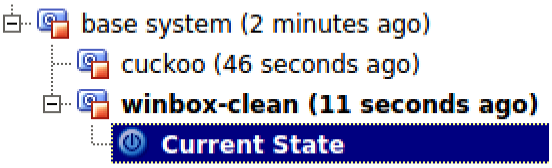


Snapshot Name: cuckoo

Snapshot Description:

Figure 19. Create snapshot of Winbox machine called cuckoo.

All snapshots for the Winbox machine are shown in Figure 20. The next steps of this exercise will be performed on the *winbox-clean* snapshot.



```
base system (2 minutes ago)
    cuckoo (46 seconds ago)
    winbox-clean (11 seconds ago)
        Current State
```

Figure 20. All snapshots in Winbox virtual machine.

4.3 Tools for artifact analysis

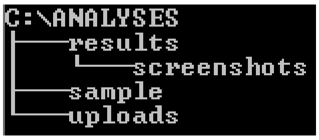
During the next steps, participants will install all required tools to the Windows (winbox-clean) snapshot. Point students towards the archive called *winbox_tools.zip*. Unpack the contents of this archive to the *C:\tools* directory and install all the required tools from the “*Install version”* subdirectory.

In the second subdirectory named “Portable version” there are the tools that don’t require installation.

4.3.1 Create directory for results of malware analysis

During this step, participants will create a set of directories. The directories will be used to store the results of malware analysis. The Styx machine will also use the content of these directories for further analysis.

Create the directory C:\analyses with subdirectories C:\analyses\results, C:\analyses\sample C:\analyses\uploads. In C:\analyses\results directory, create a subdirectory called screenshots (Figure 21).



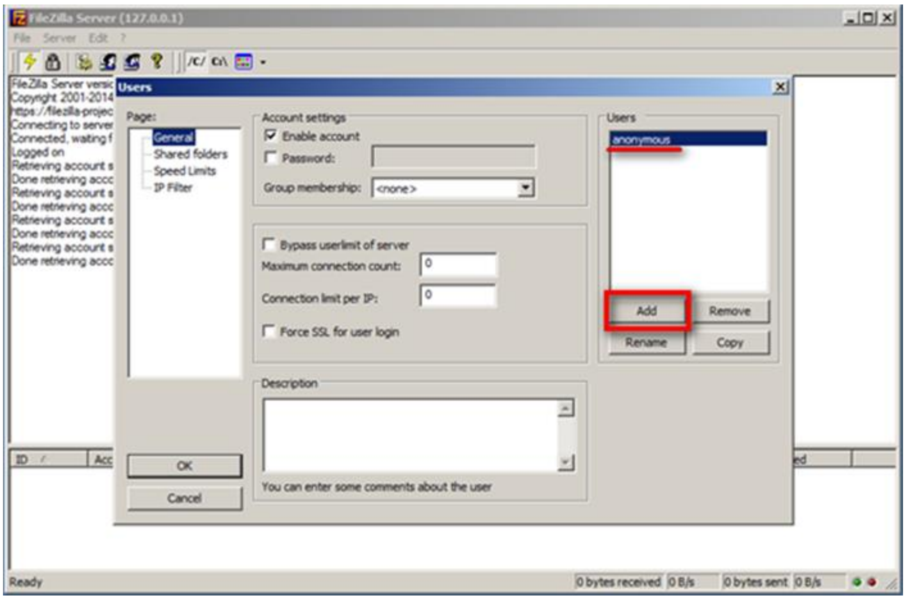
```
C:\ANALYSES
├───results
│   └───screenshots
├───sample
└───uploads
```

Figure 21. Directory tree of c:\analyses dir.

4.3.2 Installing FTP server

To share files between the Styx and Winbox machines (sending samples, retrieving manual analysis results) the FTP protocol will be used. An FTP server will be installed on the Windows machine and scripts running on Styx VM will perform synchronisation.

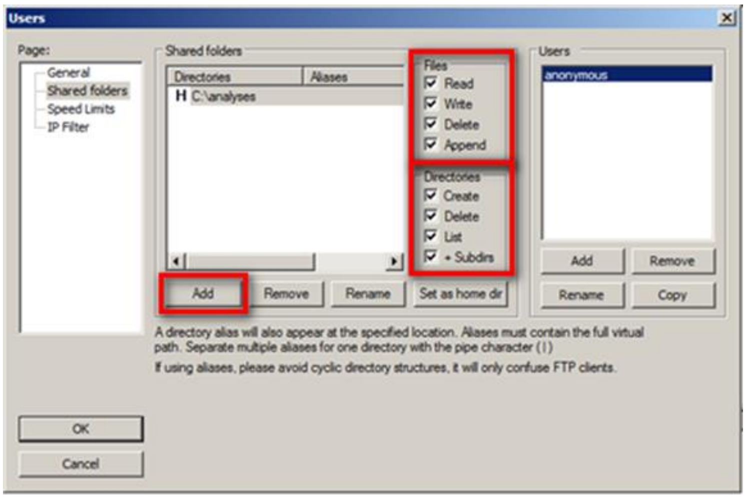
After *FileZilla Server* installation, click on the *Users icon* in the toolbar and add a new user by clicking the *Add* button. Create user *anonymous* without any password. (Figure 22)



☑ Enable account
□ Password:
Group membership: <none>
□ Bypass userlimit of server
Maximum connection count: 0
Connection limit per IP: 0
□ Force SSL for user login
You can enter some comments about the user
0 bytes received 0 B/s 0 bytes sent 0 B/s

Figure 22: Users in ftp service

In the next step, select the Shared folders page and click the Add button. Add the C:\analyses directory with all permissions (Figure 23).



Directories	Aliases
H C:\analyses	

Files
☑ Read
☑ Write
☑ Delete
☑ Append

Directories
☑ Create
☑ Delete
☑ List
☑ + Subdirs

A directory alias will also appear at the specified location. Aliases must contain the full virtual path. Separate multiple aliases for one directory with the pipe character (|)
If using aliases, please avoid cyclic directory structures, it will only confuse FTP clients.

Figure 23: Directory c:\analyses with all permissions

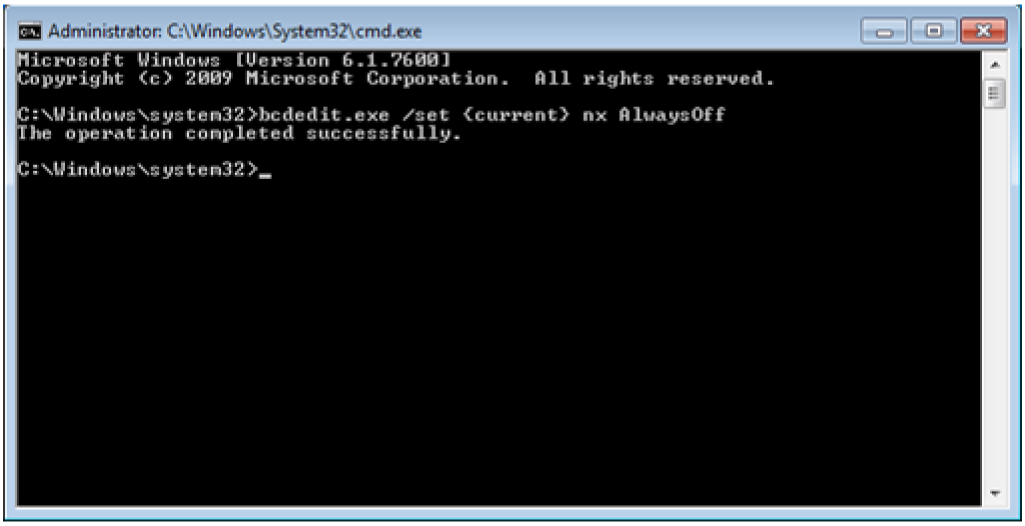
4.3.3 Disable ASLR

Address space layout randomization (ASLR) is a computer security technique to protect against buffer overflow attacks. In order to prevent an attacker from reliably jumping to a particular exploited function in memory (for example), ASLR randomly arranges the locations of key data areas of a program, including the base of the executable and the positions of the stack, heap, and libraries, in a process's address space[7][8]. In this exercise, ASLR in the Winbox virtual machine should be disabled.

To disable ASLR, run cmd.exe (as Administrator) and use bcdedit.exe as on Figure 24.

Disable ASLR

```
bcdedit.exe /set {current} nx AlwaysOff
```



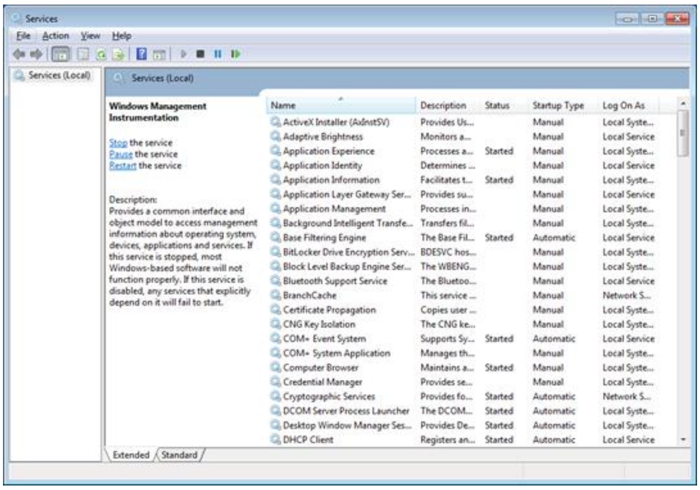
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>bcdedit.exe /set {current} nx AlwaysOff
The operation completed successfully.

C:\Windows\system32>_
```

Figure 24: Bcdedit.exe command

4.3.4 Disable services

The Simple Service Discovery Protocol (SSDP) is a network protocol based on the Internet Protocol Suite (TCP/IP) to advertise and discover network services and presence information. It might generate network traffic which may confuse malware network analysis.

To open Services, press the Start key and type the services.msc command. Find the SSDP Discovery service (Figure 25).



Windows Management Instrumentation

Stop the service
Pause the service
Restart the service

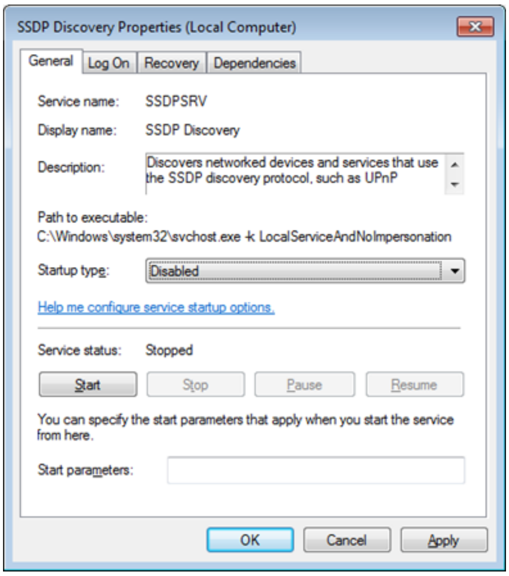
Description:
Provides a common interface and object model to access management information about operating system, devices, applications and services. If this service is stopped, most Windows-based software will not function properly. If this service is disabled, any services that explicitly depend on it will fail to start.

Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Manual	Local Syste...
Adaptive Brightness	Monitors a...		Manual	Local Service
Application Experience	Processes a...	Started	Manual	Local Syste...
Application Identity	Determines ...		Manual	Local Service
Application Information	Facilitates t...	Started	Manual	Local Syste...
Application Layer Gateway Ser...	Provides su...		Manual	Local Service
Application Management	Processes in...		Manual	Local Syste...
Background Intelligent Transfe...	Transfers fil...		Manual	Local Syste...
Base Filtering Engine	The Base Fil...	Started	Automatic	Local Service
BitLocker Drive Encryption Serv...	BDESVC hos...		Manual	Local Syste...
Block Level Backup Engine Ser...	The WBENG...		Manual	Local Syste...
Bluetooth Support Service	The Bluetoo...		Manual	Local Service
BranchCache	This service ...		Manual	Network S...
Certificate Propagation	Copies user ...		Manual	Local Syste...
CNG Key Isolation	The CNG ke...		Manual	Local Syste...
COM+ Event System	Supports Sy...	Started	Automatic	Local Service
COM+ System Application	Manages th...		Manual	Local Syste...
Computer Browser	Maintains a...	Started	Manual	Local Syste...
Credential Manager	Provides se...		Manual	Local Syste...
Cryptographic Services	Provides fo...	Started	Automatic	Network S...
DCOM Server Process Launcher	The DCOM...	Started	Automatic	Local Syste...
Desktop Window Manager Ses...	Provides De...	Started	Automatic	Local Syste...
DHCP Client	Registers an...	Started	Automatic	Local Service

Figure 25: Services in Windows Operating System

Press the stop button, set Startup type to Disabled (Figure 26) and apply changes.

[7] http://en.wikipedia.org/wiki/Address_space_layout_randomization
[8] http://www.microsoft.com/security/sir/strategy/default.aspx#!section_3_3



SSDP Discovery Properties (Local Computer)

Service name: SSDPSRV

Display name: SSDP Discovery

Description: Discovers networked devices and services that use the SSDP discovery protocol, such as UPnP

Path to executable:
C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation

Startup type: Disabled

Help me configure service startup options.

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

Figure 26: Properties of SSDP Discovery

4.3.5 Show hidden files

Malware uploaded to the Winbox VM may have hidden attributes. To see files in Windows with the hidden flag, open folder and search options (Figure 27) in the View tab select Show hidden files, folders and drives (Figure 28).

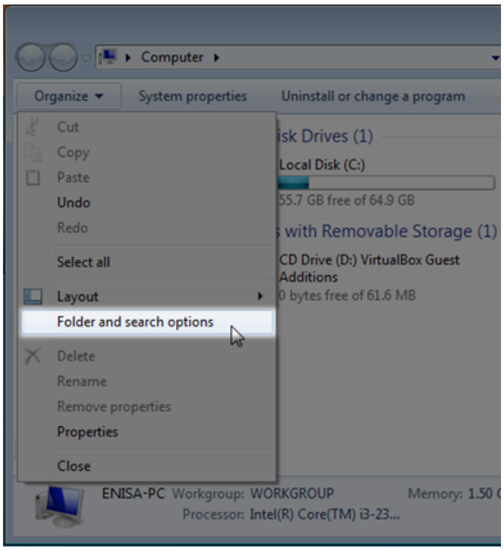
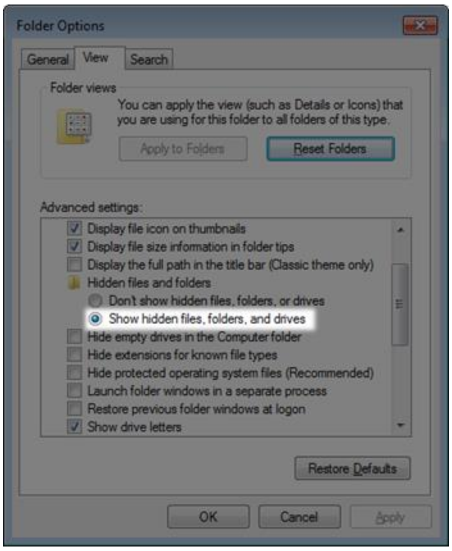


Figure 27: Folder and search options in My Computer



Folder Options

General | View | Search

Folder views

You can apply the view (such as Details or Icons) that you are using for this folder to all folders of this type.

Apply to Folders | Reset Folders

Advanced settings:

- ☑ Display file icon on thumbnails
- ☑ Display file size information in folder tips
- □ Display the full path in the title bar (Classic theme only)
- Hidden files and folders
  - ○ Don't show hidden files, folders, or drives
  - ● Show hidden files, folders, and drives
- □ Hide empty drives in the Computer folder
- □ Hide extensions for known file types
- □ Hide protected operating system files (Recommended)
- □ Launch folder windows in a separate process
- □ Restore previous folder windows at logon
- ☑ Show drive letters

Restore Defaults

OK | Cancel | Apply

Figure 28: Show hidden files, folders and drives in Folder Options

After installation, import the registry file *context_menu.reg*, found in the C:\tools directory. This registry file extends context menu with tools required for the next exercises (Figure 29).

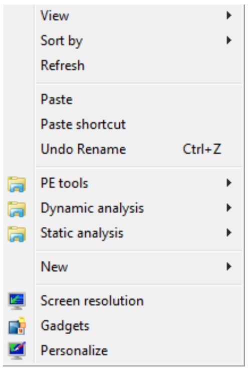


Figure 29. Context menu in Winbox.

When *Winbox* has been configured, all tools have been installed and the registry file has been imported, create a snapshot of the machine's current state.

4.4 Configuring Winbox for cuckoo automatic analysis

In this step, participants prepare the cuckoo snapshot for automatic malware analysis. In the snapshot named cuckoo there are no additional tools – it is a clean installation of windows machine with the cuckoo agent and Python with required image library.

First, restore the snapshot called "cuckoo" which was created in step 4.2.

4.4.1 Create user accounts

Malware may check the number of user accounts in a Windows machine. For that reason, participants should create random accounts in operating system. To create user accounts in the Windows operating system, press Start, choose Control panel and then choose "Add or remove user accounts" from "Users Accounts and Family Safety category" (Figure 30).

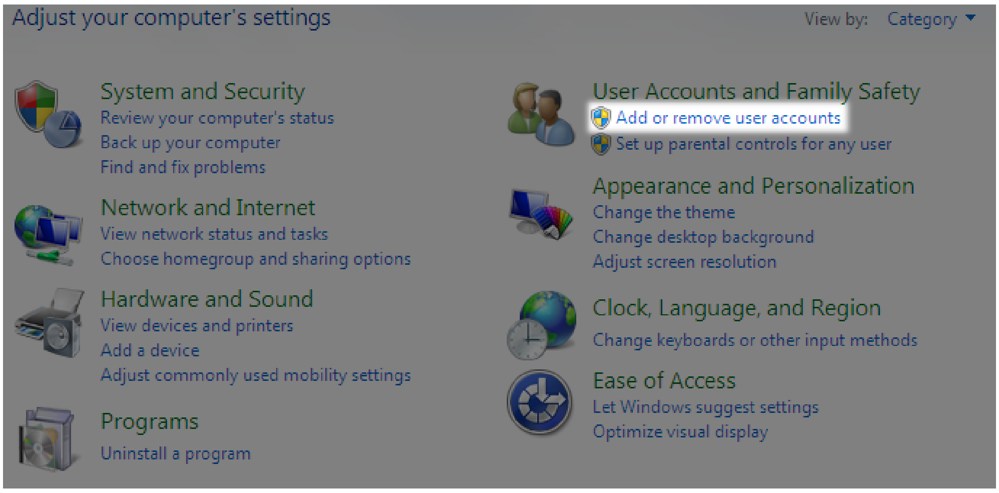

Adjust your computer's settings
View by: Category
System and Security
Review your computer's status
Back up your computer
Find and fix problems
Network and Internet
View network status and tasks
Choose homegroup and sharing options
Hardware and Sound
View devices and printers
Add a device
Adjust commonly used mobility settings
Programs
Uninstall a program
User Accounts and Family Safety
Add or remove user accounts
Set up parental controls for any user
Appearance and Personalization
Change the theme
Change desktop background
Adjust screen resolution
Clock, Language, and Region
Change keyboards or other input methods
Ease of Access
Let Windows suggest settings
Optimize visual display

Figure 30. Add or remove user account option in control panel.

Next, click “Create a new account” and follow the wizard. Set account types to Standard user. Figure 31 shows user accounts created in the Windows operating system.

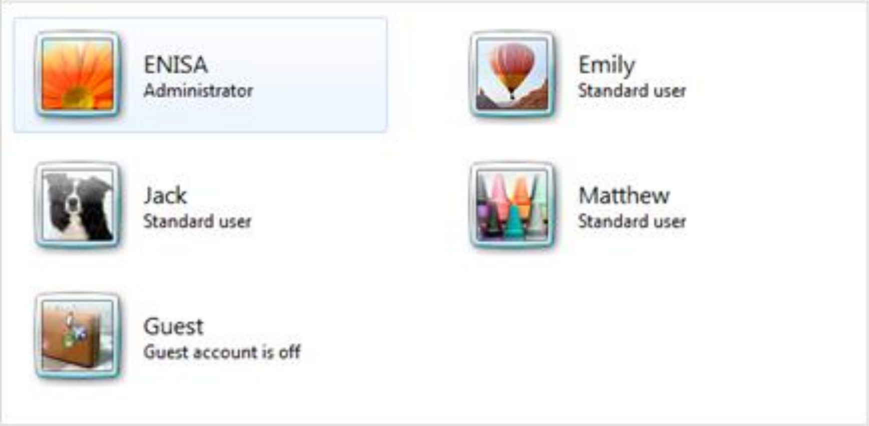


Figure 31: User accounts in Windows machine

4.4.2 Install Python and cuckoo agent

The Cuckoo[9] agent requires Python 2.7 [10]and the Python Image Library[11]. Python and the Python Image Library are available as executable installers and they need to be installed in the cuckoo snapshot. All required files (cuckoo agent, python and Python Image Library) are in the *winbox_cuckoo.zip* archive. Install Python and the Python Image Library. Additionally, the Cuckoo agent (*agent.py* file) must be added into the Startup folder. To add a file to the Startup folder, drag and drop it into *Startup* folder in *Start menu*.

It is recommend changing *agent.py* to another name. To hide the cuckoo agent window, change the file extension from *.py* to *.pyw*.

[9] http://www.cuckoosandbox.org/

[10] https://www.python.org/download/windows/

[11] http://www.pythonware.com/products/pil/

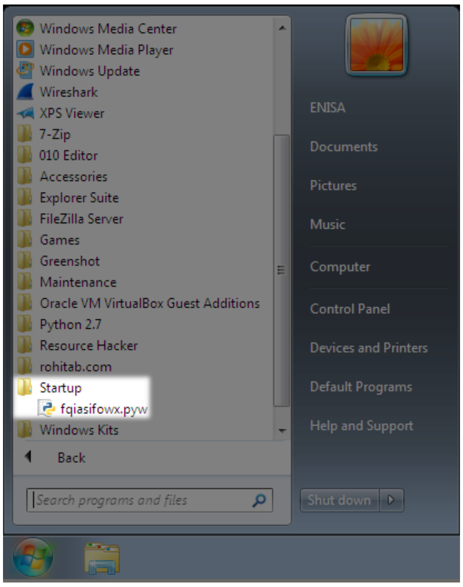


Figure 32. Cuckoo agent on Winbox machine

Delete *winbox_cuckoo.zip* and *winbox_tools.zip* from the Desktop, restart the Winbox machine, wait until it boots up and **create a snapshot of the running state of the Winbox machine**. Save the snapshot as “cuckoo” and remove the previous snapshot with the same name.

This snapshot will be used for automatic malware analysis. The Styx machine will send the potentially infected file to the cuckoo agent. Please note that it is necessary for the snapshot to be taken in the running state, because otherwise Cuckoo Sandbox will not start virtual machine during the analysis.

5 Configuring Styx virtual machine

5.1 Basic configuration

In this step, participants will perform the basic configuration of Styx and the Winbox virtual machines.

The Styx virtual machine has been prepared for this exercise. Advanced Packaging Tool (Apt) uses the *source.list* file which is stored in */etc/apt/* directory. In that file is specified an offline repository, which contains all the required packages. Packages have been stored locally in */home/enisa/enisa/packages/apt*. The Pip repository has been also modified. In */home/enisa/.pip/pip.conf* file the specified offline repository - */home/enisa/enisa/packages/pip*.

After configuring the network, participants can connect to the Styx virtual machine using the Secure Shell (SSH) protocol. To connect to the Styx machine, connect to IP address 192.168.56.10 using any SSH client of choice (for example PuTTY for Windows or the SSH shell command from a Linux terminal).

5.1.1 Network configuration

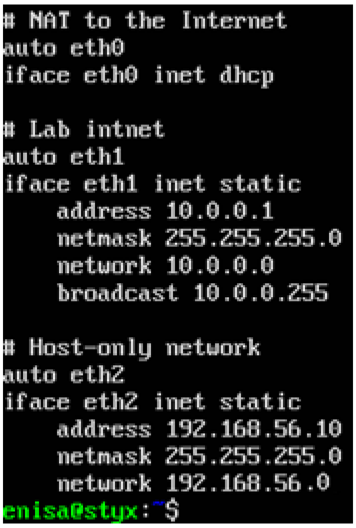
The Styx machine is a gateway for Winbox. As the gateway IP address for Winbox is 10.0.0.1, Styx should have that IP address. To configure network interfaces in Styx, start the Styx virtual machine in VirtualBox and login with *enisa* as the username. The Styx machine will be used as a gateway for Winbox machine and the IP address in Styx should be the same as given in the Winbox in Default Gateway field.

To set up network interfaces, edit the */etc/network/interfaces* file.

Edit */etc/network/interfaces*

```
$ sudo nano /etc/network/interfaces
```

The content of the */etc/network/interfaces* file is shown in Figure 33.



```
# NAT to the Internet
auto eth0
iface eth0 inet dhcp

# Lab intnet
auto eth1
iface eth1 inet static
    address 10.0.0.1
    netmask 255.255.255.0
    network 10.0.0.0
    broadcast 10.0.0.255

# Host-only network
auto eth2
iface eth2 inet static
    address 192.168.56.10
    netmask 255.255.255.0
    network 192.168.56.0
enisa@styx:~$
```

Figure 33: Network configuration for Styx

After modifying the */etc/network/interfaces* file*,* execute the *ifdown* and *ifup* commands to set IP addresses for all interfaces.

Restart network interfaces

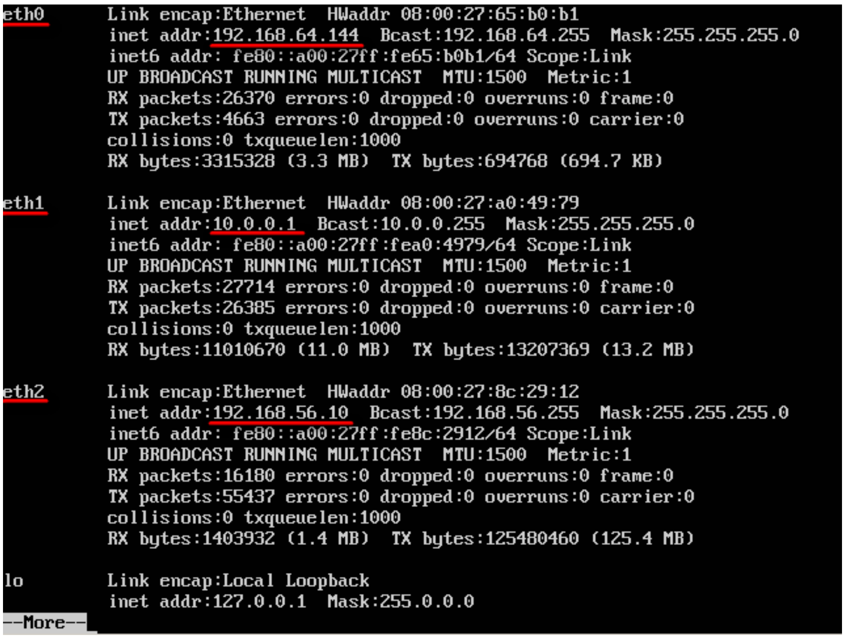
```
$ sudo ifdown eth0 eth1 eth2
$ sudo ifup eth0 eth1 eth2
```

To verify network configuration in the Styx virtual machine, execute the *ifconfig* command.

Verify network configuration in Styx machine

```
$ ifconfig | more
```

By executing *ifconfig*, the Styx machine will show its network configuration (Figure 34). The first interface (eth0) for Internet connectivity has its IP assigned by the DHCP service, so the IP address may vary. The second adapter (eth1) is the internal network between the Styx and Winbox machines and should have an IP address of 10.0.0.1. The third interface used for Cuckoo network traffic should be assigned IP address 192.168.56.10.



```
eth0      Link encap:Ethernet  HWaddr 08:00:27:65:b0:b1
          inet addr:192.168.64.144  Bcast:192.168.64.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe65:b0b1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:26370 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4663 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3315328 (3.3 MB)  TX bytes:694768 (694.7 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:a0:49:79
          inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea0:4979/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27714 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26385 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11010670 (11.0 MB)  TX bytes:13207369 (13.2 MB)

eth2      Link encap:Ethernet  HWaddr 08:00:27:8c:29:12
          inet addr:192.168.56.10  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8c:2912/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16180 errors:0 dropped:0 overruns:0 frame:0
          TX packets:55437 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1403932 (1.4 MB)  TX bytes:125480460 (125.4 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
--More--
```

Figure 34. Verifying network configuration in the Styx machine.

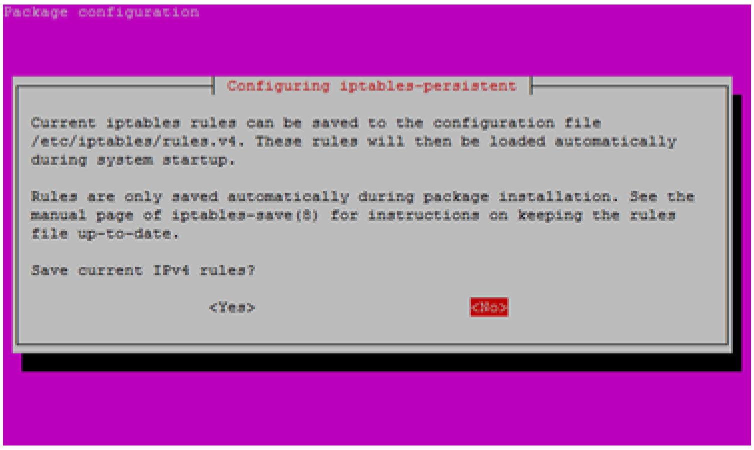
5.2 Network traffic filtering

Network traffic filtering allows users to change network connectivity between the Styx and Winbox machines. We can allow Internet access to the Winbox machine or we can send all network traffic from Styx to the Internet Network Simulator. To change network traffic connectivity, a special script was created in iptables format. To use any rules of the network script, the iptables-persistent package is required and needs to be installed.

Iptables persistent installation

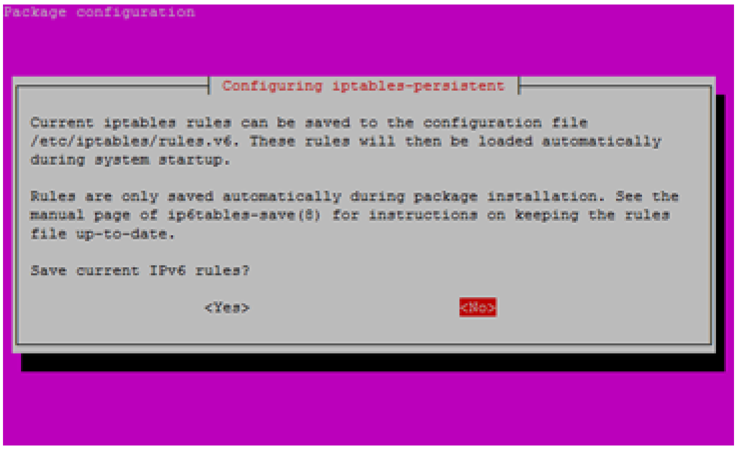
```
$ sudo apt-get install iptables-persistent
```

During the installation of iptables-persistent, the wizard will ask about saving the current IPv4 and IPv6 rules. Both answers should be “No” (Figure 35, Figure 36).



```
Package configuration

           ┤ Configuring iptables-persistent ├

  Current iptables rules can be saved to the configuration file
  /etc/iptables/rules.v4. These rules will then be loaded automatically
  during system startup.

  Rules are only saved automatically during package installation. See the
  manual page of iptables-save(8) for instructions on keeping the rules
  file up-to-date.

  Save current IPv4 rules?

                   <Yes>                         <No>
```

Figure 35: Screenshot of iptables-persistent installation and IPv4 rules



```
Package configuration

        ┤ Configuring iptables-persistent ├

 Current iptables rules can be saved to the configuration file
 /etc/iptables/rules.v6. These rules will then be loaded automatically
 during system startup.

 Rules are only saved automatically during package installation. See the
 manual page of ip6tables-save(8) for instructions on keeping the rules
 file up-to-date.

 Save current IPv6 rules?

                    <Yes>                       <No>
```

Figure 36: Screenshot of the iptables-persistent installation and IPv6 rules

After installing the iptables-persistent package, we can set the default network policy for the Winbox machine after the Styx boot. For that, we want to set up simulated Internet access for Winbox.

Set Internet Services simulation on boot for Winbox.

```
$ sudo ln -s /lab/rules/netsim.v4 /etc/iptables/rules.v4
```

If you boot the Styx and Winbox machines, all network traffic from *Winbox* will be redirected by default to the Internet simulator.

5.2.1 Tor tunnelling

The next network rule in the Styx machine allows the sending of all network traffic to the Internet using the Tor protocol. To use it, participants must install and configure Tor software on the Styx machine.

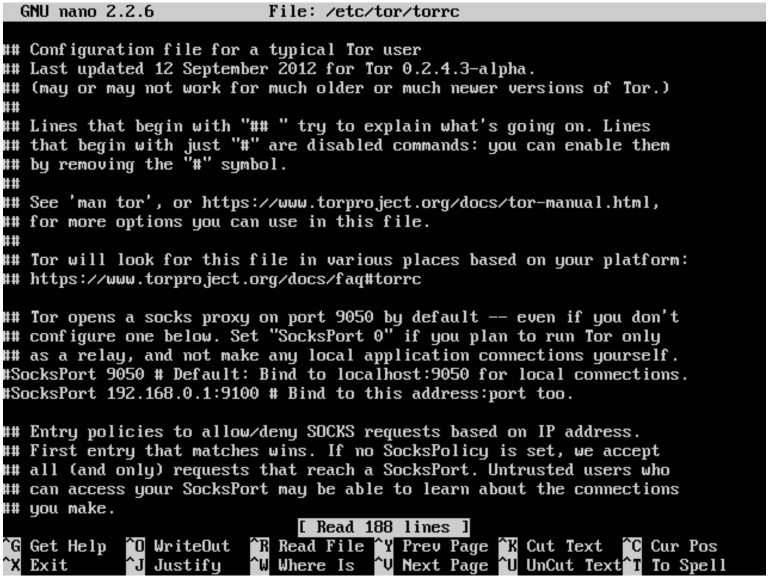
To install Tor software, run *apt-get*.

Tor installation

```
$ sudo apt-get install tor
```

After TOR installation, participants must edit the /etc/tor/torrc file to listen on 127.0.0.1 and 10.0.0.1.

To edit the file in a Linux environment, you can use the *vi* or *nano* editors. In this exercise, we will use *nano*.



```
  GNU nano 2.2.6              File: /etc/tor/torrc

## Configuration file for a typical Tor user
## Last updated 12 September 2012 for Tor 0.2.4.3-alpha.
## (may or may not work for much older or much newer versions of Tor.)
##
## Lines that begin with "## " try to explain what's going on. Lines
## that begin with just "#" are disabled commands: you can enable them
## by removing the "#" symbol.
##
## See 'man tor', or https://www.torproject.org/docs/tor-manual.html,
## for more options you can use in this file.
##
## Tor will look for this file in various places based on your platform:
## https://www.torproject.org/docs/faq#torrc

## Tor opens a socks proxy on port 9050 by default -- even if you don't
## configure one below. Set "SocksPort 0" if you plan to run Tor only
## as a relay, and not make any local application connections yourself.
#SocksPort 9050 # Default: Bind to localhost:9050 for local connections.
#SocksPort 192.168.0.1:9100 # Bind to this address:port too.

## Entry policies to allow/deny SOCKS requests based on IP address.
## First entry that matches wins. If no SocksPolicy is set, we accept
## all (and only) requests that reach a SocksPort. Untrusted users who
## can access your SocksPort may be able to learn about the connections
## you make.
                              [ Read 188 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figure 37. Nano editor with /etc/tor/torrc content.

Edit /etc/tor/torrc file

```
$ sudo nano /etc/tor/torrc
```

To scroll down to the bottom of the file, press CTRL + - and CTRL + V. At the end of the file, add the following content:

TOR configuration: /etc/tor/torrc

```
VirtualAddrNetworkIPv4 10.192.0.0/10
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 127.0.0.1
TransListenAddress 10.0.0.1
DNSPort 53
DNSListenAddress 127.0.0.1
DNSListenAddress 10.0.0.1
```

To save the modifications, press CTRL + x, press the Y button and then press Enter to save the changes into the file.

After editing the /etc/tor/torrc file, the participants should reload the tor service to apply the new configuration. (The modified file can be found in /home/enisa/enisa/ex1/files/torrc)

TOR restart service

```
$ sudo service tor restart
```

6 Network simulator

In the basic mode of operation, all network traffic coming from the analysis machine (Winbox) is redirected to the network simulator running on the Styx machine. The network simulator is built using the INetSim tool, which emulates various basic network services.

6.1 INetSim installation

In the first step, participants install INetSim.

INetSim installation

```
$ sudo apt-get install inetsim
```

6.2 INetSim configuration

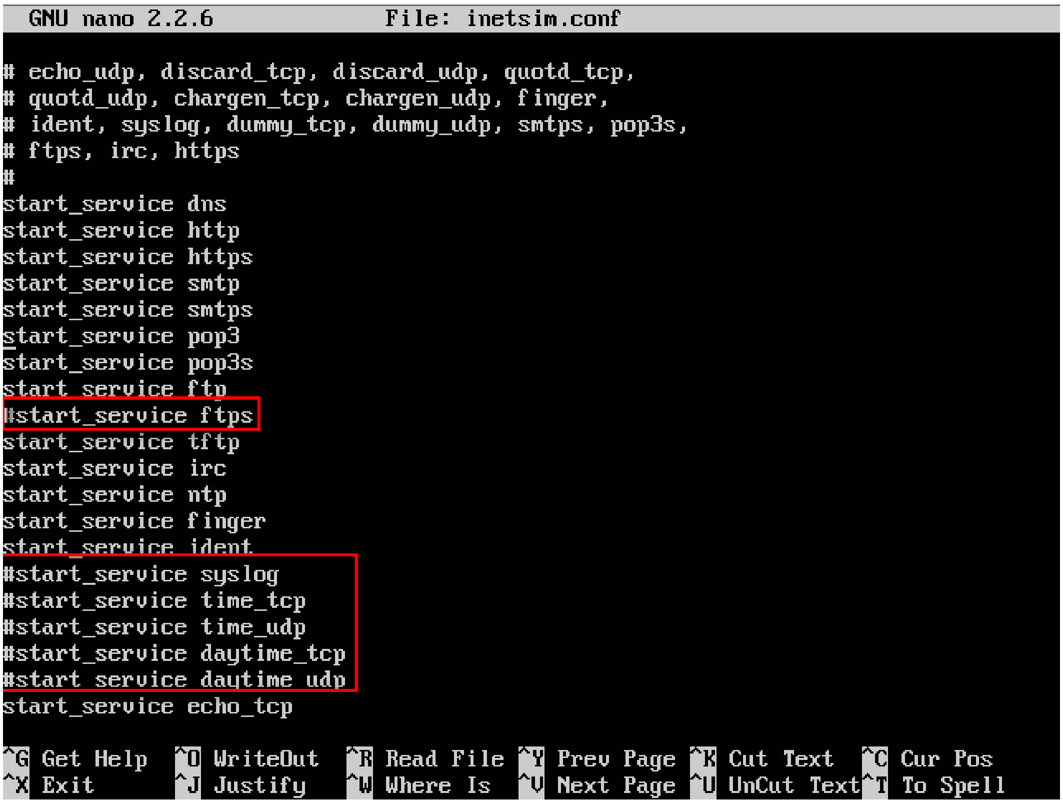
In the second step, participants configure INetSim by setting the listening address, getting a quick overview of the configuration file.

File */etc/inetsim/inetsim.conf* is the main configuration file of INetSim service. In this file, we can set various properties for each service. We can also disable unnecessary services. Participants must disable unnecessary services because malware may detect it is running in a virtual environment.

To disable services, edit */etc/inetsim/inetsim.conf* with *nano* and add the # character to specified services. To disable a service, comment out the specified *start_service* lines (Figure 38).

Edit INetSim network configuration file

```
$ sudo nano /etc/inetsim/inetsim.conf
```



```
  GNU nano 2.2.6              File: inetsim.conf

# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
start_service dns
start_service http
start_service https
start_service smtp
start_service smtps
start_service pop3
start_service pop3s
start_service ftp
#start_service ftps
start_service tftp
start_service irc
start_service ntp
start_service finger
start_service ident
#start_service syslog
#start_service time_tcp
#start_service time_udp
#start_service daytime_tcp
#start_service daytime_udp
start_service echo_tcp

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figure 38. Disabled services in inet.conf file

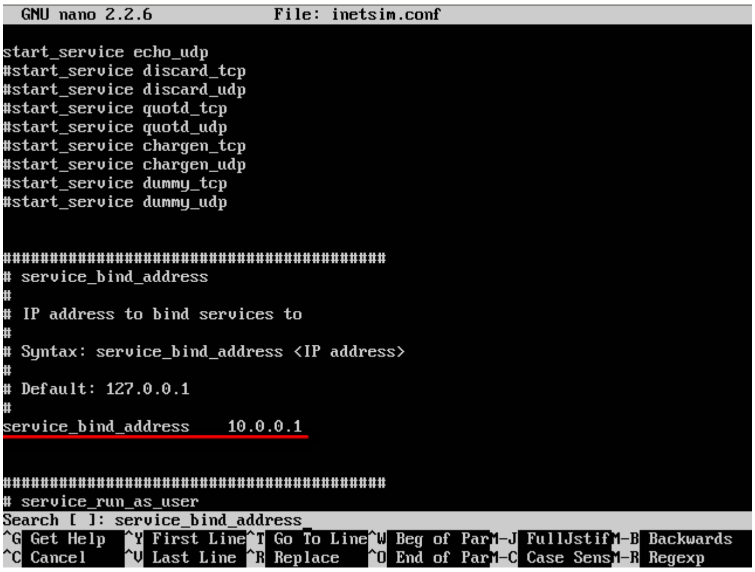
Comment out the *start_service* directive for following services:

- `ftps,`
- `syslog,`
- `time_tcp,`
- `time_udp,`
- `daytime_tcp,`

- `daytime_udp,`
- `discard_tcp,`
- `discard_udp,`
- `quotd_tcp,`
- `quotd_udp,`
- `charen_tcp,`
- `chargen_udp,`
- `dummy_tcp,`
- `dummy_udp`

For access to INetSim from the Winbox machine, we need to define a listening address. The default listening address is 127.0.0.1, which allows only connections from localhost. To give access to the 10.0.0.0 network to INetSim, find *service_bind_address* in the *inetsim.conf* configuration file.

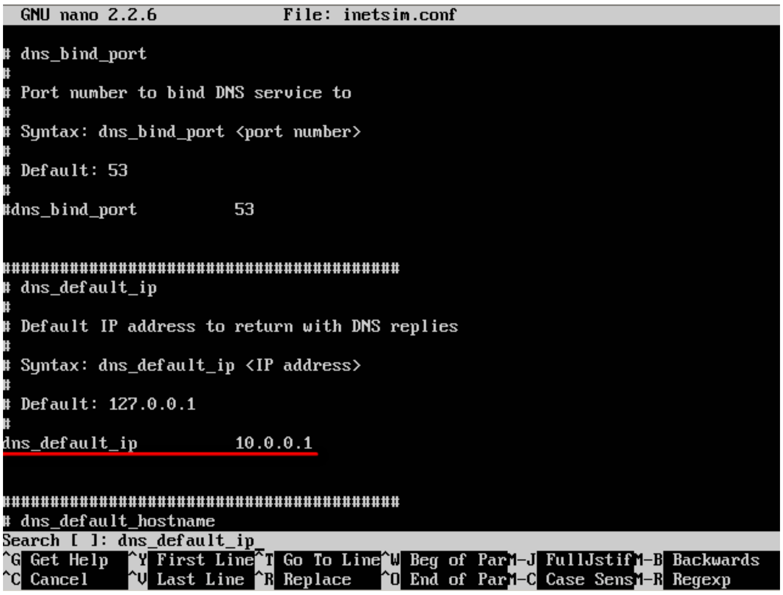
To find a specified phrase in *nano*, press CTRL + W, type *service_bind_address* and press Enter (Figure 39).



```
  GNU nano 2.2.6              File: inetsim.conf

start_service echo_udp
#start_service discard_tcp
#start_service discard_udp
#start_service quotd_tcp
#start_service quotd_udp
#start_service chargen_tcp
#start_service chargen_udp
#start_service dummy_tcp
#start_service dummy_udp


#########################################
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address    10.0.0.1


#########################################
# service_run_as_user
Search [ ]: service_bind_address
^G Get Help   ^Y First Line^T Go To Line^W Beg of ParM-J FullJstifM-B Backwards
^C Cancel     ^V Last Line ^R Replace   ^O End of ParM-C Case SensM-R Regexp
```

Figure 39. Service_bind_address directive in inetsim.conf.

The third modification in *inetsim.conf* is to change the IP address in *dns_default_ip directive*. By setting *dns_default_ip* we set the default IP address to return DNS replies. Once again, find the specified phrase by pressing CTRL + W, type *dns_default_ip* and press Enter (Figure 40).



```
  GNU nano 2.2.6              File: inetsim.conf

# dns_bind_port
#
# Port number to bind DNS service to
#
# Syntax: dns_bind_port <port number>
#
# Default: 53
#
#dns_bind_port          53


#########################################
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip          10.0.0.1


#########################################
# dns_default_hostname
Search [ ]: dns_default_ip
^G Get Help   ^Y First Line^T Go To Line^W Beg of ParM-J FullJstifM-B Backwards
^C Cancel     ^V Last Line ^R Replace   ^O End of ParM-C Case SensM-R Regexp
```

Figure 40. dns_default_ip in inetsim.conf.

The modified version of the *inetsim.conf* file can be found in the */home/enisa/enisa/ex1/files* directory.

In the next step, participants will edit banner files for certain services. By modifying these, we want to change the banner messages because analysed malware may detect patterns of a simulated Internet environment.

Depending on the file, change any dates, names, internet protocol (IP) address and so on. To edit a file, use *nano* editor. To save changes in file, press CTRL + X, press Y, and then press Enter.

To create a file in the Linux operating system, use the *touch* command.

Edit banner files for services

```
cd /var/lib/inetsim/finger (information in finger service)
$ sudo nano example.finger  (change last login time, names)
$ cd /var/lib/inetsim/ftp/ftproot (files in ftp service)
$ rm sample.txt (remove sample.txt file)
$ sudo touch file.txt (create file named file.txt)
$ sudo nano file.txt (add any text into the file.txt)
$ touch otherfile.txt (create file named otherfile.txt)
$ cd /var/lib/inetsim/http/fakefiles
$ nano sample.html (change default webpage to any other text)
```

In the next step we want to allow INetSim to start in the Styx machine. By default, starting the INetSim service is forbidden. To allow this, edit /etc/default/inetsim and change value ENABLED from 0 to 1.

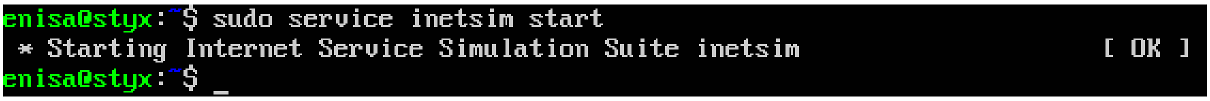
Allow to start INetSim service in styx environment

```
$ sudo nano /etc/default/inetsim
ENABLED=1
```

After making these changes in the INetSim configuration files, start the INetSim service (Figure 41).

Start INetSim service

```
$ sudo service inetsim start
```



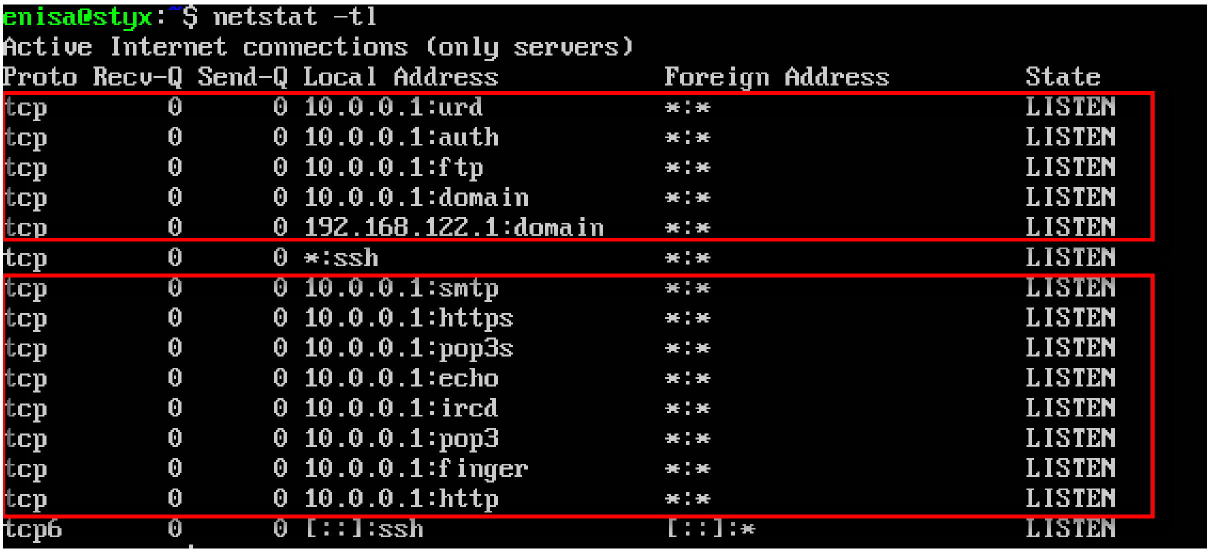
```
enisa@styx:~$ sudo service inetsim start
 * Starting Internet Service Simulation Suite inetsim                     [ OK ]
enisa@styx:~$ _
```

Figure 41. Start the INetSim service.

To verify the INetSim service, execute the *netstat* command (Figure 42).

Verify INetSim services on *Styx* using netstat command

```
$ netstat -tl
```



```
enisa@styx:~$ netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 10.0.0.1:urd            *:*                     LISTEN
tcp        0      0 10.0.0.1:auth           *:*                     LISTEN
tcp        0      0 10.0.0.1:ftp            *:*                     LISTEN
tcp        0      0 10.0.0.1:domain         *:*                     LISTEN
tcp        0      0 192.168.122.1:domain    *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 10.0.0.1:smtp           *:*                     LISTEN
tcp        0      0 10.0.0.1:https          *:*                     LISTEN
tcp        0      0 10.0.0.1:pop3s          *:*                     LISTEN
tcp        0      0 10.0.0.1:echo           *:*                     LISTEN
tcp        0      0 10.0.0.1:ircd           *:*                     LISTEN
tcp        0      0 10.0.0.1:pop3           *:*                     LISTEN
tcp        0      0 10.0.0.1:finger         *:*                     LISTEN
tcp        0      0 10.0.0.1:http           *:*                     LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
```

Figure 42. Services listening on 10.0.0.1.

Services marked in Figure 42 are listening on IP address 10.0.0.1 on certain ports, and are INetSim services.

6.3 Traffic redirection to the INetSim

As we have several iptables configuration settings in the iptables-persistent package, we can choose the network type for Winbox machine. By default, after booting the Styx machine (Network traffic filtering section), all network traffic from *Winbox* is redirected to the INetSim service. Participants can use *lab-switch-net* to choose network access.

To redirect network traffic from Winbox to the INetSim service, participants can always use the following command:

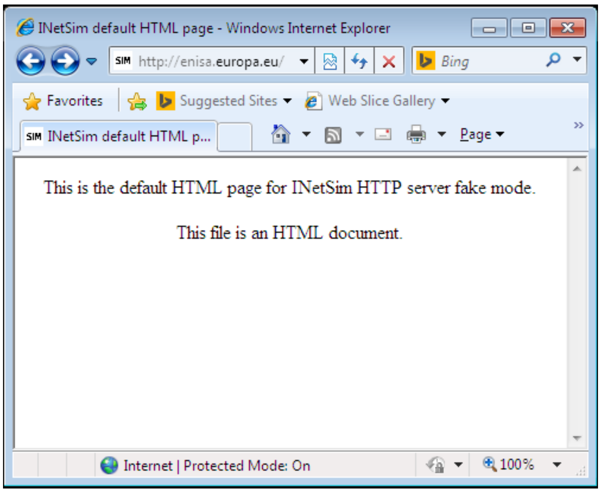
Redirect network traffic to INetSim simulator

```
$ /lab/bin/lab-switch-net netsim
```

6.4 Testing the network simulator

To test the network simulator on the Winbox machine, go back to VirtualBox, login into the Winbox machine and run the Internet Explorer browser. Participants can check the INetSim service by opening any webpage on the Winbox machine. Any web page request from the Winbox machine will be redirected to the Styx INetSim service. The INetSim service will respond with the content of sample.html file which was edited in section 6.2.

By accessing the http://enisa.europa.eu website, the participants will receive the content of sample.html file (Figure 43) as the response.


INetSim default HTML page - Windows Internet Explorer
http://enisa.europa.eu/
Bing
Favorites
Suggested Sites
Web Slice Gallery
INetSim default HTML p...
Page
Internet | Protected Mode: On
100%

This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.

Figure 43. Page http://enisa.europa.eu on Winbox machine through INetSim service.

7 Snort

Snort is a libpcap-based sniffer/logger which can be used as a network intrusion detection and prevention system. It uses a rule-based detection language as well as various other detection mechanisms and is highly extensible.[12]

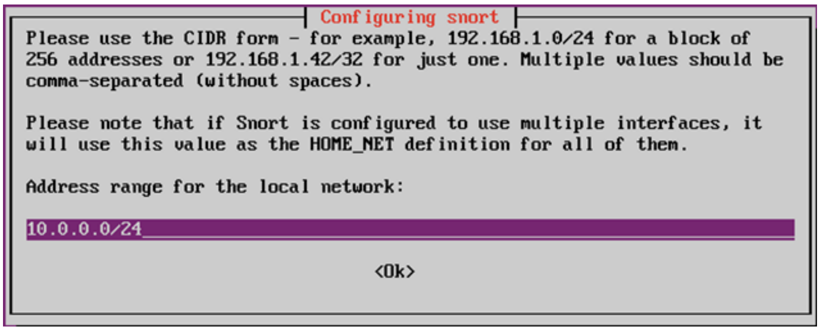
Snort is being used to detect signatures of some well-known malware. Snort will be installed on Styx virtual machine and will observe all network traffic coming from the lab network.

7.1 Snort installation

Snort installation is done on the Styx virtual machine using packages from the official repository.

Snort installation

```
$ sudo apt-get install snort
```



```
Configuring snort
Please use the CIDR form - for example, 192.168.1.0/24 for a block of
256 addresses or 192.168.1.42/32 for just one. Multiple values should be
comma-separated (without spaces).

Please note that if Snort is configured to use multiple interfaces, it
will use this value as the HOME_NET definition for all of them.

Address range for the local network:

10.0.0.0/24

<Ok>
```

Figure 44: Configuring Snort during install

[12] http://sourceforge.net/projects/snort/

During installation, snort will ask about the HOME_NET address. Our HOME_NET address is 10.0.0.0/24 (Figure 44).

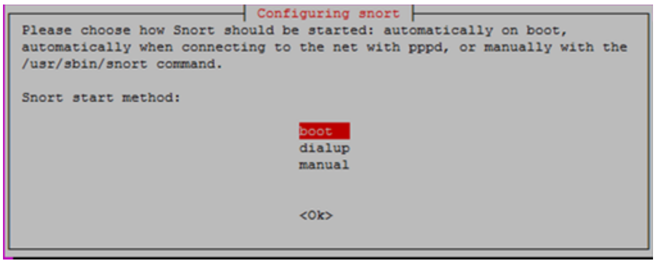
7.2 Snort configuration

In this step, participants configure Snort. After installation is finished, we need to reconfigure Snort once again.

Snort reconfiguration

```
$ sudo dpkg-reconfigure snort
```

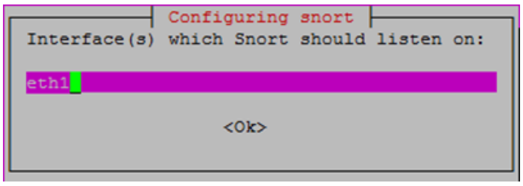
During reconfiguration, the wizard asks about the start method (Figure 45) – it should be set to *boot*.



```
Configuring snort
Please choose how Snort should be started: automatically on boot,
automatically when connecting to the net with pppd, or manually with the
/usr/sbin/snort command.

Snort start method:

                    boot
                    dialup
                    manual

                    <Ok>
```

Figure 45: Snort start method

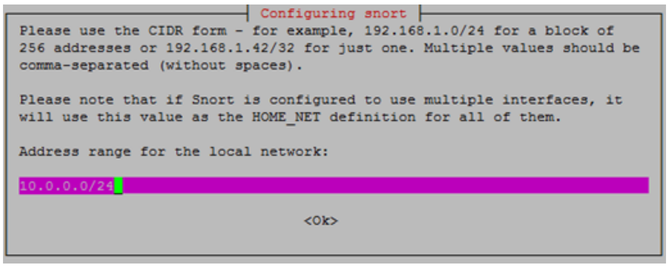
In the next step, the wizard will ask which interface Snort should listen on (Figure 46). *Styx* and *Winbox* VM communicate via the eth1 interface and snort should listen on it as well. By default, snort listens on the eth0 interface. All network traffic from the Winbox machine comes to eth1 on the Styx machine. Snort should listen on the eth1 network interface only.



```
Configuring snort
Interface(s) which Snort should listen on:

eth1

              <Ok>
```

Figure 46: Setting up snort listen interface

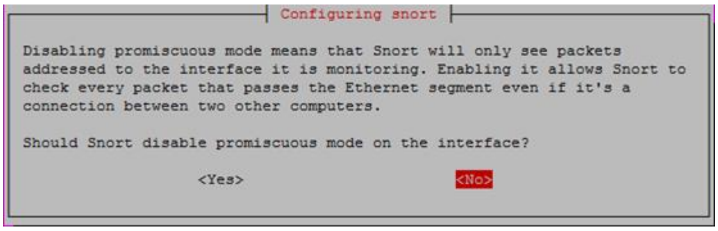
Once again the wizard will ask about home net. During the installation we set it to 10.0.0.0/24 (Figure 47).



```
Configuring snort
Please use the CIDR form - for example, 192.168.1.0/24 for a block of
256 addresses or 192.168.1.42/32 for just one. Multiple values should be
comma-separated (without spaces).

Please note that if Snort is configured to use multiple interfaces, it
will use this value as the HOME_NET definition for all of them.

Address range for the local network:

10.0.0.0/24

                    <Ok>
```

Figure 47: HOME_NET parameter

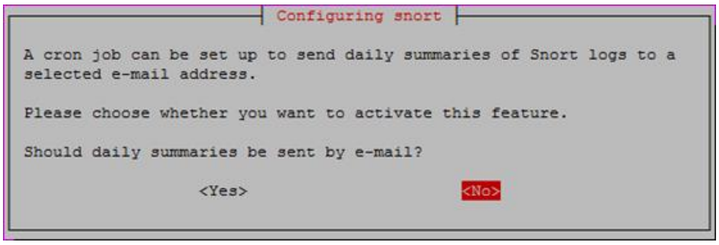
In the last steps, the wizard asks about disabling promiscuous mode (Figure 48) and daily summaries by e-mail (Figure 49). Set both answers to *No*.



```
Configuring snort

Disabling promiscuous mode means that Snort will only see packets
addressed to the interface it is monitoring. Enabling it allows Snort to
check every packet that passes the Ethernet segment even if it's a
connection between two other computers.

Should Snort disable promiscuous mode on the interface?

                <Yes>                          <No>
```

Figure 48: Disable promiscuous mode – Snort

By setting promiscuous mode on the eth1 interface, we will be able to detect any packets in the 10.0.0.0 network, not only those addressed to the gateway.



```
Configuring snort

A cron job can be set up to send daily summaries of Snort logs to a
selected e-mail address.

Please choose whether you want to activate this feature.

Should daily summaries be sent by e-mail?

                <Yes>                          <No>
```

Figure 49: Disable daily summaries by e-mail from Snort service

7.3 Snort rules update

Snort uses signatures to describe network traffic. Participants can update rules using rules downloaded from the official snort website. Downloaded Snort community rules can be also found in the */home/enisa/enisa/packages/source* directory.

Snort rules install

```
Online installation:
$ wget https://www.snort.org/downloads/community/community-
rules.tar.gz


Offline installation:
$ cd /home/enisa/enisa/packages/source


$ tar xf community-rules.tar.gz
$ sudo cp community-rules/community.rules /etc/snort/rules
```

In the next step participants must edit the Snort configuration file (*snort.conf*) and include the downloaded rules. To include these rules, edit */etc/snort/snort.conf* with the *nano* editor. The file owner of *snort.conf* is root, so we need root privileges to edit it. To edit it, precede the command with *sudo*.

Edit /etc/snort/snort.conf file

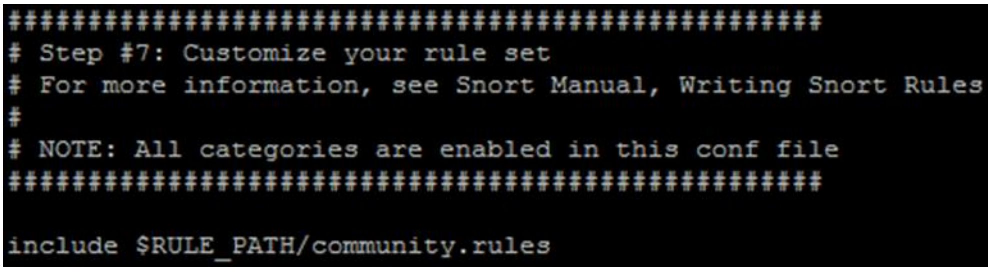
```
$ sudo nano /etc/snort/snort.conf
```

In the *snort.conf* file, we need to include community rules. These rules need to load when Snort starts. To include community rules, edit *snort.conf* and add the include line.

To jump to a specific line, press CTRL + -. Jump to line 559 and include the following entry (Figure 50):

Include community rules in Snort.

```
include $RULE_PATH/community.rules
```



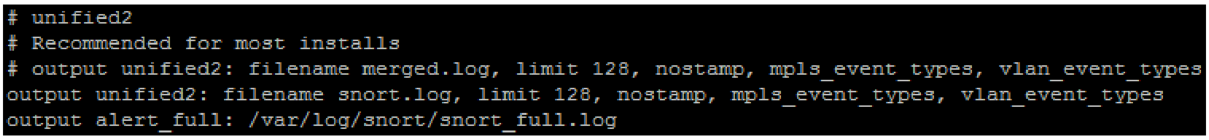
```
###################################################
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
###################################################

include $RULE_PATH/community.rules
```

Figure 50: Part of the snort.conf configuration file and community.rules

In the next step, we want to define the logging mode for Snort. In our configuration, we log any Snort alerts to a dedicated file. Jump to line (CTRL + - in *nano*) 537 and add the following line:

Log Snort alerts to snort_full.log file

```
output alert_full: /var/log/snort/snort_full.log
```



```
# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output alert_full: /var/log/snort/snort_full.log
```

Figure 51: Save alerts to /var/log/snort/snort_full.log

All alerts from snort will be stored in */var/log/snort/snort_full.log*.

After modifying */etc/snort/snort.conf*, restart the snort service to apply changes.

Snort restart service

```
$ sudo service snort restart
```

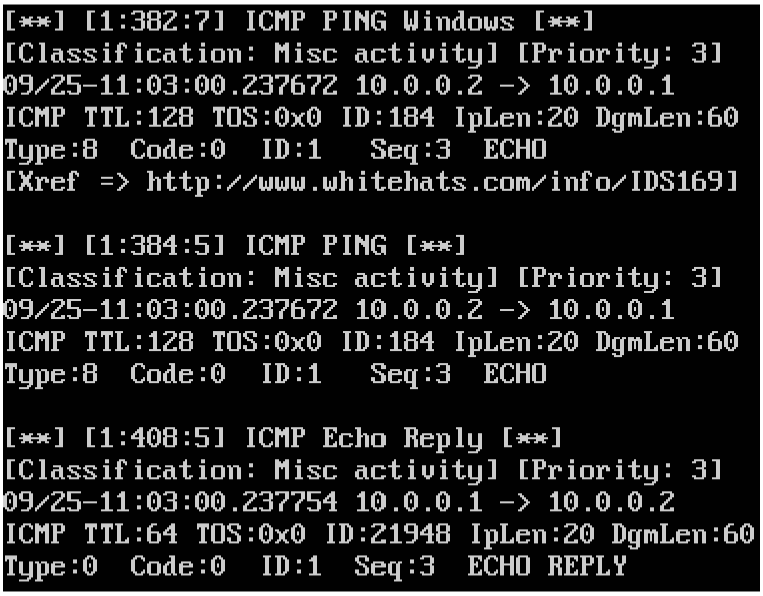
7.4 Snort tests

In this step, participants will log into the Winbox machine and try to trigger Snort rules.

If you ping the Styx machine (10.0.0.1) from the Winbox machine (10.0.0.2), Snort should generate informational data. All information from the Snort service is saved in the *snort_full.log* file in */var/log/snort*. To view this information in real time, use the *tail* command with *sudo* to access a file.

View snort_full.log file in real time

```
$ sudo tail -f /var/log/snort/snort_full.log
```



```
[**] [1:382:7] ICMP PING Windows [**]
[Classification: Misc activity] [Priority: 3]
09/25-11:03:00.237672 10.0.0.2 -> 10.0.0.1
ICMP TTL:128 TOS:0x0 ID:184 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:3  ECHO
[Xref => http://www.whitehats.com/info/IDS169]

[**] [1:384:5] ICMP PING [**]
[Classification: Misc activity] [Priority: 3]
09/25-11:03:00.237672 10.0.0.2 -> 10.0.0.1
ICMP TTL:128 TOS:0x0 ID:184 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:3  ECHO

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
09/25-11:03:00.237754 10.0.0.1 -> 10.0.0.2
ICMP TTL:64 TOS:0x0 ID:21948 IpLen:20 DgmLen:60
Type:0  Code:0  ID:1  Seq:3  ECHO REPLY
```

Figure 52: ICMP PING and ICMP REPLY activity detected by snort

Snort has detected ICMP PING activity from IP 10.0.0.2 which matched to the Windows operating system. Snort also identified an ICMP Echo Reply from 10.0.0.1 (Styx, gateway) to 10.0.0.2 (Winbox machine) (Figure 52).

8 MITMProxy

Mitmproxy is an SSL-capable man-in-the-middle HTTP proxy. It provides a console interface that allows traffic flows to be inspected and edited on the fly.[13]

8.1 MITMProxy installation

In this step participants, perform MITMProxy installation. To install the MITMProxy package, use *apt-get* command.

MITMProxy installation

```
$ sudo apt-get install mitmproxy
```

8.2 MITMProxy test

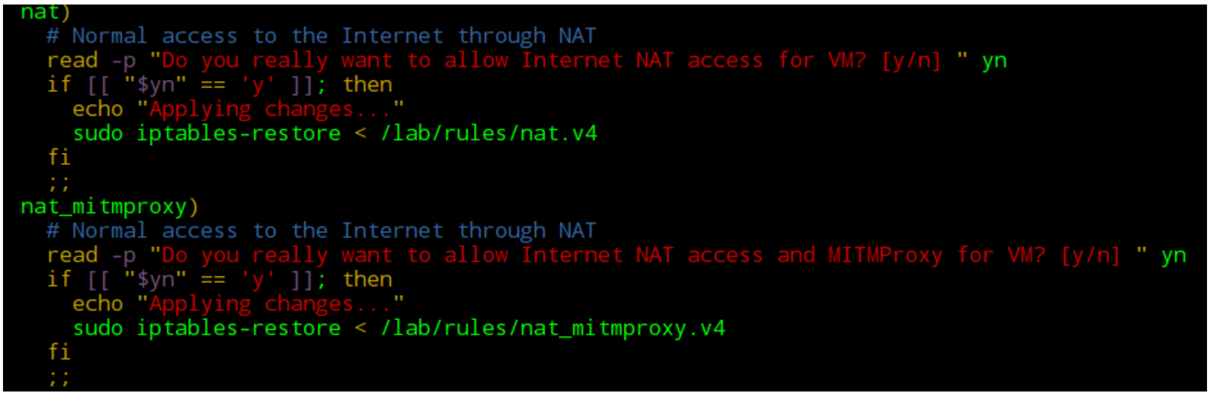
To change network access for *Winbox,* we can use *lab-switch-net* script. In *lab-switch-net* we can use a defined rule called *nat_mitmproxy. Nat_mitmproxy* is network access using network address translation (NAT) through MITMProxy.

NAT access has been limited due to security reasons to avoid malware communication to the outside world.

Edit *lab-switch-net* script and enable NAT access for nat_mitmproxy

```
$ sudo nano /lab/bin/lab-switch-net
```

[13] http://mitmproxy.org/doc/index.html



```
 nat)
   # Normal access to the Internet through NAT
   read -p "Do you really want to allow Internet NAT access for VM? [y/n] " yn
   if [[ "$yn" == 'y' ]]; then
     echo "Applying changes..."
     sudo iptables-restore < /lab/rules/nat.v4
   fi
   ;;
 nat_mitmproxy)
   # Normal access to the Internet through NAT
   read -p "Do you really want to allow Internet NAT access and MITMProxy for VM? [y/n] " yn
   if [[ "$yn" == 'y' ]]; then
     echo "Applying changes..."
     sudo iptables-restore < /lab/rules/nat_mitmproxy.v4
   fi
   ;;
```

Figure 53. Enabling Internet access for nat_mitmproxy

If you want to test MITMProxy with NAT access, use the command below:

MITMProxy network redirect

```
$ sudo /lab/bin/lab-switch-net nat_mitmproxy
```

This rule redirects all HTTP (80) and HTTPS (443) traffic to port 8080 on the Styx machine. It also masquerades the connection. After configuring iptables, start the MITMProxy tool.

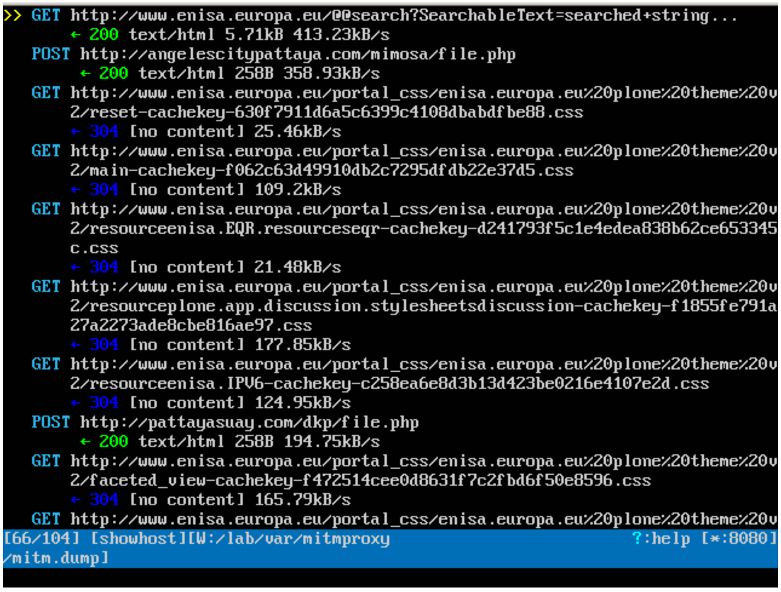
To start the MITMProxy tool, enter the following command:

Start MITMProxy

```
$ /lab/bin/lab-mitmproxy
```

Using this simple script, all traffic from the MITMProxy tool will be logged to the /lab/var/mitmproxy/mitm.dump file.

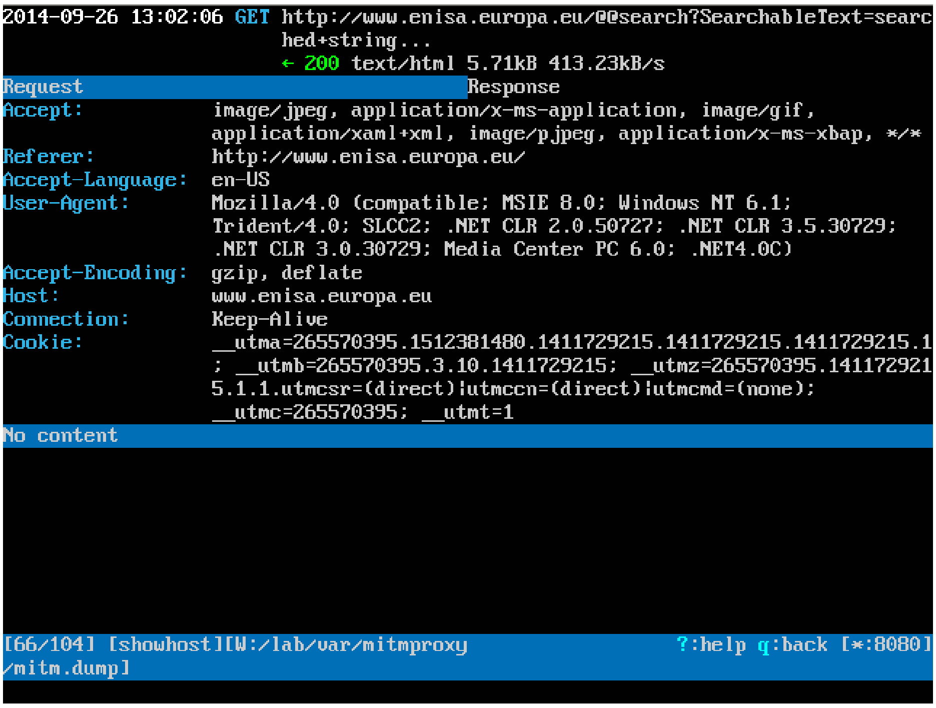
Now all traffic from *Winbox* will pass through *Styx* and the MITMProxy tool. To generate web traffic from *Winbox*, login to the Winbox machine and access any webpage from Internet Explorer. In MITMProxy, you should see the requested site. Please remember to disable all services that may listen on port 8080 (i.e. *sudo service inetsim stop*, Cuckoo web panel etc.)



```
>> GET http://www.enisa.europa.eu/@@search?SearchableText=searched+string...
        ← 200 text/html 5.71kB 413.23kB/s
   POST http://angelescitypattaya.com/mimosa/file.php
         ← 200 text/html 258B 358.93kB/s
   GET http://www.enisa.europa.eu/portal_css/enisa.europa.eu%20plone%20theme%20v
       2/reset-cachekey-630f7911d6a5c6399c4108dbabdfbe88.css
       ← 304 [no content] 25.46kB/s
   GET http://www.enisa.europa.eu/portal_css/enisa.europa.eu%20plone%20theme%20v
       2/main-cachekey-f062c63d49910db2c7295dfdb22e37d5.css
       ← 304 [no content] 109.2kB/s
   GET http://www.enisa.europa.eu/portal_css/enisa.europa.eu%20plone%20theme%20v
       2/resourceenisa.EQR.resourceseqr-cachekey-d241793f5c1e4edea838b62ce653345
       c.css
       ← 304 [no content] 21.48kB/s
   GET http://www.enisa.europa.eu/portal_css/enisa.europa.eu%20plone%20theme%20v
       2/resourceplone.app.discussion.stylesheetsdiscussion-cachekey-f1855fe791a
       27a2273ade8cbe816ae97.css
       ← 304 [no content] 177.85kB/s
   GET http://www.enisa.europa.eu/portal_css/enisa.europa.eu%20plone%20theme%20v
       2/resourceenisa.IPV6-cachekey-c258ea6e8d3b13d423be0216e4107e2d.css
       ← 304 [no content] 124.95kB/s
   POST http://pattayasuay.com/dkp/file.php
         ← 200 text/html 258B 194.75kB/s
   GET http://www.enisa.europa.eu/portal_css/enisa.europa.eu%20plone%20theme%20v
       2/faceted_view-cachekey-f472514cee0d8631f7c2fbd6f50e8596.css
       ← 304 [no content] 165.79kB/s
   GET http://www.enisa.europa.eu/portal_css/enisa.europa.eu%20plone%20theme%20v
[66/104] [showhost][W:/lab/var/mitmproxy                          ?:help [*:8080]
/mitm.dump]
```

Figure 54: Winbox accessing http://www.enisa.europa.eu webpage through MITMProxy

By using MITMProxy, participants can see all requested sites from the Winbox machine. By using arrows, participants can choose a specific request and see its details.



```
2014-09-26 13:02:06 GET http://www.enisa.europa.eu/@@search?SearchableText=searc
                        hed+string...
                        ← 200 text/html 5.71kB 413.23kB/s
Request                                 Response
Accept:                  image/jpeg, application/x-ms-application, image/gif,
                         application/xaml+xml, image/pjpeg, application/x-ms-xbap, */*
Referer:                 http://www.enisa.europa.eu/
Accept-Language:         en-US
User-Agent:              Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1;
                         Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729;
                         .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C)
Accept-Encoding:         gzip, deflate
Host:                    www.enisa.europa.eu
Connection:              Keep-Alive
Cookie:                  __utma=265570395.1512381480.1411729215.1411729215.1411729215.1
                         ; __utmb=265570395.3.10.1411729215; __utmz=265570395.141172921
                         5.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
                         __utmc=265570395; __utmt=1
No content







[66/104] [showhost][W:/lab/var/mitmproxy                     ?:help q:back [*:8080]
/mitm.dump]
```

Figure 55. Details of enisa.europa.eu request.

9 Volatility

Volatility is an open source framework for the extraction of digital artifacts from volatile memory (RAM) samples. It is now (2014) developed and supported by The Volatility Foundation. This tool will be used in Exercise 4 “Advanced artifact handling”.

To install volatility tool on the Styx machine use apt-get command as below:

Install volatility and dependencies

```
$ sudo apt-get -y install volatility volatility-* libdistorm64-dev
$ sudo ln -s /usr/share/volatility/vol.py /usr/bin/vol.py
```

In the next step participants must copy zeusscan.py to the proper directory.

Copy zeusscan.py plugin

```
$ sudo cp /home/enisa/enisa/ex4/zeusscan.py /usr/lib/python2.7/dist-packages/volatility/plugins/
```

Distrom3 is a disassembler library for x86/AMD64 architectures. The library converts a binary data stream into assembler instructions, represented as python data structures. These are needed by Volatility’s apihooks, impscan, callbacks, volshell, linux_volshell and mac_volshell plug-ins.

Unpack and install distorm3

```
$ cd /home/enisa/enisa/ex4/
$ unzip distorm3.zip
$ cd distrom3/
$ python setup.py build
$ sudo python setup.py build install
```

10 Cuckoo sandbox

Cuckoo Sandbox[14] will be used to perform some simple automatic analyses. In this step, we will show how to install and configure Cuckoo Sandbox. Cuckoo Sandbox will be running on the Styx virtual machine. To run automatic analyses, Cuckoo Sandbox will need to manage virtualisation on the host machine – automatically starting and stopping the virtual machines. To do this, the libvirt protocol will be used.

10.1 Cuckoo Sandbox installation

To run the Cuckoo Sandbox properly, participants must install the required python libraries (the autoinstaller script is available at */home/enisa/enisa/ex1/files/cuckoo_deps.sh*).

Cuckoo Sandbox dependencies – python libraries

```
$ sudo apt-get install python-sqlalchemy python-bson python-dpkt
python-jinja2 python-magic python-pymongo python-gridfs python-
libvirt python-django python-bottle python-pefile python-chardet
```

Cuckoo was developed in Python and integrated with tools like Yara, Ssdeep and MongoDB. All of these tools are required to be installed in order to run Cuckoo Sandbox.

First, participants must install the Yara tool. Yara is a tool aimed at helping malware researchers to identify and classify malware samples[15]. To compile and install the Yara tool, first install the required compilers and python development tools.

Yara tool: dependencies (compilers and python development tools)

```
$ sudo apt-get install build-essential automake python2.7-dev
libtool
```

YARA tool: installation

```
Online installation:
$ wget https://www.github.com/plusvic/yara/archive/v2.1.0.tar.gz

Offline installation:
$ cd /home/enisa/enisa/packages/source

$ tar xf v2.1.0.tar.gz
$ cd yara-2.1.0
$ sh build.sh
$ sudo make install
```

[14] http://www.cuckoosandbox.org/
[15] http://plusvic.github.io/yara/

```
$ cd yara-python
$ python setup.py build
$ sudo python setup.py install
```

The second tool required by Cuckoo Sandbox is Ssdeep. This is a program for computing context-triggered piecewise hashes (CTPH). Also called fuzzy hashes, CTPH can match inputs that have homologies[16].

To install the Ssdeep tool, install the prepared .deb package:

ssdeep installation

```
$ sudo dpkg -i /home/enisa/enisa/packages/extra/ssdeep_2.10-
1_i386.deb
```

The last tool the Cuckoo Sandbox requires is a MongoDB[17] database. It is an open-source document database that provides high performance, high availability, and automatic scaling, and is required by the Cuckoo interface. To install it, use *apt-get install* on the Styx machine.

Cuckoo dependency: MongoDB installation

```
$ sudo apt-get install mongodb python-pymongo
```

After installing all dependent components and tools, participants must unpack the Cuckoo Sandbox archive with source code to the */opt* directory.

Cuckoo Sandbox – unpack cuckoo archive

```
$ cd /home/enisa/enisa/packages/source
$ sudo tar xf cuckoo-current.tar.gz -C /opt
$ cd /opt/cuckoo
```

10.2 Cuckoo configuration

Cuckoo contains many configuration files. The participants must edit each file described in this exercise.

Cuckoo.conf is a file that contains generic configuration options. In cuckoo.conf, we need to disable the version_check parameter, which is responsible for checking for a new version of Cuckoo software. The second parameter is machinery, which defines the module for virtualisation software. The last parameter (*ip*) defines the IP address and port that Cuckoo used to bind the result server.

Cuckoo file configuration: /opt/cuckoo/conf/cuckoo.conf

```
$ sudo nano /opt/cuckoo/conf/cuckoo.conf
version_check = off
machinery = libvirt
ip = 10.0.0.1
```

[16] http://ssdeep.sourceforge.net/
[17] http://www.mongodb.org/

Libvirt.conf is a file that must be created in the conf directory. This is a configuration file for machinery[18] libvirt in Cuckoo. In this file, we declare the basic information for the Winbox machine like snapshot name and IP address.

Cuckoo file configuration: /opt/cuckoo/conf/libvirt.conf

```
$ sudo nano /opt/cuckoo/conf/libvirt.conf
[libvirt]
machines = winbox
[winbox]
label = winbox
platform = windows
ip = 10.0.0.2
snapshot = cuckoo
interface = eth1
```

The reporting.conf file contains information on automated reports generation[19]. In this file we want to enable reporting from the MongoDB database to the web interface.

Cuckoo file configuration: /opt/cuckoo/conf/reporting.conf

```
$ vim /opt/cuckoo/conf/reporting.conf
“enabled” to “yes” for [mongodb]
```

In auxiliary.conf[20] we specify a Berkley packet filter to pass to Tcpdump. In this file we specify that we don't want to log any ARP or 10.0.0.1 network traffic.

Cuckoo file configuration: auxiliary.conf

```
$ nano /opt/cuckoo/conf/auxiliary.conf

bpf = not arp and not host 10.0.0.1
```

In the next step, participants must create a machinery module called *libvirt.py*. In this module, we will save connection information to VirtualBox on the host machine via the *libvirt* protocol. First, copy *kvm.py* file as *libvirt.py* and edit its content.

Copy libvirt cuckoo module: /opt/cuckoo/modules/machinery/libvirt.py

```
$ cd /opt/cuckoo/modules/machinery
$ sudo cp kvm.py libvirt.py
$ nano libvirt.py
```

Modify line dsn with IP address of vboxnet0.

Edit cuckoo machinery module: /opt/cuckoo/modules/machinery/libvirt.py

```
$ sudo nano /opt/cuckoo/modules/machinery/libvirt.py

dsn = “vbox+tcp://192.168.56.1/session”
```

In the last step, we want to give access to *Tcpdump* to non-root users. To do this, use the command below.

Tcpdump capabilities

```
$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
$ getcap /usr/sbin/tcpdump
```

[18] http://cuckoo.readthedocs.org/en/latest/installation/host/configuration/#machinery-conf
[19] http://cuckoo.readthedocs.org/en/latest/installation/host/configuration/#reporting-conf
[20] http://cuckoo.readthedocs.org/en/latest/installation/host/configuration/#auxiliary-conf

In this step, Cuckoo has been configured to use tools like *Yara*, *Ssdeep* and a *MongoDB* database. Cuckoo has been configured to also use the *libvirt* module, which participants have created.

To connect *Styx* and the Host machine, libvirt protocol will be used. The Styx machine has been configured to use the libvirt protocol, but the host machine must be also configured.

On your host machine (not *Styx* nor *Winbox*) you must change the configuration of the libvirtd service. It's required that Styx can start Winbox through your host machine.

Install libvirt

```
$ sudo apt-get install libvirt-bin
```

After installation, you must edit the libvirt configuration file, which is stored in */etc/libvirt/libvirtd.conf*. Change values as below:

Cuckoo file configuration: /etc/libvirt/libvirtd.conf

```
listen_tls = 0
listen_tcp = 1
tcp_port = "16509"
listen_addr = "192.168.56.1" (this is vboxnet0 IP address)
mdns_name = "ENISA Host System"
unix_sock_group = "libvirtd"
unix_sock_rw_perms = "0770"
auth_unix_ro = "none"
auth_unix_rw = "none"
auth_tcp = "none"
log_level = 2
```

Libvirt must be started as a daemon. To start it as a daemon participants must edit */etc/default/libvirt-bin* file and add "-d" flag in options.

Libvirt configuration file: /etc/default/libvirt-bin

```
libvirtd_opts = '-d -l'
```

After the changes are made on the host machine, participants must restart the libvirt service. To restart it, type the command:

Restart libvirt service on host machine

```
service libvirt-bin restart
```

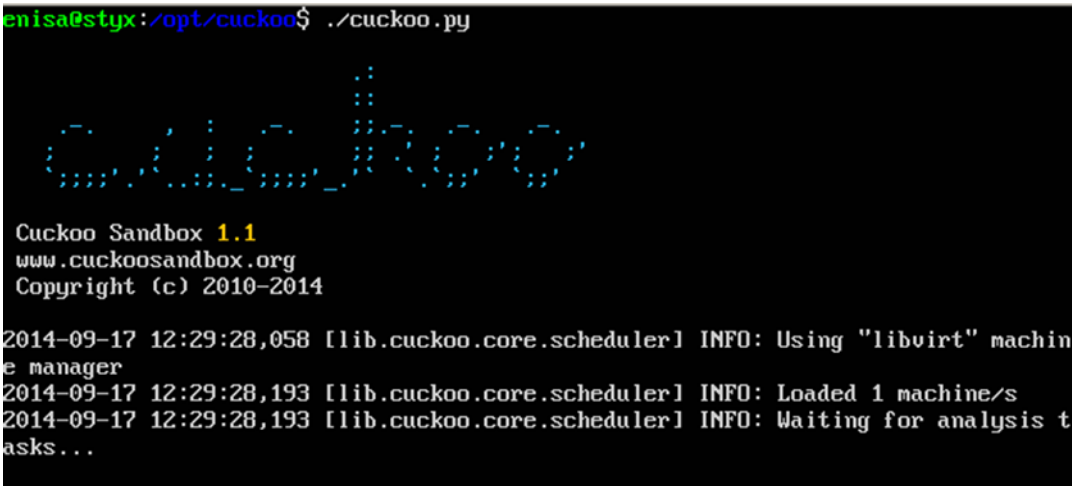
Module *libvirt* was configured to use a connection to 192.168.56.1 (the IP address of the vboxnet0 interface) to start and stop the virtual machines. In the machine configuration file (*libvirt.conf*) only one machine (*Winbox*) has been defined. This file also defined the IP address for *Winbox*, the snapshot which will be used for analysis (named "cuckoo") and the network interface on which the Cuckoo server should receive packets from the virtual machine.

10.3 Testing the Cuckoo Sandbox

Participants will test the automatic analyses by performing a sample analysis of a harmless file. First, participants must start the Cuckoo server (Figure 56) and then the Cuckoo webserver (Figure 57).

Start Cuckoo server

```
$ cd /opt/cuckoo
$ ./cuckoo.py
```



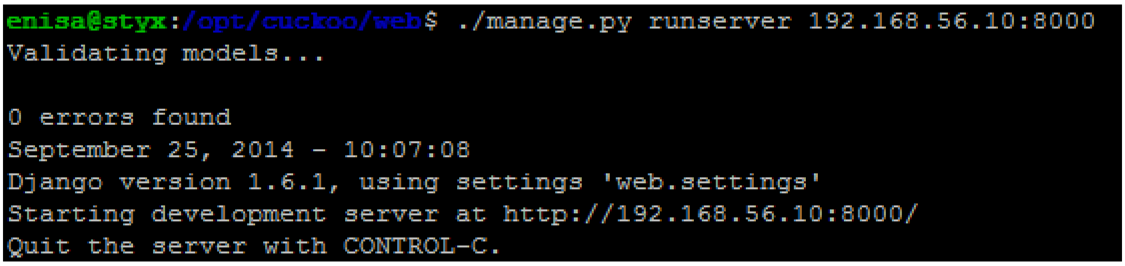
```
enisa@styx:/opt/cuckoo$ ./cuckoo.py

Cuckoo Sandbox 1.1
www.cuckoosandbox.org
Copyright (c) 2010-2014

2014-09-17 12:29:28,058 [lib.cuckoo.core.scheduler] INFO: Using "libvirt" machin
e manager
2014-09-17 12:29:28,193 [lib.cuckoo.core.scheduler] INFO: Loaded 1 machine/s
2014-09-17 12:29:28,193 [lib.cuckoo.core.scheduler] INFO: Waiting for analysis t
asks...
```

Figure 56: Cuckoo server start

To start the webserver cuckoo interface, run the *manage.py* script from the */opt/cuckoo/web* directory. (Figure 57).

Start Cuckoo web server

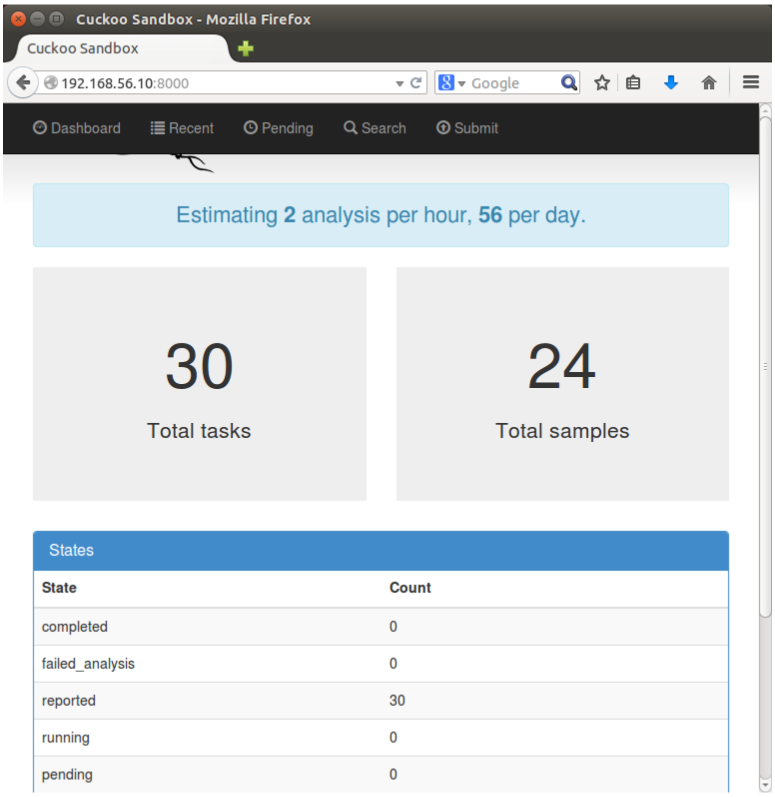
```
$ cd /opt/cuckoo/web
$ ./manage.py runserver 192.168.56.10:8000
```



```
enisa@styx:/opt/cuckoo/web$ ./manage.py runserver 192.168.56.10:8000
Validating models...

0 errors found
September 25, 2014 - 10:07:08
Django version 1.6.1, using settings 'web.settings'
Starting development server at http://192.168.56.10:8000/
Quit the server with CONTROL-C.
```

Figure 57: Cuckoo webserver start

When *manage.py* script has started, participants are able to open http://192.168.56.10:8000 on the host machine to access the Cuckoo web panel.



Estimating **2** analysis per hour, **56** per day.

30
Total tasks

24
Total samples

States

State	Count
completed	0
failed_analysis	0
reported	30
running	0
pending	0

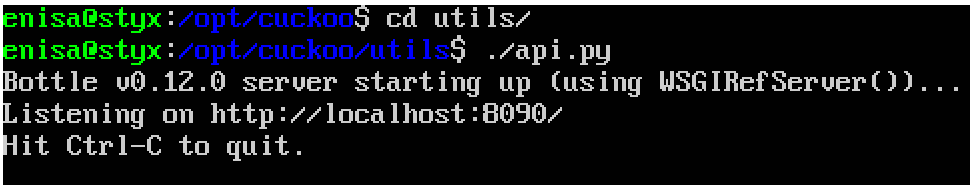
Figure 58: Cuckoo web panel at http://192.168.56.10:8000.

The Cuckoo Sandbox also offers API functionality allowing for example to remotely upload new samples. The Cuckoo API will be used in later exercises (it's not necessary to start it now). To start the Cuckoo API service execute the following commands:

Start Cuckoo API

```
$ cd /opt/cuckoo/utils
$ ./api.py
```

The Cuckoo API should then start listening on tcp/8090 port (Figure 59).



```
enisa@styx:/opt/cuckoo$ cd utils/
enisa@styx:/opt/cuckoo/utils$ ./api.py
Bottle v0.12.0 server starting up (using WSGIRefServer())...
Listening on http://localhost:8090/
Hit Ctrl-C to quit.
```

Figure 59. Cuckoo API listening on port 8090.

11 Analysis environment management

Before and after each analysis, certain tasks should be performed – starting network capture, clearing network simulator and snort logs, etc. Doing this each time by hand would be tedious. In this step, participants will learn how to automate certain tasks.

11.1 Starting analysis

Participants will write a simple script, which will be started before each manual analysis. This script will clear all necessary logs (snort, INetSim) and start network traffic capture in the background.

The script called *lab-cleanlogs* which is stored in */lab/bin* directory, cleans all logs from INetSim, Snort, MITMProxy and pcap files from /lab/var/pcaps. The script also restarts the INetSim service.

Before starting a new analysis, we must remove all previous results.

Execute */lab/bin/lab-cleanlogs* script

```
$ sudo /lab/bin/lab-cleanlogs
```

Source code of */lab/bin/lab-cleanlogs* script

```
#!/bin/bash

# Cleaning INetSim logs
# (restart needed for inetsim to write down old report)

sudo service inetsim stop
rm -f /lab/var/inetsim/*.log
rm -f /lab/var/inetsim/report/*
sudo service inetsim start

# Cleaning Snort logs
rm -f /lab/var/snort/*

# Cleaning Mimproxy dumps
rm -f /lab/var/mitmproxy/*

# Cleaning Pcaps
rm -f /lab/var/pcaps/*
```

11.2 Stopping analysis

Participants will write a simple script, which will be run after each completed analysis. This script will download result files from the virtual machine on which the analysis was conducted. Script *lab-getresults* can be used with the *–d* parameter to specify the destination folder.

Source code of */lab/bin/lab-getresults* script

```
#!/bin/bash

VM_HOST="10.0.0.2"
DIR="."

usage() {
    echo "Usage: $0 [-d <dir>]" 1>&2;
    exit 1;
}

while getopts ":d:h" opt; do
    case "${opt}" in
        d)
          DIR="$OPTARG"
          ;;
        h|*)
          usage
          ;;
    esac
done

echo "Storing results to: $DIR"
wget -q -r -P ${DIR} -nH --cut-dirs=1 ftp://${VM_HOST}/results/
```

11.3 Network script

Participants will write a script to switch network traffic. Through this script, participants can choose how network traffic will be routed from Winbox through the Styx machine. In this script, participants can use several arguments:

- none, which means no Internet access for *Winbox*,
- nat, for Internet access for *Winbox*,
- nat_mitmproxy for Internet access via the MITMProxy tool (only 80 and 443 port are redirected)
- INetSim or netsim for simulated Internet environment access,
- netsim_mitmproxy for simulated Internet environment access and MITMProxy
- tor for access through Tor network

Source code of */lab/bin/lab-switch-net* script

```
#!/bin/bash

# Restoring proper iptables file

case "$1" in
    none)
      # No access to the Internet from the analysis VM
      echo "Applying changes..."
      sudo iptables-restore < /lab/rules/none.v4
      ;;
    nat)
      # Normal access to the Internet through NAT
      read -p "Do you really want to allow Internet NAT access for VM? [y/n] " yn
      if [[ "$yn" == 'y' ]]; then
        echo "Applying changes..."
        sudo iptables-restore < /lab/rules/nat.v4
      fi
      ;;
    nat_mitmproxy)
      # Normal access to the Internet through NAT
      read -p "Do you really want to allow Internet NAT access and MITMProxy for VM? [y/n]
" yn
      if [[ "$yn" == 'y' ]]; then
        echo "Applying changes..."
        sudo iptables-restore < /lab/rules/nat_mitmproxy.v4
      fi
      ;;
    inetsim|netsim)
      # No access to the Internet. Traffic redirected to network simulator
      echo "Applying changes..."
      sudo iptables-restore < /lab/rules/netsim.v4
      ;;
    netsim_mitmproxy)
      # No access to the Internet. Traffic redirected to network simulator
      echo "Applying changes..."
      sudo iptables-restore < /lab/rules/netsim_mitmproxy.v4
      ;;
    tor)
      # Access to the Internet through TOR network
      echo "Applying changes..."
      sudo iptables-restore < /lab/rules/tor.v4
      ;;
    *)
      echo "Usage: $0 {none|nat|nat_mitmproxy|netsim|netsim_mitmproxy|tor}"
      exit 1
esac
```

11.4 Sending samples to the analysis machine

Using this script, participants can upload malware file from *Styx* to the Winbox machine. The script called *lab-sendfile* requires at least one argument – a path to a file or a directory.

The script *lab-sendfile* uses FTP protocol to upload files to the Winbox machine. The script logs in as user *anonymous* without any password to 10.0.0.2.

Source code of */lab/bin/lab-sendfile*

```
#!/bin/bash

VM_HOST="10.0.0.2"
USER="anonymous"
PASS="none"

DIR="."

usage() {
    echo "Usage: $0 <file|dir> [<file|dir> ...]" 1>&2;
    exit 1;
}

# Processing options

while getopts ":h" opt; do
    case "${opt}" in
        h|*)
          usage
          ;;
    esac
done
shift $(($OPTIND-1))

# Uploading files

while (( "$#" )); do
    if [[ -e "$1" ]]; then
        ncftpput -u "$USER" -p "$PASS" -m -R $VM_HOST /uploads/ "$1"
    else
        echo "Can't find file or directory: '$1'" 1>&2
        usage
    fi
    shift
done
```

12 Conclusions

In this exercise, participants have created and configured a laboratory for malware analysis. They have created a malware laboratory using best practices.

The malware laboratory was created using VirtualBox software on a host machine. In VirtualBox, two machines have been added – *Styx* and *Winbox*. The Styx machine is an Ubuntu Linux machine that contains the proper tools for network capture (MITMProxy) and packet analysis (Snort). *Styx* also contains other software like Cuckoo for malware analysis, Internet Services Simulation Suite and Tor tunnelling proxy. *Styx* also contains scripts in the /lab directory. These scripts automate certain actions like changing a network route to the Internet Services Simulation Suite, the Tor network, traffic through MITMProxy and so on because *Styx* is also a gateway for *Winbox*.

Winbox is a Microsoft Windows 7 machine, which contains two snapshots. In the first snapshot, participants prepare the operating system for static and dynamic malware analysis by installing required software. In the second snapshot of the Winbox machine, participants prepare the operating system for automatic malware analysis.

13 Tools repository

Name	Version	URL	System	Category
Yara	2.1.0	http://plusvic.github.io/yara/	GNU/Linux	Utils
ClamAV		http://www.clamav.net	GNU/Linux	Antivirus software
strings	-	-	GNU/Linux	Utils
objdump	-	-	GNU/Linux	Utils
hexdump	-	-	GNU/Linux	Utils
snort	2.0.2	http://www.snort.org	GNU/Linux	Utils

Name	Version	URL	System	Category
Greenshot	1.1.9.13	http://getgreenshot.org/	Windows 7	Utils
IDA Pro Free	5.0	https://www.hex-rays.com	Windows 7	Disassambler
ILSpy	2.2.0.1706	http://ilspy.net/	Windows 7	Decompiler
Exe2Aut	0.10	https://exe2aut.com/	Windows 7	Decompiler
JD-GUI	0.3.6	http://jd.benow.ca/	Windows 7	Decompiler
OllyDbg v1 (+ plugins)	1.10	http://www.ollydbg.de/	Windows 7	Debugger
Immunity Debugger	Tbd	http://www.immunityinc.com/	Windows 7	Debugger
WinDbg	6.3.9600	http://www.windbg.org/	Windows 7	Debugger
HxD	1.7.7.0	http://mh-nexus.de/en/hxd/	Windows 7	PE structure analysis
ImpRec	1.6 / 1.7f	http://www.woodmann.com/collaborative/tools/index.php/ImpREC	Windows 7	PE structure analysis
LordPE	1.41 (Deluxe b)	http://www.woodmann.com/collaborative/tools/index.php/LordPE	Windows 7	PE structure analysis
PEview	0.9.9	http://wjradburn.com/software/	Windows 7	PE structure analysis
CFF Explorer	III	http://www.ntcore.com/exsuite.php	Windows 7	PE structure analysis
Exeinfo PE	0.0.3.5 Beta	http://exeinfo.atwebpages.com/	Windows 7	PE structure analysis

PEiD	0.9.5	http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml	Windows 7	PE structure analysis
Resource Hacker	3.6.0	http://www.angusj.com/resourcehacker/	Windows 7	PE structure analysis
BinText	3.0.3	http://www.mcafee.com/us/downloads/free-tools/bintext.aspx	Windows 7	PE structure analysis
Dependency Walker	2.2.6000	http://www.dependencywalker.com/	Windows 7	PE structure analysis
PeStudio	8.30	http://www.winitor.com/	Windows 7	PE structure analysis
API Monitor	Alpha r13	http://www.rohitab.com/apimonitor	Windows 7	Dynamic analysis
Regshot	1.9.0.7	http://sourceforge.net/projects/regshot/	Windows 7	Dynamic analysis
Fiddler2		http://www.telerik.com/fiddler	Windows 7	Dynamic analysis
Wireshark	1.10.8	http://wireshark.org/	Windows 7	Dynamic analysis
Malzilla	1.2.0	http://malzilla.sourceforge.net/	Windows 7	Dynamic analysis
Sysinterlans suite	-	http://technet.microsoft.com/pl-pl/sysinternals/bb842062.aspx	Windows 7	Dynamic analysis
RootkitRevealer	1.71	http://technet.microsoft.com/en-us/sysinternals/bb897445.aspx	Windows 7	Rootkit detector
GMER	2.1.19357	http://www.gmer.net/	Windows 7	Rootkit detector
Upx	3.91	http://upx.sourceforge.net/	Windows 7	Packers
OfficeMalScanner	0.5	http://www.reconstructer.org/code.html	Windows 7	Microsoft Office analysis tool
PDF Stream Dumper	0.9.525	http://sandsprite.com/blogs/index.php?uid=7&pid=57	Windows 7	PDF analysis tool
Pdfid.py		http://blog.didierstevens.com/programs/pdf-tools/#pdfid	Windows 7 / Linux	
winpmem	1.4.1	http://sourceforge.net/projects/volatility.mirror/	Windows 7	Utils
ncat-portable	5.59BETA1	http://nmap.org/	Windows 7	Utils

14 References:

1	**5 Steps to Building a Malware Analysis Toolkit Using Free Tools**
	http://zeltser.com/malware-analysis-toolkit/ http://blog.zeltser.com/ http://zeltser.com/remnux/
	A nice blog with malware analysis materials by Lenny Zeltser. He also maintains REMNUX, an Ubuntu-based malware analysis Linux distribution
2	**Malware Analysis Lab – A Fast and Cost Effective "HowTo"**
	http://www.cybersquared.com/2012/06/malware-analysis-lab-a-fast-and-cost-effective-howto/
	Another illustrated how-to.
3	**Building a Malware Analysis Lab**
	http://www.windowsecurity.com/articles-tutorials/viruses_trojans_malware/Building-Malware-Analysis-Lab.html
4	**Forensic Live CD issues**
	http://www.forensicswiki.org/wiki/Forensic_Linux_Live_CD_issues
	Why a lab really needs a writeblocker.
5	**Belkasoft Live RAM Capturer**
	http://forensic.belkasoft.com/en/ram-capturer
	A tool for dumping memory, works with newest Windows systems, 64bit compatible
6	**Prioritizing Malware Analysis**
	http://blog.sei.cmu.edu/post.cfm/prioritizing-malware-analysis-309
	Article on the prioritization of malware analysis, can be used in discussion on the organizational part of the lab.
7	**Malware Analysis: Environment Design and Architecture**
	https://www.sans.org/reading-room/whitepapers/threats/malware-analysis-environment-design-artitecture-1841
	Not the newest (2007) but still relevant.

ENISA
European Union Agency for Network and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office
1 Vass. Sofias & Meg. Alexandrou
Marousi 151 24, Athens, Greece

PO Box 1309, 710 01 Heraklion, Greece
Tel: +30 28 14 40 9710
info@enisa.europa.eu
www.enisa.europa.eu