

This PDF Lecture is made by
Abid Biswas, **Ethical Hacker (Purple Teaming)**
(CEH, NSE1, NSE2, NSE3,
ISC²(CC), Google Cyber Security Professional Cert
BSc in Information and Communications Engineering (ICE) at EWU



To install Apache2 Web Server on Ubuntu Linux Platform:

We can even install Apache web server on windows

sudo apt-get install apache2

What's going to happen is it's going to reach out to any package repositories configured for this specific machine and pull that down over the Internet. Now naturally, from a security standpoint, web application security begins at the foundation. It means making sure the operating system, in this case, Ubuntu Linux, is kept up-to-date, and configured securely, including install packages being updated like the Apache2 web server.

So, what we can do is we can always run sudo apt-get update to make sure that our list of package repositories is up to date and you don't have to point to the Internet. Your organization if it uses Linux extensively, might have local private repositories with carefully curated software packages that are made available to install on Linux hosts internally. We can also run the sudo apt-get upgrade command to upgrade.

~\$ sudo apt-get update

\$ sudo apt-get upgrade

So, we know that this is an important part of web application security, at this level, the operating system and web server stack level.

So, the next thing to do then is to make sure that the Apache web server, daemon or background service is up and running. I can do that with sudo service, it's called apache2.

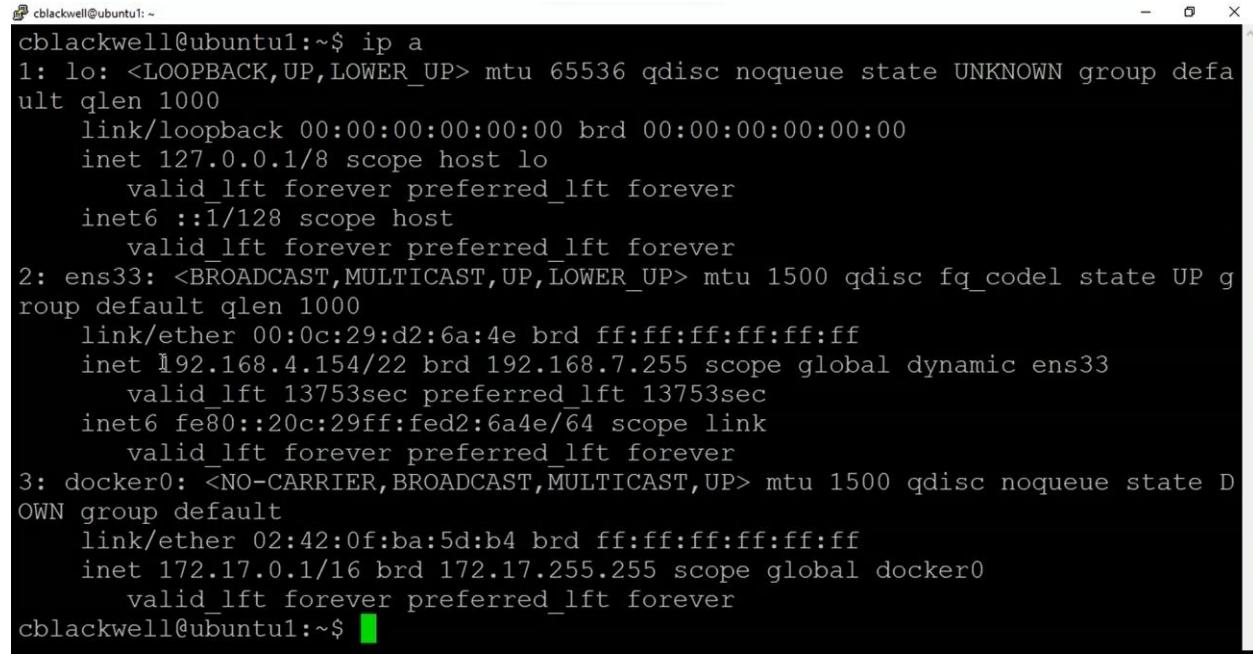
~\$ sudo service apache2 status

```
cblackwell@ubuntu1:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pre>
   Active: active (running) since Sun 2022-04-24 16:25:07 UTC; 8min ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 2112 (apache2)
    Tasks: 55 (limit: 4575)
   Memory: 7.8M
      CPU: 0.100 seconds
     CGroup: /system.slice/apache2.service
             └─2112 /usr/sbin/apache2 -k start
                  ├─2113 /usr/sbin/apache2 -k start
                  ├─2114 /usr/sbin/apache2 -k start
                  └─2115 /usr/sbin/apache2 -k start

Apr 24 16:25:07 ubuntu1 systemd[1]: Starting The Apache HTTP Server...
Apr 24 16:25:07 ubuntu1 apachectl[2109]: AH00558: apache2: Could not reliably>
Apr 24 16:25:07 ubuntu1 systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

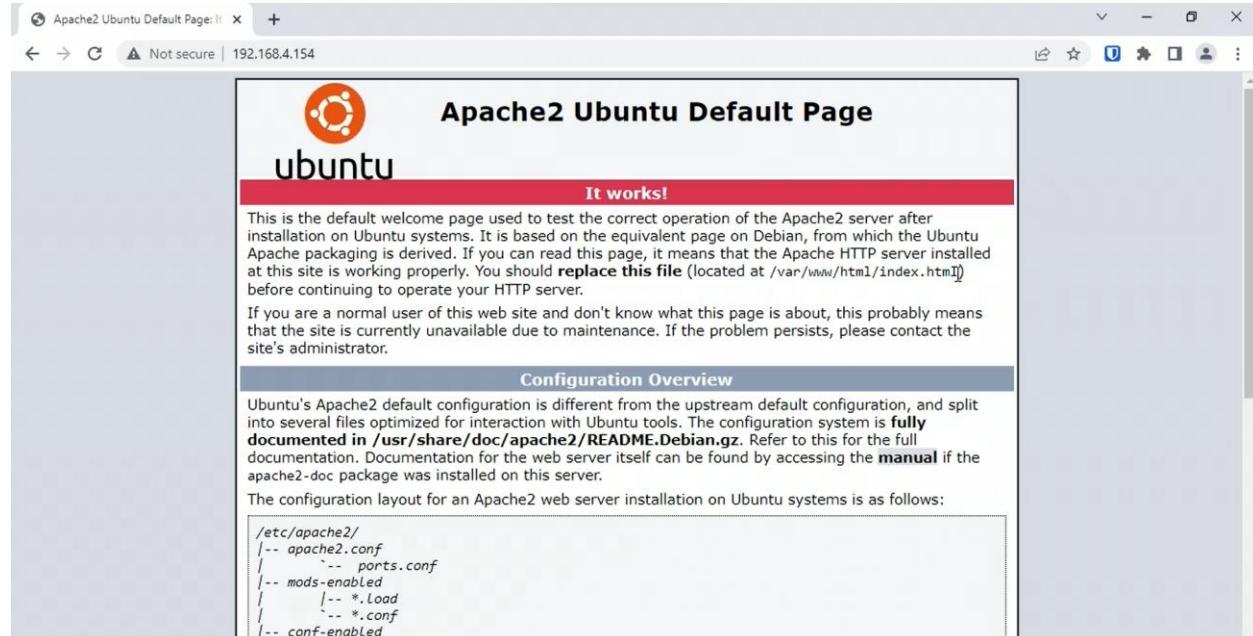
and then I'll pass it the status parameter. So, we get a return to listing that shows us that the Apache2 web server daemon is active and up and running, so the Apache HTTP server then has been started. **Press Q**

for quit. I'm going to run `~$ ip a`, IP address which will show me my local loopback IP along with any other IPv6 and IPv4 addresses for this host. For example, I'm interested in the IPv4 address 192.168.4.154.



```
cblackwell@ubuntul:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:d2:6a:4e brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.154/22 brd 192.168.7.255 scope global dynamic ens33
        valid_lft 13753sec preferred_lft 13753sec
    inet6 fe80::20c:29ff:fed2:6a4e/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:0f:ba:5d:b4 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
cblackwell@ubuntul:~$
```

Why don't we open up a web browser on this network on a different host and just try to connect to the Apache web page? So, when I do that, it opens up the Apache2 Ubuntu default web page, and it says it works. So, we know then that the default web page configuration for that web server is now up and running on our Ubuntu Linux host.



And, of course, we've also patched it to make sure we've got the latest updates applied. Now, the great thing about a lot of these web server default web page is they also serve double duty as not only a test page, but

also a configuration page where it gives you some examples of what to do. Here **it shows us the configuration file layout on this host for the Ubuntu Apache2 web server, such as going into /etc/apache2**, let's explore that a little bit.

So, back here in Linux, I'm going to **change directory to /etc/apache2**. I'll clear the screen, I'll do an **ls**. So, we've got the apache2.conf file along with the ports.conf file.

```
cblackwell@ubuntu1: /etc/apache2
cblackwell@ubuntu1:~$ cd /etc/apache2
cblackwell@ubuntu1:/etc/apache2$
```

```
cblackwell@ubuntu1: /etc/apache2
cblackwell@ubuntu1:/etc/apache2$ ls
apache2.conf      conf-enabled  magic          mods-enabled  sites-available
conf-available   envvars       mods-available  ports.conf    sites-enabled
cblackwell@ubuntu1:/etc/apache2$
```

```
cblackwell@ubuntu1:/etc/apache2$ ls
apache2.conf      conf-enabled  magic          mods-enabled  sites-available
conf-available   envvars       mods-available  ports.conf    sites-enabled
cblackwell@ubuntu1:/etc/apache2$ cat ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
I
<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cblackwell@ubuntu1:/etc/apache2$
```

it defines the listening ports for the web server. **Here it's listening on port 80, if we've got the ssl_module enabled for the web server with the PKI certificate, then it's going to listen by default on port 443, but we know that SSL really is deprecated, should never be used. Instead, we should be using TLS, Transport Layer Security. Now, if the ssl_module is not enabled for the web server stack, then it won't be listening for that type of connection on 443 using the SSL network security protocol. So, we could even comment out those lines if we wanted to. But really, as long as we don't enable the ssl_module, we're good.**

Now speaking of that, why don't we take a look at the **apache2 conf file**, the config file, **the main config file for the web server**. So, I'll clear the screen, I'm actually **going to use the nano editor to open it up and it's called apache2.conf**.

```
: /etc/apache2$ nano apache2.conf
```

Comments are noted on the line with a hashtag symbol or a pound symbol. That means that line is a comment, it is thought to be interpreted as anything such as a configuration directive. So, notice for example, the ServerRoot directive is commented out, the default is /etc/apache2.

```
ServerRoot "/etc/apache2
```

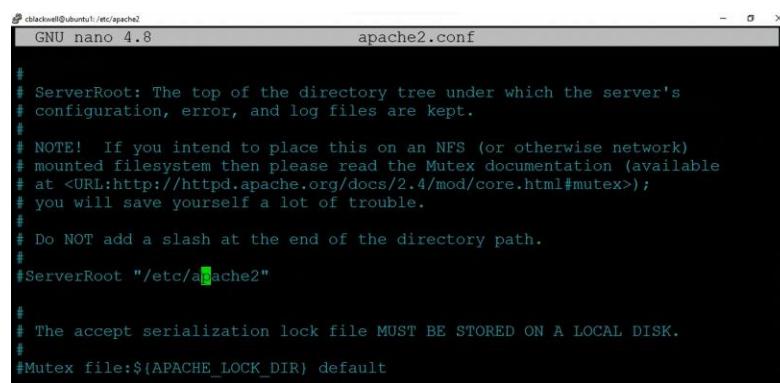
We then have the **ErrorLog** directive, which is using a built-in APAHE_LOG-DIR variable /error.log as the default **error log** unless otherwise specified for the Apache web server.

```
ErrorLog ${APACHE_LOG_DIR}error.log
```

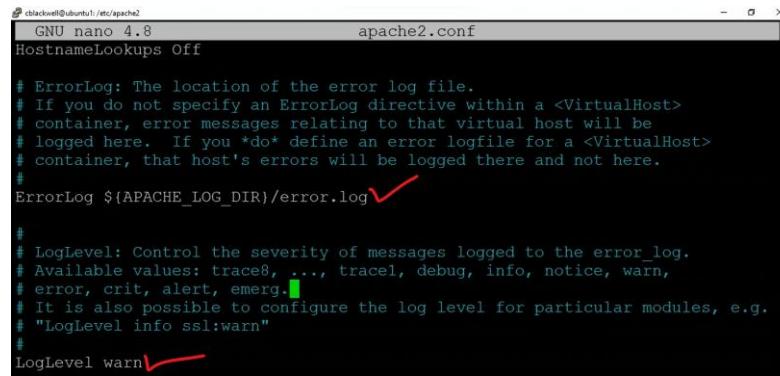
```
IncludeOptional mods-enabled/* .load
```

```
IncludeOptional mods-enabled/* .conf
```

The LogLevel is set to warning. Then we've got an Include module section for additional modules that provide additional functionality for the web server.



```
cbawell@ubuntu1:/etc/apache2$ nano 4.8 apache2.conf
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the Mutex documentation (available
# at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
#ServerRoot "/etc/apache2"
#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
#Mutex file:${APACHE_LOCK_DIR} default
```



```
cbawell@ubuntu1:/etc/apache2$ nano 4.8 apache2.conf
HostnameLookups Off

# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here. If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog ${APACHE_LOG_DIR}/error.log✓

#
# LogLevel: Control the severity of messages logged to the error_log.
# Available values: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.■
# It is also possible to configure the log level for particular modules, e.g.
# "LogLevel info ssl:warn"
#
LogLevel warn✓
```

emerg: This is the most severe level, indicating an emergency condition where the server cannot continue to run.

alert: A condition that should be corrected immediately, but the server can still continue to run.

crit: A critical condition that might require immediate attention.

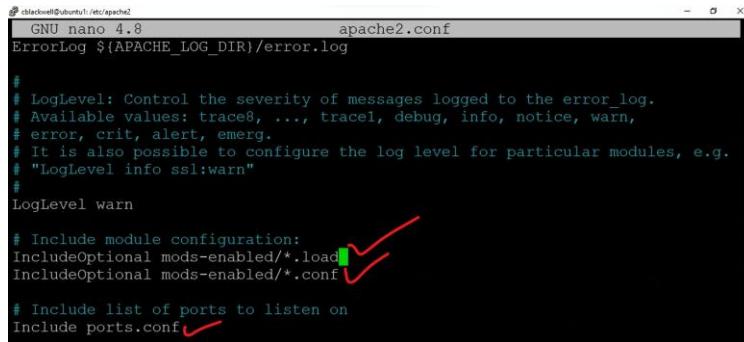
error: A general error condition.

warn: A warning message, indicating a potential problem that does not prevent the server from running.

notice: A notice message, providing general information about server activity.

info: Informational messages about the server's operation.

debug: Detailed debugging information for developers.



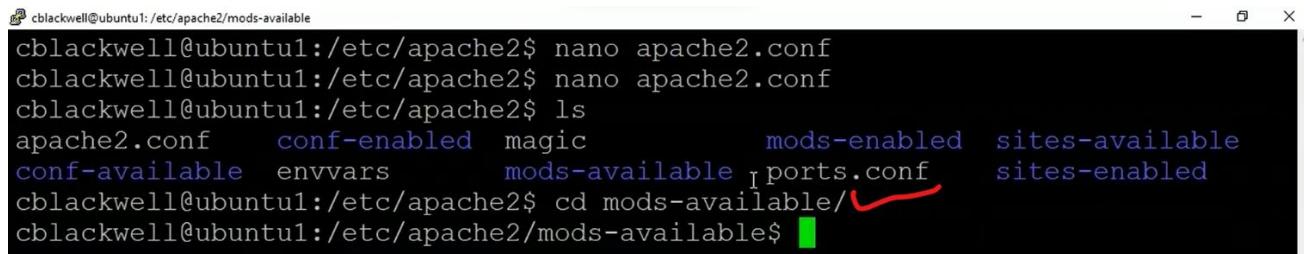
```
GNU nano 4.8 apache2.conf
ErrorLog ${APACHE_LOG_DIR}/error.log

#
# LogLevel: Control the severity of messages logged to the error log.
# Available values: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the log level for particular modules, e.g.
# "LogLevel info ssl:warn"
#
LogLevel warn

# Include module configuration:
IncludeOptional mods-enabled/*.load ✓
IncludeOptional mods-enabled/*.conf ✓

# Include list of ports to listen on
Include ports.conf ✓
```

Then we've got an **Include module section** for additional modules that provide additional functionality for the web server. All the files that end with .load or .conf in the mods enabled location, which would determine things like whether SSL or TLS is being used. Whether we have some kind of Java server-side components that are up and running and notice it's **also including the ports.conf file** that we just looked at. As we go further down through the apache2.conf file, we also have some options for what permissions are allowed for different subdirectories on the web server. So I'm going to **press Ctrl+X to close out of that file**. We haven't made any changes.



```
cblackwell@ubuntu1:/etc/apache2$ nano apache2.conf
cblackwell@ubuntu1:/etc/apache2$ nano apache2.conf
cblackwell@ubuntu1:/etc/apache2$ ls
apache2.conf      conf-enabled      magic          mods-enabled    sites-available
conf-available   envvars           mods-available  ports.conf     sites-enabled
cblackwell@ubuntu1:/etc/apache2$ cd mods-available/ ✓
cblackwell@ubuntu1:/etc/apache2/mods-available$
```

ls

And if I were to do an ls in there, we've got all kinds of files that end with either .conf or .load as per our apache2.conf file. For example, we have an ssl.conf and an ssl.load file.

```
chadwell@ubuntu1:/etc/apache2/mods-available
```

Module	Status
autoindex.conf	log_debug.load
autoindex.load	log_forensic.load
brotli.load	lua.load
bufuffer.load	macro.load
cache_disk.conf	md.load
cache_disk.load	mime.conf
cache.load	mime.load
cache_socache.load	mime_magic.conf
cern_meta.load	mime_magic.load
cgid.conf	mpm_event.conf
cgid.load	mpm_event.load
cgi.load	mpm_prefork.conf
charset_lite.load	mpm_prefork.load
data.load	mpm_worker.conf
dav_fs.conf	mpm_worker.load
dav_fs.load	negotiation.conf
dav.load	negotiation.load
dav_lock.load	proxy_ajp.load
dbd.load	proxy_balancer.conf
default.conf	proxy_balancer.load
	session.load
	setenvif.conf
	setenvif.load
	slotmem_plain.load
	slotmem_shm.load
	socache_dbm.load
	socache_memcache.load
	socache_redis.load
	socache_shmcb.load
	speling.load
	ssl.conf✓
	ssl.load✓
	status.conf
	status.load
	substitute.load
	suexec.load
	unique_id.load
	userdir.conf
	userdir.load
	workerload.load

So, let's open up the ssl.conf file, notice **there's not a TLS file**. These days, there's a lot of old references to SSL, even though we would be using its superseding network security protocol TLS, so I'm going to open up the **ssl.conf file**. I'm going to use the sudo prefix, so I can write to it. So I'm going to use the nano text editor built into Ubuntu Linux, ssl.conf.

```
/etc/apache2/mods-available$ sudo nano ssl.conf
```

```
GNU nano 4.8          ssl.conf
# SSL server cipher order preference:
# Use server priorities for cipher algorithm choice.
# Clients may prefer lower grade encryption. You should enable this
# option if you want to enforce stronger encryption, and can afford
# the CPU cost, and did not override SSLCipherSuite in a way that puts
# insecure ciphers first.
# Default: Off
#SSLHonorCipherOrder on

#   The protocols to enable.
#   Available values: all, SSLv3, TLSv1, TLSv1.1, TLSv1.2
#   SSL v2 is no longer supported
#SSLProtocol all +TLSv1.2
#           I
#   Allow insecure renegotiation with clients which do not yet support
#   secure renegotiation protocol. Default: Off
#SSLInsecureRenegotiation on

#   Whether to forbid non-SNI clients to access name based virtual hosts
```

What I'm interested in doing is making sure that the only protocol that is allowed in my particular case is **TLSv1.2**. Now notice the SSLProtocol directive here currently has **all** and the **+** means we want to also add support for **TLSv1.2**. You could also alternatively change that to **-** if you wanted to remove something or I could just say **SSLProtocol**. I'm not going to use the word **all**, but instead **+TLSv1.2** to make sure only that network security protocol version is to be used for this particular web server. Now, syntactically, you could also just remove the plus sign, which implies that we only want to support TLSv1.2 on this host. Just bear in mind that if we want to support a newer version, such as **TLSv1.3**, we have to bear in mind that our configuration will have to change because we'll have to reference it in the ssl.conf file.

So, I'm going to go ahead and press **Ctrl+X** it asks to save the modified buffer I'll type in the letter **Y** I want to overwrite the same file, I'll just press **Enter** and it's done. At least the config is done, but it's not in effect until I restart the Apache2 web server daemon. Which I'll do with this you sudo service apache2 restart. I'll just use the Up arrow key to bring up that previous command, and I'm going to change restart to status.

```
cblackwell@ubuntul:/etc/apache2/mods-available$ sudo service apache2 restart
cblackwell@ubuntul:/etc/apache2/mods-available$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pre>
   Active: active (running) since Sun 2022-04-24 16:49:29 UTC; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Process: 39187 ExecStart=/usr/sbin/apachectl start (code=exited, status=0>
 Main PID: 39209 (apache2)
   Tasks: 55 (limit: 4575)
  Memory: 5.8M
    CGroup: /system.slice/apache2.service
            └─39209 /usr/sbin/apache2 -k start
              ├─39210 /usr/sbin/apache2 -k start
              ├─39211 /usr/sbin/apache2 -k start
              └─39212 /usr/sbin/apache2 -k start

Apr 24 16:49:29 ubuntul systemd[1]: Starting The Apache HTTP Server...
Apr 24 16:49:29 ubuntul apachectl[39208]: AH00558: apache2: Could not reliably
Apr 24 16:49:29 ubuntul systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

⚠️⚠️⚠️ This document is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this document, or any portion of it, may result in severe civil and criminal penalties and will be prosecuted to the maximum extent possible under the law.

For permissions to use or reproduce this document or for any questions related to copyright, please contact:
abidbiswas2021@gmail.com