**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **16-06-2020** | **Name:** | **Anand kumar k** |
| **Course:** | **Cyber security** | **USN:** | **4al16ec002** |
| **Topic:** | **Blockchain in cyber security**  **Career and industry landscape** | **Semester & Section:** | **8thsem ‘A’ sec** |
| **Github Repository:** | **Anand-courses** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**      The high level of dependency on the internet and technology today has resulted in new revenue streams and business models for organizations, but with this arises new gaps and opportunities for hackers to exploit.  Cybercriminals have become increasingly complex and are attempting to steal valuable data like financial data, health records, personal identifiable information (PII) and intellectual property, and are resorting to highly profitable strategies like disrupting the overall operations of a business via DDoS attacks, or monetizing data access via the utilization of advanced ransomware techniques.  So, will blockchain technology be a cybersecurity help or hindrance? A blockchain is basically a decentralized, digitized, public ledger of all cryptocurrency transactions and uses what is known as the Distributed Ledger Technology. This could potentially help enhance cyber-defense as the platform can prevent fraudulent activities via consensus mechanisms, and detect data tampering depending on its underlying characteristics of operational resilience, data encryption, auditability, transparency and immutability  Blockchain resolves the ‘lack of trust’ problem between counterparties at a very basic level. Blockchain is a distributed database used in both private and public applications rather than a centralized structure where all the information is stored in few very large databases. The data pertaining to each batch of valid transactions is stored within its own block; every block is connected to the block which is situated in the position before it and grows continuously as new blocks of information are appended.  Owing to their distributed nature, blockchains provide no ‘hackable’ entrance or a central point of failure and, thereby, provide more security when compared with various present database-driven transactional structures.  **Eliminating Human Factor from Authentication**  Businesses are able to authenticate devices and users without the need for a password with the help of blockchain technology. This eliminates human intervention from the process of authentication, thereby avoiding it from becoming a potential attack vector.  The utilization of a centralized architecture and simple logins are the big weakness of conventional systems. No matter how much money an organization invests in security, all these efforts go in vain if the employees and customers use passwords that are easy to steal or crack. Blockchain offers strong authentication and resolving single point of attack at the same time.  With the help of blockchain, a security system used in an organization can leverage a distributed public key infrastructure for authenticating devices and users. This security system provides each device with a specific SSL certificate instead of a password. Management of certificate data is carried out on the blockchain and this makes it virtually impossible for attackers to utilize fake certificates.  **Decentralized Storage**  Blockchain users can maintain their data on their computer in their network. Because of this, they can make sure that the chain won’t collapse. For instance, if someone who is not the owner of a component of data (such as an attacker) attempts to tamper with a block, the entire system examines each and every data block to locate the one that differs from the rest. If this type of block is located by the system, it simply excludes the block from the chain, recognizing it as false.  Blockchain is designed in a way that the storage location or central authority doesn’t exist. On the network, every user has a role to play in storing some or all the blockchain. Everyone in the blockchain network is responsible for verifying the data that is shared and/or maintained to ensure existing data can’t be removed and false data can’t be added.  **Traceability**  Every transaction added to a private or public blockchain is timestamped and signed digitally. This means that companies can trace back to a particular time period for every transaction and locate the corresponding party on the blockchain through their public address.  This feature relates to non-repudiation: the assurance that someone can’t verify their signature’s authenticity on a file, or the authorship a transaction that they originated. This blockchain’s functionality increases the system’s reliability as every transaction is associated cryptographically to a user.  Any new transaction that gets appended to a blockchain results in the transformation of ledger’s global state. This implies that with the system’s every new iteration, the previous state will be stored resulting in a history log that is completely traceable.  The audit capability of blockchain offers companies with a level of security and transparency over every iteration. From the perspective of cybersecurity, this offers entities with an additional level of reassurance that the data hasn’t been tampered with and is authentic.  **DDoS**  Blockchain transactions can be denied easily if the participating units are impeded from sending transactions. For example, a DDoS attack on a set of entities or an entity can cripple the entire attendant infrastructure and the blockchain organization. These kind of attacks can introduce integrity risks to blockchain.  At present, the difficulty in impeding the attacks due to DDoS comes from the existing Domain Name System. The implementation of blockchain technology would completely decentralize the DNS, distributing the contents to more number of nodes thereby making it almost impossible for cyber-attackers to hack. A system can make sure that it is invulnerable to hackers by using blockchains to safeguard data unless every single node is wiped clean at the same time.  Blockchain technology is here to stay and it will help us protect as companies, individuals, and governments. The innovative blockchain utilization is already becoming a component of other fields beyond cryptocurrencies and is mainly useful to [enhance cybersecurity](https://mindmajix.com/cyber-security-training).  Though few of the underlying capabilities of blockchains offer data availability, integrity, and confidentiality, like various other systems, cybersecurity standards and controls must be followed by companies within their technical infrastructure with the help of blockchains to protect them from outside attacks. |
|  |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **16-06-2020** | **Name:** | **Anand kumar k** | |
| **Course:** |  | **USN:** | **4al16ec002** | |
| **Topic:** | **mysql** | **Semester & Section:** | **8thsem ‘A’ sec** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session**      The procedure for starting a single MySQL server manually from the command line is described in [Section 2.3.4.6, “Starting MySQL from the Windows Command Line”](https://dev.mysql.com/doc/refman/8.0/en/windows-start-command-line.html). To start multiple servers this way, you can specify the appropriate options on the command line or in an option file. It is more convenient to place the options in an option file, but it is necessary to make sure that each server gets its own set of options. To do this, create an option file for each server and tell the server the file name with a [--defaults-file](https://dev.mysql.com/doc/refman/8.0/en/option-file-options.html#option_general_defaults-file) option when you run it.  Suppose that you want to run one instance of **[mysqld](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html" \o "4.3.1 mysqld — The MySQL Server)** on port 3307 with a data directory of C:\mydata1, and another instance on port 3308 with a data directory of C:\mydata2. Use this procedure:   1. Make sure that each data directory exists, including its own copy of the mysql database that contains the grant tables. 2. Create two option files. For example, create one file named C:\my-opts1.cnf that looks like this: 3. [mysqld] 4. datadir = C:/mydata1   port = 3307  Create a second file named C:\my-opts2.cnf that looks like this:  [mysqld]  datadir = C:/mydata2  port = 3308   1. Use the [--defaults-file](https://dev.mysql.com/doc/refman/8.0/en/option-file-options.html#option_general_defaults-file) option to start each server with its own option file: 2. C:\> C:\mysql\bin\mysqld --defaults-file=C:\my-opts1.cnf   C:\> C:\mysql\bin\mysqld --defaults-file=C:\my-opts2.cnf  Each server starts in the foreground (no new prompt appears until the server exits later), so you will need to issue those two commands in separate console windows.  To shut down the servers, connect to each using the appropriate port number:  C:\> C:\mysql\bin\mysqladmin --port=3307 --host=127.0.0.1 --user=root --password shutdown  C:\> C:\mysql\bin\mysqladmin --port=3308 --host=127.0.0.1 --user=root --password shutdown  Servers configured as just described permit clients to connect over TCP/IP. If your version of Windows supports named pipes and you also want to permit named-pipe connections, specify options that enable the named pipe and specify its name. Each server that supports named-pipe connections must use a unique pipe name. For example, the C:\my-opts1.cnf file might be written like this:  [mysqld]  datadir = C:/mydata1  port = 3307  enable-named-pipe  socket = mypipe1  Modify C:\my-opts2.cnf similarly for use by the second server. Then start the servers as described previously.  On Windows, a MySQL server can run as a Windows service. The procedures for installing, controlling, and removing a single MySQL service are described in [Section 2.3.4.8, “Starting MySQL as a Windows Service”](https://dev.mysql.com/doc/refman/8.0/en/windows-start-service.html).  To set up multiple MySQL services, you must make sure that each instance uses a different service name in addition to the other parameters that must be unique per instance.  For the following instructions, suppose that you want to run the **[mysqld](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html" \o "4.3.1 mysqld — The MySQL Server)** server from two different versions of MySQL that are installed at C:\mysql-5.5.9 and C:\mysql-8.0.22, respectively. (This might be the case if you are running 5.5.9 as your production server, but also want to conduct tests using 8.0.22.)  To install MySQL as a Windows service, use the --install or --install-manual option. For information about these options, see [Section 2.3.4.8, “Starting MySQL as a Windows Service”](https://dev.mysql.com/doc/refman/8.0/en/windows-start-service.html).  Based on the preceding information, you have several ways to set up multiple services. The following instructions describe some examples. Before trying any of them, shut down and remove any existing MySQL services.   * ***Approach 1:*** Specify the options for all services in one of the standard option files. To do this, use a different service name for each server. Suppose that you want to run the 5.5.9 **[mysqld](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html" \o "4.3.1 mysqld — The MySQL Server)** using the service name of mysqld1 and the 8.0.22 **[mysqld](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html" \o "4.3.1 mysqld — The MySQL Server)** using the service name mysqld2. In this case, you can use the [mysqld1] group for 5.5.9 and the [mysqld2] group for 8.0.22. For example, you can set up C:\my.cnf like this: * # options for mysqld1 service * [mysqld1] * basedir = C:/mysql-5.5.9 * port = 3307 * enable-named-pipe * socket = mypipe1 * # options for mysqld2 service * [mysqld2] * basedir = C:/mysql-8.0.22 * port = 3308 * enable-named-pipe   socket = mypipe2  Install the services as follows, using the full server path names to ensure that Windows registers the correct executable program for each service:  C:\> C:\mysql-5.5.9\bin\mysqld --install mysqld1  C:\> C:\mysql-8.0.22\bin\mysqld --install mysqld2  To start the services, use the services manager, or **NET START** or **SC START** with the appropriate service names:  C:\> SC START mysqld1  C:\> SC START mysqld1 | | | |
|  | | | |