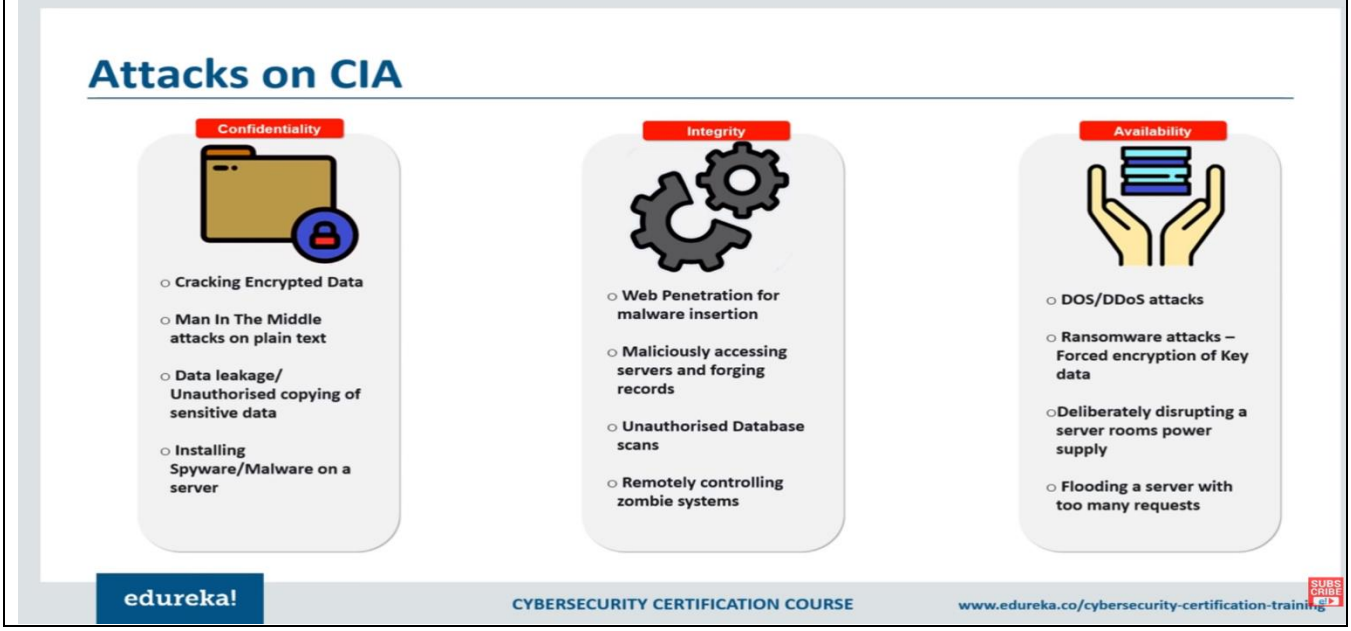
DAILY ASSESSMENT FORMAT

Date:	16th June 2020	**Name:**	Soundarya NA
Course:	Cyber security	**USN:**	4AL16EC077
Topic:	Cyber security	**Semester & Section:**	8th - B

FORENOON SESSION DETAILS

Image of session



Report:

Definition of Cyber Security:

Cyber security refers to the body of technologies, processes, and practices designed to protect networks, devices, programs, and data from attack, damage, or unauthorized access. Cyber security may also be referred to as information technology security.

Importance of Cyber Security:

Cyber security is important because government, military, corporate, financial, and medical organizations collect, process, and store unprecedented amounts of data on computers and other devices. A significant portion of that data can be sensitive information, whether that be intellectual property, financial data, personal information, or other types of data for which unauthorized access or exposure could have negative consequences. Organizations transmit sensitive data across networks and to other devices in the course of doing businesses, and cyber security describes the discipline dedicated to protecting that information and the systems used to process or store it. As the volume and sophistication of cyber-attacks grow, companies and organizations, especially those that are tasked with safeguarding information relating to national security, health, or financial records, need to take steps to protect their sensitive business and personnel information. As early as March 2013, the nation's top intelligence officials cautioned that cyber-attacks and digital spying are the top threat to national security, eclipsing even terrorism.

Challenges of cyber security:

For an effective cyber security, an organization needs to coordinate its efforts throughout its entire information system. Elements of cyber encompass all of the following:

Network security: The process of protecting the network from unwanted users, attacks and intrusions.

Application security: Apps require constant updates and testing to ensure these programs are secure from attacks.

Endpoint security: Remote access is a necessary part of business, but can also be a weak point for data. Endpoint security is the process of protecting remote access to a company's network.

Data security: Inside of networks and applications is data. Protecting company and customer information is a separate layer of security.

Identity management: Essentially, this is a process of understanding the access every individual has in an organization.

Database and infrastructure security: Everything in a network involves databases and physical equipment. Protecting these devices is equally important.

Cloud security: Many files are in digital environments or "the cloud". Protecting data in a 100% online environment presents a large amount of challenges.

Mobile security: Cell phones and tablets involve virtually every type of security challenge in and of themselves.

Disaster recovery/business continuity planning: In the event of a breach, natural disaster or other event data must be protected and business must go on. For this, you'll need a plan.End-user education: Users may be employees accessing the network or customers logging on to a company app. Educating good habits (password changes, 2-factor authentication, etc.) is an important part of cybersecurity.
The most difficult challenge in cyber security is the ever-evolving nature of security risks themselves. Traditionally, organizations and the government have focused most of their cyber security resources on perimeter security to protect only their most crucial system components and defend against known treats. Today, this approach is insufficient, as the threats advance and change more quickly than organizations can keep up with. As a result, advisory organizations promote more proactive and adaptive approaches to cyber security. Similarly, the National Institute of Standards and Technology (NIST) issued guidelines in its risk assessment framework that recommend a shift toward continuous monitoring and real-time assessments, a data-focused approach to security as opposed to the traditional perimeter-based model.

Managing Cyber Security:

The National Cyber Security Alliance, through SafeOnline.org, recommends a top-down approach to cyber security in which corporate management leads the charge in prioritizing cyber security management across all business practices. NCSA advises that companies must be prepared to "respond to the inevitable cyber incident, restore normal operations, and ensure that company assets and the company's reputation are protected." NCSA's guidelines for conducting cyber risk assessments focus on three key areas: identifying your organization's "crown jewels," or your most valuable information requiring protection; identifying the threats and risks facing that information; and outlining the damage your organization would incur should that data be lost or wrongfully exposed. Cyber risk assessments should also consider any regulations that impact the way your company collects, stores, and secures data, such as PCI-DSS, HIPAA, SOX, FISMA, and others. Following a cyber risk assessment, develop and implement a plan to mitigate cyber risk, protect the "crown jewels" outlined in your assessment, and effectively detect and respond to security incidents. This plan should encompass both the processes and technologies required to build a mature cyber security program. An ever-evolving field, cyber security best practices must evolve to accommodate the increasingly sophisticated attacks carried out by attackers. Combining sound cyber security measures with an educated and security-minded employee base provides the best defense against cyber criminals attempting to gain access to your company's sensitive data. While it may seem like a daunting task, start small and focus on your most sensitive data, scaling your efforts as your cyber program matures.

AFTERNOON SESSION			
Date:	16th June 2020	**Name:**	Soundarya NA
Course:	UDEMY	**USN:**	4AL16EC077
Topic:	MySQL	**Semester & Section:**	8th - B

Image:

```
hg$
hg$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.17 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW tables;
+----------------------------------------------------+
| Tables_in_test                                     |
+----------------------------------------------------+
```

Report:
Server code:

```
int main(int argc, char **argv)
 {
   _cust_check_startup();
   (void) thr_setconcurrency(concurrency);
   init_ssl();
   server_init();                    // 'bind' + 'listen'
   init_server_components();
   start_signal_handler();
   acl_init((THD *)0, opt_noacl);
   init_slave();
   create_shutdown_thread();
   create_maintenance_thread();
   handle_connections_sockets(0);          // !
   DBUG_PRINT("quit",("Exiting main thread"));
   exit(0);
 }
handle_connections_sockets (arg __attribute__((unused))
 {
    if (ip_sock != INVALID_SOCKET)
    {
      FD_SET(ip_sock,&clientFDs);
      DBUG_PRINT("general",("Waiting for connections."));
      while (!abort_loop)
      {
        new_sock = accept(sock, my_reinterpret_cast(struct sockaddr*)
          (&cAddr),          &length);
        thd= new THD;
        if (sock == unix_sock)
        thd->host=(char*) localhost;
```

```
      create_new_thread(thd);          // !
      }
create_new_thread(THD *thd)
 {
  pthread_mutex_lock(&LOCK_thread_count);
  pthread_create(&thd->real_id,&connection_attrib,
     handle_one_connection,                    // !
     (void*) thd));
  pthread_mutex_unlock(&LOCK_thread_count);
 }
handle_one_connection(THD *thd)
 {
  init_sql_alloc(&thd->mem_root, MEM_ROOT_BLOCK_SIZE, MEM_ROOT_PREALLOC);
  while (!net->error && net->vio != 0 && !thd->killed)
  {
    if (do_command(thd))          // !
     break;
  }
  close_connection(net);
  end_thread(thd,1);
  packet=(char*) net->read_pos;
bool do_command(THD *thd)
{
 net_new_transaction(net);
 packet_length=my_net_read(net);
 packet=(char*) net->read_pos;
 command = (enum enum_server_command) (uchar) packet[0];
 dispatch_command(command,thd, packet+1, (uint) packet_length);
// !
}
```

```
bool dispatch_command(enum enum_server_command command, THD *thd,
      char* packet, uint packet_length)
{
 switch (command) {
  case COM_INIT_DB:         ...
  case COM_REGISTER_SLAVE:   ...
  case COM_TABLE_DUMP:       ...
  case COM_CHANGE_USER:      ...
  case COM_EXECUTE:
      mysql_stmt_execute(thd,packet);
  case COM_LONG_DATA:        ...
  case COM_PREPARE:
      mysql_stmt_prepare(thd, packet, packet_length);   // !
  /* and so on for 18 other cases */
  default:
   send_error(thd, ER_UNKNOWN_COM_ERROR);
   break;
  }
 bool dispatch_command(enum enum_server_command command, THD *thd,
     char* packet, uint packet_length)
 {
 switch (command) {
  case COM_INIT_DB:         ...
  case COM_REGISTER_SLAVE:   ...
  case COM_TABLE_DUMP:       ...
  case COM_CHANGE_USER:      ...
  case COM_EXECUTE:
      mysql_stmt_execute(thd,packet);                // !
  case COM_LONG_DATA:        ...
  case COM_PREPARE:
```

```
      mysql_stmt_prepare(thd, packet, packet_length);
  /* and so on for 18 other cases */
  default:
   send_error(thd, ER_UNKNOWN_COM_ERROR);
   break;
  }
 void mysql_stmt_execute(THD *thd, char *packet)
 {
  if (!(stmt=find_prepared_statement(thd, stmt_id, "execute")))
  {
   send_error(thd);
   DBUG_VOID_RETURN;
  }
  init_stmt_execute(stmt);
  mysql_execute_command(thd);          // !
 }
 void mysql_execute_command(THD *thd)
     switch (lex->sql_command) {
     case SQLCOM_SELECT: ...
     case SQLCOM_SHOW_ERRORS: ...
     case SQLCOM_CREATE_TABLE: ...
     case SQLCOM_UPDATE: ...
     case SQLCOM_INSERT: ...                // !
     case SQLCOM_DELETE: ...
     case SQLCOM_DROP_TABLE: ...
     }
case SQLCOM_INSERT:
{
 my_bool update=(lex->value_list.elements ? UPDATE_ACL : 0);
 ulong privilege= (lex->duplicates == DUP_REPLACE ?
```

```
                INSERT_ACL | DELETE_ACL : INSERT_ACL | update);
  if (check_access(thd,privilege,tables->db,&tables->grant.privilege))
    goto error;
  if (grant_option && check_grant(thd,privilege,tables))
    goto error;
  if (select_lex->item_list.elements != lex->value_list.elements)
  {
    send_error(thd,ER_WRONG_VALUE_COUNT);
    DBUG_VOID_RETURN;
  }
  res = mysql_insert(thd,tables,lex->field_list,lex->many_values,
                     select_lex->item_list, lex->value_list,
                     (update ? DUP_UPDATE : lex->duplicates));
// !
  if (thd->net.report_error)
    res= -1;
  break;
}
int ha_myisam::write_row(byte * buf)
{
  statistic_increment(ha_write_count,&LOCK_status);
   /* If we have a timestamp column, update it to the current time */
   if (table->time_stamp)
    update_timestamp(buf+table->time_stamp-1);
   /*
  If we have an auto_increment column and we are writing a changed row
    or a new row, then update the auto_increment value in the record.
  */
  if (table->next_number_field && buf == table->record[0])
    update_auto_increment();
```

```
  return mi_write(file,buf);     // !
}
```