

# RHCE BOOT CAMP

Web Services



redhat.®

---

**CERTIFIED**  
**E N G I N E E R**



# APACHE CONFIGURATION

- The main apache configuration file is `httpd.conf` and is found in `/etc/httpd/conf/`. This configuration file stores the core configuration of the web server.
- In Apache 2, the `/etc/httpd/conf.d` directory stores configurations that are specific to a particular Apache module. All files in this directory ending in `.conf` will be parsed as a configuration file.



# APACHE CONFIGURATION

- You can find this example Apache VirtualHost definition at the bottom of `httpd.conf`:

```
<VirtualHost _____>  
    ServerName name  
  
    ServerAlias alias  
  
    DocumentRoot path  
  
    CustomLog /path/to/access_log combined  
  
    ErrorLog /path/to/error_log  
  
</VirtualHost>
```

- The `NameVirtualHost` directive **must be used** to specify an IP that can host multiple websites.



# LAB

1. Configure two websites on your server. “X” represents your station #.
2. `wwwX.example.com` should be served from `/var/www/html` and should also respond to requests for the short hostname `wwwX`.
3. `vhostX.example.com` should be served from `/home/linus/html` and should also respond to requests for the short hostname `vhostX`.
4. Both should be listening on your primary ip address, but `wwwX.example.com` should be the default site.



# SECURING APACHE

- Apache support access control through allow and deny directives:
  - `allow from <host|network|ALL>`
  - `deny from <host|network|ALL>`
- These can be applied in the given order:
  - `order allow,deny` Allows explicitly allowed clients and **denies everyone else**. Anyone matching both the allow and deny are denied.
  - `order deny,allow` Denies explicitly denied clients and **allows everyone else**. Anyone matching both the allow and deny are allowed.



# SECURING APACHE

- These access control directives are applied through a per-Directory or per-File basis.
- The `allow`, `deny` and `order` directives are placed inside one of the following tags:
  - `<Directory>`
  - `<File>`



# LAB

1. Reconfigure your two websites such that:
  - `wwwX.example.com` is accessible to everyone except for the person sitting to your left.
  - `vhostX.example.com` is only accessible to the person sitting to your right.



# CGI SCRIPTING

- Scripting involves making Apache *execute* a file and return it's *output*, as opposed to simply returning the file itself.
- There is an entire framework for facilitating this operation, and allowing the webserver to communicate basic information to script through the use of environment variables, and sometimes input.
- This is known as CGI scripting, or Common Gateway Interface scripting.



# BASIC SCRIPTING

- Some of the simplest scripting requires only a shell script.  
Consider:

```
#!/bin/bash
```

```
echo -e "Content-type: text/html\n"
```

```
echo "<h1>Hello world!</h1>"
```



# BASIC SCRIPTING

- If we put the appropriate execute permissions on the script, then we can see it output the expected content at the command line:

```
# chmod +x myscript
```

```
# ./myscript
```

```
Content-type: text/html
```

```
<h1>Hello world!</h1>
```



# BASIC SCRIPTING

- If this file is placed in a location identified to Apache as supporting executables ( CGI scripts ), then we have a working CGI!



# LAB

1. Install `httpd-manual` if you have not already done so.
2. Look up the `ScriptAlias` directive in the manual.
3. Use this directive and your simple shell script to create a simple, dynamic webpage. Maybe have it report the current date and time with the `date` command.



# SQUID

- Squid is designed to cache internet objects and can act as a proxy server for HTTP, FTP, and many other types of requests.
- Squid is highly flexible and powerful, but for the RHCE exam, you only need to demonstrate the ability to set it up and proxy web services, possibly denying access to a given subnet.
- The configuration file for Squid is

`/etc/squid/squid.conf`



# KEY SQUID SETTINGS

- **http\_port** *3128 by default*
- **visible\_hostname** *the hostname Squid broadcasts*



# KEY SQUID SETTINGS

- Access control in squid is handled via ACL definitions coupled with access definitions, as:

```
acl mynet src 192.168.0.0/255.255.255.0
```

```
acl yournet src 192.168.1.0/255.255.255.0
```

```
http_access allow mynet
```

```
http_access deny yournet
```

- Look for “HERE” in the config file. This is the best place for new ACL entries.



# LAB

1. Configure your server to offer Squid proxy service to the person sitting on your right, but not to the person sitting on your left.
2. This service should listen on port 8080.



```
slideshow.end();
```