[April 2018 Free Lead2pass (ISC)2 CISSP Exam Questions Download 2873q

Free Share CISSP PDF Dumps With Lead2pass Updated Exam Questions: https://www.lead2pass.com/cissp.html QUESTION 21 What is called the percentage of valid subjects that are falsely rejected by a Biometric Authentication system? A. False Rejection Rate (FRR) or Type I ErrorB. False Acceptance Rate (FAR) or Type II ErrorC. Crossover Error Rate (CER)D. True Rejection Rate (TRR) or Type III ErrorAnswer: AExplanation:The percentage of valid subjects that are falsely rejected is called the False Rejection Rate (FRR) or Type I Error. QUESTION 22What is called the percentage of invalid subjects that are falsely accepted by a Biometric authentication system? A. False Rejection Rate (FRR) or Type I ErrorB. False Acceptance Rate (FAR) or Type II ErrorC. Crossover Error Rate (CER)D. True Acceptance Rate (TAR) or Type III Error Answer: BExplanation:The percentage of invalid subjects that are falsely accepted is called the False Acceptance Rate (FAR) or Type II Error. QUESTION 23What is called the percentage at which the False Rejection Rate equals the False Acceptance Rate? A. False Rejection Rate (FRR) or Type I Error B. False Acceptance Rate (FAR) or Type II ErrorC. Crossover Error Rate (CER)D. Failure to enroll rate (FTE or FER) Answer: CExplanation:The percentage at which the False Rejection Rate equals the False Acceptance Rate is called the Crossover Error Rate (CER). Another name for the CER is the Equal Error Rate (EER), any of the two terms could be used. Equal error rate or crossover error rate (EER or CER) It is the rate at which both accept and reject errors are equal. The EER is a quick way to compare the accuracy of devices with different ROC curves. In general, the device with the lowest EER is most accurate. The other choices were all wrong answers: The following are used as performance metrics for biometric systems:False accept rate or false match rate (FAR or FMR): the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs which are incorrectly accepted. This is when an impostor would be accepted by the system false reject rate or false non-match rate (FRR or FNMR): the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percent of valid inputs which are incorrectly rejected. This is when a valid company employee would be rejected by the system Failure to enroll rate (FTE or FER): the rate at which attempts to create a template from an input is unsuccessful. This is most commonly caused by low quality inputs. QUESTION 24Considerations of privacy, invasiveness, and psychological and physical comfort when using the system are important elements for which of the following? A. Accountability of biometrics systemsB. Acceptability of biometrics systemsC. Availability of biometrics systemsD. Adaptability of biometrics systems Answer: BExplanation:Acceptability refers to considerations of privacy, invasiveness, and psychological and physical comfort when using the system. QUESTION 25Which of the following offers advantages such as the ability to use stronger passwords, easier password administration, one set of credential, and faster resource access? A. Smart cardsB. Single Sign-On (SSO)C. Symmetric CiphersD. Public Key Infrastructure (PKI) Answer: B Explanation:The advantages of SSO include having the ability to use stronger passwords, easier administration as far as changing or deleting the passwords, minimize the risks of orphan accounts, and requiring less time to access resources. QUESTION 26Which of the following describes the major disadvantage of many Single Sign-On (SSO) implementations? A. Once an individual obtains access to the system through the initial log-on, they have access to all resources within the environment that the account has access to.B. The initial logon process is cumbersome to discourage potential intruders.C. Once a user obtains access to the system through the initial log-on, they only need to logon to some applications.D. Once a user obtains access to the system through the initial log-on, he has to logout from all other systems Answer: AExplanation:Single Sign-On is a distrubuted Access Control methodology where an individual only has to authenticate once and would have access to all primary and secondary network domains. The individual would not be required to re-authenticate when they needed additional resources. The security issue that this creates is if a fraudster is able to compromise those credential they too would have access to all the resources that account has access to. All the other answers are incorrect as they are distractors. QUESTION 27Which of the following is implemented through scripts or smart agents that replays the users multiple log-ins against authentication servers to verify a user's identity which permit access to system services? A. Single Sign-OnB. Dynamic Sign-OnC. Smart cardsD. Kerberos Answer: AExplanation:SSO can be implemented by using scripts that replay the users multiple log-ins against authentication servers to verify a user's identity and to permit access to system services. Single Sign on was the best answer in this case because it would include Kerberos. When you have two good answers within the 4 choices presented you must select the BEST one. The high level choice is always the best. When one choice would include the other one that would be the best as well. QUESTION 28Which of the following is NOT true of the Kerberos protocol? A. Only a single login is required per session.B. The initial authentication steps are done using public key algorithm.C. The KDC is aware of all systems in the network and is trusted by all of themD. It performs mutual authentication Answer: BExplanation:Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. It has the following characteristics: - It is secure: it never sends a password unless it

is encrypted.- Only a single login is required per session. Credentials defined at login are then passed between resources without the need for additional logins.- The concept depends on a trusted third party ?a Key Distribution Center (KDC). The KDC is aware of all systems in the network and is trusted by all of them.- It performs mutual authentication, where a client proves its identity to a server and a server proves its identity to the client. Kerberos introduces the concept of a Ticket-Granting Server/Service (TGS). A client that wishes to use a service has to receive a ticket from the TGS a ticket is a time-limited cryptographic message-giving it access to the server. Kerberos also requires an Authentication Server (AS) to verify clients. The two servers combined make up a KDC. Within the Windows environment, Active Directory performs the functions of the KDC. The following figure shows the sequence of events required for a client to gain access to a service using Kerberos authentication. Each step is shown with the Kerberos message associated with it, as defined in RFC 4120 "The Kerberos Network Authorization Service (V5)". Kerberos Authentication Step by Step - Step 1: The user logs on to the workstation and requests service on the host. The workstation sends a message to the Authorization Server requesting a ticket granting ticket (TGT). - Step 2: The Authorization Server verifies the user's access rights in the user database and creates a TGT and session key. The Authorization Sever encrypts the results using a key derived from the user's password and sends a message back to the user workstation. The workstation prompts the user for a password and uses the password to decrypt the incoming message. When decryption succeeds, the user will be able to use the TGT to request a service ticket. - Step 3: When the user wants access to a service, the workstation client application sends a request to the Ticket Granting Service containing the client name, realm name and a timestamp. The user proves his identity by sending an authenticator encrypted with the session key received in Step 2 - Step 4: The TGS decrypts the ticket and authenticator, verifies the request, and creates a ticket for the requested server. The ticket contains the client name and optionally the client IP address. It also contains the realm name and ticket lifespan. The TGS returns the ticket to the user workstation. The returned message contains two copies of a server session key ?one encrypted with the client password, and one encrypted by the service password. - Step 5: The client application now sends a service request to the server containing the ticket received in Step 4 and an authenticator. The service authenticates the request by decrypting the session key. The server verifies that the ticket and authenticator match, and then grants access to the service. This step as described does not include the authorization performed by the Intel AMT device, as described later. - Step 6: If mutual authentication is required, then the server will reply with a server authentication message. The Kerberos server knows "secrets" (encrypted passwords) for all clients and servers under its control, or it is in contact with other secure servers that have this information. These "secrets" are used to encrypt all of the messages shown in the figure above. To prevent "replay attacks," Kerberos uses timestamps as part of its protocol definition. For timestamps to work properly, the clocks of the client and the server need to be in synch as much as possible. In other words, both computers need to be set to the same time and date. Since the clocks of two computers are often out of synch, administrators can establish a policy to establish the maximum acceptable difference to Kerberos between a client's clock and server's clock. If the difference between a client's clock and the server's clock is less than the maximum time difference specified in this policy, any timestamp used in a session between the two computers will be considered authentic. The maximum difference is usually set to five minutes. Note that if a client application wishes to use a service that is "Kerberized" (the service is configured to perform Kerberos authentication), the client must also be Kerberized so that it expects to support the necessary message responses.For more information about Kerberos, see http://web.mit.edu/kerberos/www/. QUESTION 29Which of the following is used to create and modify the structure of your tables and other objects in the database? A. SQL Data Definition Language (DDL)B. SQL Data Manipulation Language (DML)C. SQL Data Relational Language (DRL)D. SQL Data Identification Language (DIL) Answer: AExplanation:Explanation:The SQL Data Definition Language (DDL) is used to create, modify, and delete views and relations (tables).Data Definition Language The Data Definition Language (DDL) is used to create and destroy databases and database objects. These commands will primarily be used by database administrators during the setup and removal phases of a database project. Let's take a look at the structure and usage of four basic DDL commands: CREATE Installing a database management system (DBMS) on a computer allows you to create and manage many independent databases. For example, you may want to maintain a database of customer contacts for your sales department and a personnel database for your HR department. The CREATE command can be used to establish each of these databases on your platform. For example, the command: CREATE DATABASE employees creates an empty database named "employees" on your DBMS. After creating the database, your next step is to create tables that will contain data. (If this doesn't make sense, you might want to read the article Microsoft Access Fundamentals for an overview of tables and databases.) Another variant of the CREATE command can be used for this purpose. The command: CREATE TABLE personal_info (first_name char(20) not null, last_name char(20) not null, employee_id int not null) establishes a table titled "personal_info" in the current database. In our example, the table contains three attributes: first_name, last_name and employee_id. Don't worry about the other information included in the command -- we'll cover that in a future article. USEThe USE command allows you to specify the database you wish to work with within your DBMS. For example, if we're

currently working in the sales database and want to issue some commands that will affect the employees database, we would preface them with the following SQL command: USE employees It's important to always be conscious of the database you are working in before issuing SQL commands that manipulate data. ALTEROnce you've created a table within a database, you may wish to modify the definition of it. The ALTER command allows you to make changes to the structure of a table without deleting and recreating it. Take a look at the following command: ALTER TABLE personal_infoADD salary money null This example adds a new attribute to the personal_info table -- an employee's salary. The "money" argument specifies that an employee's salary will be stored using a dollars and cents format. Finally, the "null" keyword tells the database that it's OK for this field to contain no value for any given employee. DROPThe final command of the Data Definition Language, DROP, allows us to remove entire database objects from our DBMS. For example, if we want to permanently remove the personal_info table that we created, we'd use the following command: DROP TABLE personal_infoSimilarly, the command below would be used to remove the entire employees database: DROP DATABASE employees Use this command with care! Remember that the DROP command removes entire data structures from your database. If you want to remove individual records, use the DELETE command of the Data Manipulation Language. That's the Data Definition Language in a nutshell.Data Manipulation Language The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands: The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands: INSERTThe INSERT command in SQL is used to add records to an existing table. Returning to the personal_info example from the previous section, let's imagine that our HR department needs to add a new employee to their database. They could use a command similar to the one shown below: INSERT INTO personal_info values('bart','simpson',12345,$45000) Note that there are four values specified for the record. These correspond to the table attributes in the order they were defined: first_name, last_name, employee_id, and salary. SELECTThe SELECT command is the most commonly used command in SQL. It allows database users to retrieve the specific information they desire from an operational database. Let's take a look at a few examples, again using the personal_info table from our employees database. The command shown below retrieves all of the information contained within the personal_info table. Note that the asterisk is used as a wildcard in SQL. This literally means "Select everything from the personal_info table." SELECT *FROM personal_info Alternatively, users may want to limit the attributes that are retrieved from the database. For example, the Human Resources department may require a list of the last names of all employees in the company. The following SQL command would retrieve only that information: SELECT last_nameFROM personal_info Finally, the WHERE clause can be used to limit the records that are retrieved to those that meet specified criteria. The CEO might be interested in reviewing the personnel records of all highly paid employees. The following command retrieves all of the data contained within personal_info for records that have a salary value greater than $50,000: SELECT *FROM personal_infoWHERE salary > $50000 UPDATEThe UPDATE command can be used to modify information contained within a table, either in bulk or individually. Each year, our company gives all employees a 3% cost-of-living increase in their salary. The following SQL command could be used to quickly apply this to all of the employees stored in the database: UPDATE personal_infoSET salary = salary * 103 On the other hand, our new employee Bart Simpson has demonstrated performance above and beyond the call of duty. Management wishes to recognize his stellar accomplishments with a $5,000 raise. The WHERE clause could be used to single out Bart for this raise: UPDATE personal_infoSET salary = salary + $5000WHERE employee_id = 12345 DELETE Finally, let's take a look at the DELETE command. You'll find that the syntax of this command is similar to that of the other DML commands. Unfortunately, our latest corporate earnings report didn't quite meet expectations and poor Bart has been laid off. The DELETE command with a WHERE clause can be used to remove his record from the personal_info table: DELETE FROM personal_info WHERE employee_id = 12345JOIN Statements Now that you've learned the basics of SQL, it's time to move on to one of the most powerful concepts the language has to offer ?the JOIN statement. Quite simply, these statements allow you to combine data in multiple tables to quickly and efficiently process large quantities of data. These statements are where the true power of a database resides. We'll first explore the use of a basic JOIN operation to combine data from two tables. In future installments, we'll explore the use of outer and inner joins to achieve added power. We'll continue with our example using the PERSONAL_INFO table, but first we'll need to add an additional table to the mix. Let's assume we have a table called DISCIPLINARY_ACTION that was created with the following statement: CREATE TABLE disciplinary_action (action_id int not null, employee_id int not null, comments char(500)) This table contains the results of disciplinary actions on company employees. You'll notice that it doesn't contain any information about the employee other than the employee number. It's then easy to imagine many scenarios where we might want to combine information from the DISCIPLINARY_ACTION and PERSONAL_INFO tables. Assume we've been tasked with creating a report that lists the disciplinary actions taken against all employees with a salary greater than $40,000 The use of a

JOIN operation in this case is quite straightforward. We can retrieve this information using the following command: SELECT personal_info.first_name, personal_info.last_name, disciplinary_action.comments FROM personal_info, disciplinary_action WHERE personal_info.employee_id = disciplinary_action.employee_id AND personal_info.salary > 40000 As you can see, we simply specified the two tables that we wished to join in the FROM clause and then included a statement in the WHERE clause to limit the results to records that had matching employee IDs and met our criteria of a salary greater than $40,000 Another term you must be familiar with as a security mechanism in Databases is: VIEW What is a view? In database theory, a view is a virtual or logical table composed of the result set of a query. Unlike ordinary tables (base tables) in a relational database, a view is not part of the physical schema: it is a dynamic, virtual table computed or collated from data in the database. Changing the data in a table alters the data shown in the view. The result of a view is stored in a permanent table whereas the result of a query is displayed in a temporary table. Views can provide advantages over tables; They can subset the data contained in a tableThey can join and simplify multiple tables into a single virtual table Views can act as aggregated tables, where aggregated data (sum, average etc.) are calculated and presented as part of the dataViews can hide the complexity of data, for example a view could appear as Sales2000 or Sales2001, transparently partitioning the actual underlying table Views take very little space to store; only the definition is stored, not a copy of all the data they presentDepending on the SQL engine used, views can provide extra security. Limit the exposure to which a table or tables are exposed to outer world Just like functions (in programming) provide abstraction, views can be used to create abstraction. Also, just like functions, views can be nested, thus one view can aggregate data from other views. Without the use of views it would be much harder to normalise databases above second normal form. Views can make it easier to create lossless join decomposition. Rows available through a view are not sorted. A view is a relational table, and the relational model states that a table is a set of rows. Since sets are not sorted - per definition - the rows in a view are not ordered either. Therefore, an ORDER BY clause in the view definition is meaningless and the SQL standard (SQL:2003) does not allow this for the subselect in a CREATE VIEW statement. QUESTION 30Which of the following is used to monitor network traffic or to monitor host audit logs in real time to determine violations of system security policy that have taken place? A. Intrusion Detection SystemB. Compliance Validation SystemC. Intrusion Management System (IMS)D. Compliance Monitoring System Answer: AExplanation:Explanation:An Intrusion Detection System (IDS) is a system that is used to monitor network traffic or to monitor host audit logs in order to determine if any violations of an organization's system security policy have taken place. **CISSP dumps full version (PDF&VCE):** https://www.lead2pass.com/cissp.html Large amount of free **CISSP** exam questions on Google Drive: https://drive.google.com/open?id=1393N8RayZN4QJ8sxg6_3cIRxwNv8QGTq