Today we are going to concentrate on setting up a malware lab.

Setting up a good malware test environment is very important when it comes to analysing malware, so I will give you some tips on how to do this well.
When we are talking about a malware test environment there are 4 essential components:

Malware labs need to be
• Easy to restore (to revert the changes made by the malware). We need to be able to alter the lab environment, and then reset.
• Have correct analysis tools (be configured with all of your analysis tools pre-installed, for speed of testing)
• Isolated (Isolated from infecting clean systems) BUT it must be easy to connect to the internet if needed (most modern malware needs the internet to run)
• Easy to upgrade / manage (Easy to add new software, easily extendable, etc). We need to be able to mould the lab to fit the threat. E.g. Does it need a Jabber server?

Personally I use 2 setups
• Basic, Portable Lab Setup (for when I’m on the move or carrying out investigations on-site)
• Full Professional Lab Setup (for thorough investigations. Ideally with remote access available)
Lets start by describing the Basic Lab Setup at a high level – and later on this lecture we will be walking through step by step how to create this

A basic portable lab setup consists of the following:

• Install a Windows 10 64-bit Virtual Machine on the host machine. There are two main pieces of software I recommend for this – either VMWare or VirtualBox. VirtualBox has the advantage of being free – but both are perfectly acceptable for our module
  • At this stage its important to pause and ask – does everyone know what a Virtual Machine (VM) is? If you are not familiar with a Virtual Machine, the concept of Snapshots, or have never used with VMWare or Virtual Box before I have included a link in the notes. A complete 101 guide to these is beyond this course – but luckily there are plenty of tutorials and overviews out there. It will also become clear as we go along
  • Virtual Box Overview: https://www.virtualbox.org/manual/ch01.html (All sections are useful – but especially 1.1, 1.2, 1.3, and 1.10.1)
• Once we have our Windows 10 VM Setup – we next want to strangely make it as vulnerable as possible, because we want to actually make it easier for malware to run in this environment. In order to do that we will first running all Windows Updates (as some of the tools need them). But we will then disable any further Windows Update, and also disable the Windows Firewall, AV and several other security settings.
• We also want to ensure at this point that we can isolate the network if needed (using the Host only Networking setting of VMWare and VirtualBox) – however by default I recommend to have the VM able to connect to the Internet, as most malware will need this
• Where possible, use a home DSL connection that allows you to change your external IP. Malware can block your IP if it realises you are monitoring it, and then your network is useless.
• Once we have an updated, but insecure, Windows environment to work with – we then proceed to add our analysis tools – and we’ll mention the various tools later in the lecture.
• At this point we now have a VM that has our TOOLS installed, that can be ISOLATED, and can be EXTENDED – this is one other key component left, the ability to quickly RESET your lab back to base settings after analysis – so its ready to analyse the next sample
  • To do this we simply take a Snapshot of the VM – this allows you to save the current status of the machine

There are a number of Pros, and Cons to the basic portable lab setup

• Pros
  • Very portable + quick to setup
  • Allows you to run multiple machines on one test machine (limited only by RAM)
• Cons
  • Limited setup
  • Having sniffing tools on the victim machine is not ideal
  • Some malware will behave differently if it knows its running in a Virtual machine

Having a portable system is often very useful – you do not always have access to a full lab when you have to analyse a sample – e.g. Helping remove malware from a friends PC, when travelling etc

Be very careful not to connect your portable lab to a production network – Worms love when you do that!

Finally one of the most common questions at this stage is “Where can I get an installer for Microsoft Windows?” Windows is of course commercial software, but you may have licences from your organisation. If not there are a number of Virtual Machines available with a 90 DAY LICENSE from Microsoft and we will be putting a link to those in the notes at the end of this lecture.

LINKS
VMWare – http://www.vmware.com
VirtualBox – http://www.virtualbox.org/
Next up we have a more Professional Lab setup, and this is one that is beyond what you are expected to do on this course – but if you are part of an InfoSec team for example, it's recommended to build one.

This lab setup copies many of the same principles of the personal lab we just described – but with everything scaled up. For many of the Virtual Machines you can actually run all of these on one server if you have a powerful enough one with enough RAM and Processor power.

- Again we have multiple VMs of different OSs / patch levels
- We have a configurable Linux Gateway
- We have Mail / DNS / Web Servers
- Again – Network must be isolated, but also ensure you can change the external IP address and connect to the Internet as needed.

Let's look at a diagram – Note all of these machines are Virtual (and all run your analysis tools)

- Here for example we have 2 Versions of Windows 10, one with the latest patches and all security settings installed – and another with a very low patch level, and security settings disabled
- You could also include a Server OS, or old legacy OS like Windows 7 – especially if you have some devices like this in your company
- Here I have also included other OS like Mac OS, or an Android Smartphone Emulator – using a tool like GenyMotion (https://www.genymotion.com/)
- A Linux gateway helps to prevent infection from Windows malware. Remnux is a good distribution (more in a minute).
- You can use IP Tables on the Gateway to redirect all SMTP traffic to the mail server so you can analyse SPAM, HTTP traffic to the web server so that you can fake responses and use your DNS server so you can lie about DNS results.
- Your ideal test network might not be able to incorporate all of this, but at least setup the basic network (and consider adding the gateway). The goal here is to emulate at a minimum the different sort of devices you have on your actual corporate network
- In addition to using Virtual machines, you can also use some physical machines as well. If a suspicious file behaves differently in a VM to a physical machine – that is definitely worthy of consideration. To be able to quickly reimage a Physical machine I recommend using:
  - dd (http://en.wikipedia.org/wiki/Dd_(Unix))
  - Deep Freeze (http://www.faronics.com/standard/deep-freeze/)

The Pros of this setup are obvious in that it is MUCH more thorough and full featured, but that comes with some big Cons as well. A setup like this requires substantially more powerful computers, and is also more difficult to setup and to maintain.
When it comes to what OS to use for the Gateway – again people have different preferences.

Personally I am a big fan of the REMNux linux distribution from Larry Zeltser. This is a distribution based on Ubuntu and it comes with a lot of very useful analysis tools installed on it. You can download it from [http://zeltser.com/remnux/](http://zeltser.com/remnux/)

Some of the features it includes are:
- Many tools for Static Analysis
- Reverse Engineering tools
- Document Analysis tools
- But especially for our Gateway use case – its has powerful tools such as FakeDNS, FakeMail, InetSim, FakeNet and various Proxy interception tools

InetSim in particular is very useful – it lets you simulate most web services.

Remember that you will also need 2 physical or virtual Network Interfaces on your gateway to be able to act as a bridge between your test network and the internet.

For this course we will not go over REMnux in detail, instead using a suite of tools for Windows called FakeNet. People have varying degrees of exposure to Linux, and it is something that is best learned at your own pace – but I definitely recommend to take a good look at REMnux after the course.
In our personal lab environments of course we do not have access to all of these Mail, DNS, Web servers and so on that we had in our Professional one.

However another tool called FakeNet-NG helps here. This is a tool that aids in the dynamic analysis of malicious software. The tool simulates a network so that malware interacting with a remote host continues to run allowing the analyst to observe the malware’s network activity from within a safe environment.

It allows for things such as intercepting all web and DNS traffic, and replying to the malware with responses of your own choosing.

The project is easy to install and use under Windows and we’ll be using it in our own Lab Setup.

Further Info:
https://github.com/fireeye/flare-fakenet-ng
With the high-level explanation of the lab setup and components discussed, let’s go through some additional tips for setting up a lab:

- **Plenty of RAM**: When building Virtual Labs, you never can have enough RAM. As a rough estimate, allow 4 GB for a Windows 10 Image, 2 GB for older versions, 1 GB for Linux, and 1 GB for the Physical Host.

- **USB Keys**: If you are using USB Keys to physically move content between machines in the lab, be careful with any that are introduced to the network—threat them as infected as soon as they are used with the malware setup, and then keep them away from clean machines.

- **Connecting to Internet**: Be very careful when connecting the lab network to the Internet—there can be legal issues of launching an attack on another company.

- **Document Everything**: Document your lab setup! No one likes writing documentation, but this is a crucial step in being able to maintain your lab setup.

And of course, **test your lab setup!**

- Ensure all machines/images can connect to each other (ping)
- Ensure all machine/images can connect to the Internet
- Ensure the gateway can monitor and manage all traffic

One last thing that may seem obvious— you should not have AntiVirus software in your test environment! This could prevent your malware from running—at least it should do if it is working correctly!
A common question is what tools should you install on your testing images. In a lot of cases this is a matter of personal taste. You will also want to install popular software such as browsers, Microsoft Office (if you have a licence), IM clients and so on.

Over this course we will cover many tools that will help you in analysing malware, and on each lecture we give the links for the major ones – as well as links for optional ones also.

Rather than list all of the tools again here now in these notes, we will shortly go over a step by step guide that will install all the core tools for this course. As you go through the course you might find that you want to add some of the alternative or optional tools as well, which can simply be added to your Lab environment.
There is one final note I would like to make on malware analysis tools - which is unfortunate – but important to know. Malware Analysis tools tend to follow a certain cycle.

1. A lot of the best malware analysis tool are generally free, and created by talented folks as a hobby to give back to the community. Some fall outside this definition such as IDA, or the tools from SysInternals which are maintained by Microsoft
2. Those tools become actively maintained for a good while by that one person. However that person then becomes well known and gets a nice job (rightfully) based on the tool they wrote
3. Person now has less time to maintain tool due to work commitments and updates become less frequent
4. Person now reaches late 20s early 30s, gets married / has a kid. Time to work on tool drops to zero
5. Tool continues to be publically available with no updates, but as OS or malware change it becomes less useful
6. Eventually several years later someone will get tired of the old tools being bad, and will write their own replacement tool - and the cycle restarts

Right now the packer detection tool PEID is around the last stage for example. You’ll still see it referenced all the time in online documentation, but other tools we will cover a bit later on in the course have risen to take its place

NOTE: This long lifetime, and lack of updates of tools – can regularly lead to them needing older versions of certain libraries or coding languages to function. For example it is not uncommon for malware analysts to run older versions of Python (2.X) on their machines just to support older tools and scripts that will never be updated
Now that we have covered the concepts – let’s jump in and do a step by step guide to get your Malware Lab Setup!

Setting up a malware lab is not a one time thing. For starters tools of course evolve or update over time, and it’s a good idea to upgrade your lab at least once a year. But you will find the act of building a personal malware lab to become a bit of a recurring theme at different points in your careers.

The method we are going to take requires several hours to do – and it’s best to set aside half a day. HOWEVER very little of that is manual work. As you will see creating a malware lab today is a task that has become heavily automated – so it’s more a case of waiting for scripts to run, downloads to happen, and tools to install – rather than manually having to do things.

NOTE WHEN SETTING UP THE LAB – FOLLOW EVERY STEP IN THIS GUIDE IN ORDER (THINGS MISSED TEND TO CAUSE LATER STEPS TO FAIL)
For the first step we are going to install VirtualBox. If you have a copy of VMWare Workstation instead you can also perfectly use that (I personally have VMWare Fusion on my Mac) – but we will use VirtualBox on the course as it is free to use, and honestly just as good for our purposes. Also while it is possible to get free version of VMWare – the free versions to not support snapshots which are critical for malware analysis – but Virtualbox does.

Installing Virtualbox itself is very straightforward. Dowload the latest version for your OS (Windows, Mac or Linux) from https://www.virtualbox.org/

Installing Virtualbox is as easy as any other installer, but if you get stuck you can find detailed documentation in the notes at https://www.virtualbox.org/manual/UserManual.html

You don’t need to install any extensions or anything – just the base install is fine. Once that’s done you do not need to create a Virtual Machine yet – we will be doing that in the next step
For our lab we will be using Windows 10 64 bit. If you have access to a licensed version of the software – by all means you can use that here, and follow the instruction in Section 1.7 of the Virtual Box documentation (“Creating your first Virtual Machine”) to get yourself setup.

If you don’t have a licensed version of Windows for your virtual machine, you can download a free Windows 10 VM from Microsoft. Go to the Microsoft Edge page for downloading virtual machines on [https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/](https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/). Select “MSEdge on Win 10 (x64)” and pick the virtualization platform that matches the one you have i.e. VirtualBox

After downloading and extracting the archive, follow the steps appropriate for your virtualization software to start the VM. For example, for VMware you’d extract the files into a dedicated folder, then launch the file named “MSEdge – Win10.vmx”. For Virtualbox you can again follow the instructions, but this time Section 1.14 of the guide to add it to your installation.

Just a small note on settings at this point:
- The Windows OS in this VM expires after 90 days.
- The password Microsoft assigned to this virtual machine is “Passw0rd!” , in case you need it when logging in
- If you can afford 4GB of RAM do so, but Windows 10 will run with 2GB
- Before starting up the virtual machine – go to the VM Settings (see the documentation) and set its Hard Disk size to 60 GB. That seems like a lot – but Windows 10 with all tools installed is a hefty setup. You can also keep you Virtual machine on an external hard drive if you are low on space, in which case try for a USB 3.0 or better to not have as big a speed hit. NOTE: The VM needs to be in a powered off state (shutdown properly) otherwise the option to change the Hard Disk size will be greyed out
- You can also leave the Networking settings at the default for now so that it can connect properly to the Internet, instead of the other more isolated options such as “Host only Networking”
- If you are struggling to find a way to free up 60GB of space – a backup plan can be to use one of the Windows 7 VM instead – which will need about 30GB of Space. The steps will be broadly similar, but we will be assuming Windows 10 for the rest of the instructions and supporting that setup throughout the course.
Now that you have your VM installed the next step is to Update it. To do this simply power on the VM for the first time and login with the password of “Passw0rd!”. Make sure you are connected to the internet.

Next use the search menu (Magnifying glass down the bottom) and Check for Updates. Download and apply all Windows Updates that are needed to get the VM fully up to date.

This likely will require a reboot, and once done – run Updates again – and repeat this cycle until you have a full up to date Windows VM.
Now that we have gotten our Windows VM up to date – we want to make it as insecure as possible 😊 Or at least we want to make Windows Security settings as little of a hindrance as possible.

This step is actually a little trickier than it looks – as Windows really, really wants to keep its AV running it seems. Luckily there is some great guides to do this, and I recommend the following steps in Order:

2. While in the Security Center I recommend to also disable the following settings:
   2. Firewall & Network Protection -> Domain, Private, and Public Network firewalls
   3. App & Browser Control -> Check Apps & Files, SmartScreen for Microsoft Edge and Microsoft Store
3. Go to Windows Update and Disable Automatic Updates

These settings are partially to prevent things that might mess up you malware – and partially to prevent updates and other activities running during you analysis that might pollute your logs.

Finally its time to change your username – for two reasons. Firstly the default use of IEUser is known by some malware to be used in Sandboxes and analysis machines, and secondly so everyone on the course has a unique but easy to identify Username which helps enormously for debugging any problems is setup later.

In Windows use search for “control” to access classic Control Panel, select User Accounts. Then select “Manage another account, “IEUser” “ and change your account name to YOUR COLLEGE STUDENT NUMBER

At this point Reboot your VM (needed for the Group Policy effects to apply properly), and then create a “Snapshot” (i.e. Saved State) of your VM using the steps in the documentation. Take a rest – you’ve earned it. Most of the hard work is done!
At this point your hard work is done, and now your friends at FireEye are going to take over and help you with the rest of your setup 😊

In essence what FireEye have done is to create a project called Flare that makes it easy to install many of the analysis tools used today for modern Malware Analysis, and which we will be using throughout the course. In fact they will have installed all but a handful of tools that we can add at the end ourselves. Flare VM makes use of a solution called Malboxes is which helps to quickly deploy a Windows Malware test environment, and a package manager called Chocolately.

Installation could not be easier – head on over to https://github.com/fireeye/flare-vm and follow the instructions in the “Installation (Install Script)” (you have already done a snapshot so skip that part). Be sure to open Powershell as Administrator (to do this search for Powershell and Right Click -> Run as Administrator). Finally run the install.ps1 script

At this point Flare is going to spend possibly several hours depending on your internet speed download lots and lots of tools and configuring setups – but it should take care of everything itself, including reboots. So at this point put the kettle on, grab a coffee or tea (actually several as it will take a while – but it’s a lot faster than having to do all this manually!)
At this there is very little left to do – just manually install a handful of extra tools mentioned in the notes that are not included in the Flare Setup. Each of these has their own setup, but many are standalone tools also.

For each of these tools I have included a link in the notes, and also the recommended folder to copy them to (or a shortcut) from under the “C:\ProgramData\Microsoft\Windows\Start Menu\Programs\FLARE”

**TOOLS:**
- Decompilers
  - IIftenor - [http://il4re.ml/](http://il4re.ml/)
- Forensics
  - Trend Micro ATTK – (64 Bit Online Version)
- Net
  - Captipper - [https://github.com/omriher/CapTipper/tree/python3_support](https://github.com/omriher/CapTipper/tree/python3_support)
  - (NOTE: This specific version is needed, not the old version)
  - Winscp - [https://winscp.net/eng/download.php](https://winscp.net/eng/download.php)
- OSINT (add this)
  - Maltego - [https://www.maltego.com/downloads/](https://www.maltego.com/downloads/)
- Utilities
  - ExeInfoPE - [http://exeinfo.byethost18.com/?i=1](http://exeinfo.byethost18.com/?i=1)
  - Trid - [https://mark0.net/soft-trid-e.html](https://mark0.net/soft-trid-e.html)
- Web Application
  - Chrome – (actually in Flare, but sometimes does not install correctly)
  - Tor Browser - [https://www.torproject.org/download/](https://www.torproject.org/download/)
And now the finishing line is in sight!

If you wish, install in the VM utilities such as VirtualBox Guest Additions and VMware Tools, which come with the VM software. They will make it easier to share clipboard contents and files between your host machine and the VM.

At this stage I recommend to do one more Reboot, and to check once again is the Windows AV is still disabled (and re-disable it if not).

Then finally take one more final snapshot – and you can remove your initial one to save space.

And you’re done! All of your hard work (well a lot of it watching Flare do its thing) is over, and we are ready to analyse some malware!
### Required Exercise – Setup a Lab

- Setup your own Personal Lab *(Required for all of the rest of this course)*

For this week’s Exercise the goal is simple – setup your own copy of a Personal Lab with the guide detailed in the Step by Step guide section of this lecture.

Unlike other Exercises on the course which are optional this exercise is a Requirement, as the lab will be used constantly for the rest of the module. So set aside some time this week and try to get it setup.