

Домінський В.О. ІІІ-93
Системи безпеки програм і даних.
КР№1-2. 9311

1. Конфіденційність. Цілісність. Доступність (2 бали).

- Напишіть есе (200-500 знаків), щодо основних проблеми безпеки програм і даних, опишіть, як розумієте авторизацію та аутентифікацію, можна навести приклад:

Проблем, котрі пов'язані з безпекою програм і даних, з кожним днем стає все більше й більше, оскільки ІТ індустрія дуже швидко росте. Ось кілька з доволі розповсюджених загроз:

- Фішинг та соціальна інженерія
 - Кіберзлочинці обманом змушують інсайдерів розкрити свої облікові дані або натиснути на заражені посилання чи вкладення, видаючи себе за друзів чи інші довірені джерела або пропонуючи несподівані призи від популярних брендів. Потрапивши всередину, вони можуть легко скомпрометувати мережеву безпеку
- Обмін даними за межами компанії
 - Співробітники діляться конфіденційними даними компанії, такими як інтелектуальна власність або конфіденційна інформація, захищена законами про захист даних, наприклад, персональна інформація або дані про стан здоров'я, публічно або з третіми особами за межами компанії.
- Тіньове ІТ
 - Використання несанкціонованого стороннього програмного забезпечення, додатків або інтернет-сервісів на робочому місці часто важко відстежити ІТ-відділу, звідси і походить термін "тіньове ІТ"

Відмінність між автентифікацією та авторизацією можна пояснити тим, що автентифікація використовує паролі, біометричні дані або інші сутності для підтвердження особи користувача, в той час як авторизація - підтверджує доступ користувача. Таким чином, коли ми говоримо про процес управління доступом, то на першому місці стоїть етап автентифікації, а потім авторизації.

2. За допомогою сайту jwt.io створить JWT токен з наступними клеймами (2 бали):

- name - ім'я та прізвище;
- sub (subject) – email;
- iat (issued at time) – час, коли токен було створено;
- exp (expiration time) – час, коли токен стане не валідним.

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmFsZW50eW4gRG9taW5za3lpIiwic3ViIjoibmFsZW50eW4yMjAyQGdtYWlsLmNvbSIsIm1hdCI6MTY3MjY3MTU0NSwiZXhwIjojNjc5Njc5NjA1fQ.SdF306ujGV-zxOnqRPD8z3eZvicPjpPT8ex3pAIMBkQ

<pre>eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmFsZW50eW4gRG9taW5za3lpIiwic3ViIjoibmFsZW50eW4yMjAyQGdtYWlsLmNvbSIsIm1hdCI6MTY3MjY3MTU0NSwiZXhwIjojNjc5Njc5NjA1fQ.SdF306ujGV-zxOnqRPD8z3eZvicPjpPT8ex3pAIMBkQ</pre>	<table><tr><td>HEADER: ALGORITHM & TOKEN TYPE</td></tr><tr><td><pre>{ "alg": "HS256", "typ": "JWT" }</pre></td></tr><tr><td>PAYLOAD: DATA</td></tr><tr><td><pre>{ "name": "Valentyn Dominskyi", "sub": "Valentyn2202@gmail.com", "iat": 1672671545, "exp": 1672671605 }</pre></td></tr></table>	HEADER: ALGORITHM & TOKEN TYPE	<pre>{ "alg": "HS256", "typ": "JWT" }</pre>	PAYLOAD: DATA	<pre>{ "name": "Valentyn Dominskyi", "sub": "Valentyn2202@gmail.com", "iat": 1672671545, "exp": 1672671605 }</pre>
HEADER: ALGORITHM & TOKEN TYPE					
<pre>{ "alg": "HS256", "typ": "JWT" }</pre>					
PAYLOAD: DATA					
<pre>{ "name": "Valentyn Dominskyi", "sub": "Valentyn2202@gmail.com", "iat": 1672671545, "exp": 1672671605 }</pre>					

Час життя – 1 хвилина

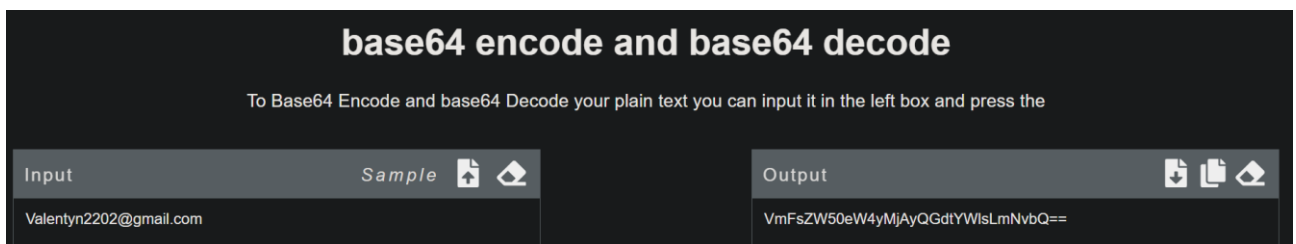
3. Назвіть приклади алгоритмів та надайте пояснення, для чого вони можуть використовуватись (6 балів):

- Вільне кодування\декодування інформації. Закодуйте повідомлення, яке містить наступний текст: емейл-адреса.

Base64 - це група схожих binary-to-text encoding схем, які представляють двійкові дані в ASCII-форматі методом переведення в radix-64 подання

Кодування Base64 широко використовується у випадках, коли потрібно перекодувати двійкові дані для передавання каналом, пристосованим для передавання текстових даних. Це робиться з метою захисту двійкових даних від будь-яких можливих пошкоджень під час передання.

Base64 широко використовується в багатьох додатках, включно з електронною поштою (MIME), під час збереження великих обсягів даних в XML, або ж для передачі великих бітових структур



base64 encode and base64 decode	
To Base64 Encode and base64 Decode your plain text you can input it in the left box and press the	
Input Valentyn2202@gmail.com	Output VmFsZW50eW4yMjAyQGdtYWlsLmNvbQ==

- Повідомлення для кодування - Valentyn2202@gmail.com
- Закодоване повідомлення - VmFsZW50eW4yMjAyQGdtYWlsLmNvbQ==
- Симетричні алгоритми шифрування Зашифруйте і розшифруйте повідомлення, яке містить цифри 9311

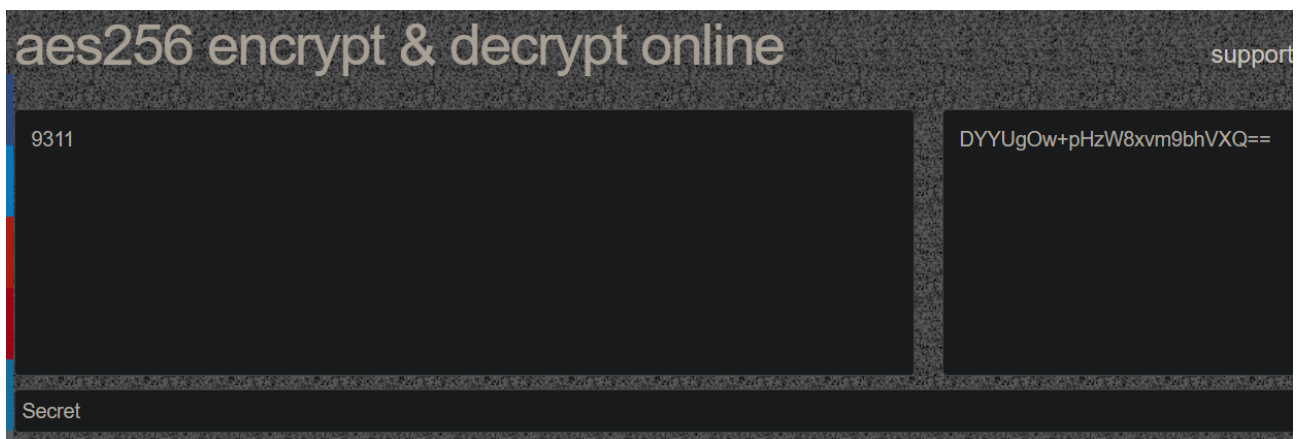
AES розшифровується як Advanced Encryption Standard, що є стандартом, який використовується в усьому світі для шифрування даних.

256 означає розмір ключа - чим більший розмір, тим більше можливих ключів. Щоб зрозуміти масштаби зусиль, необхідних для перебору всіх можливих комбінацій ключів, 256-бітове шифрування пропонує більше

комбінацій, ніж зірок у Всесвіті (септильйон або 1024 зірки), і для його зламу знадобилося б більше років роботи мільярдів комп'ютерів, ніж вік Всесвіту (13,8 мільярда років).

Використовуються в:

- Wi-Fi
- Мобільні додатки
- Підтримка нативних процесорів
- Бібліотеки на багатьох мовах розробки програмного забезпечення
- Реалізації VPN
- Компоненти операційної системи, такі як файлові системи



- Повідомлення для закодування - 9311
- Закодоване повідомлення - DYYUgOw+pHzW8xvm9bhVXQ==
- Secret - Secret
- Асиметричні алгоритми шифрування. Згенеруйте відкритий і секретні ключі. Зашифруйте і розшифруйте повідомлення, яке містить цифри 9311

Алгоритм RSA є асиметричним алгоритмом криптографії. Асиметричний насправді означає, що він працює на двох різних ключах, тобто на відкритому і закритому ключах.

Як випливає з назви, відкритий ключ надається кожному, а закритий ключ зберігається в таємниці.

Шифрування RSA часто використовується в поєднанні з іншими схемами шифрування або для цифрових підписів, які можуть довести автентичність і цілісність повідомлення. Зазвичай воно не використовується для шифрування цілих повідомлень або файлів, оскільки є менш ефективним і більш ресурсоємним, ніж шифрування з симетричним ключем.

RSA Encryption

Enter Plain Text to Encrypt

9311

Enter Public/Private key

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCV6vKsfMbT8q+9DeiPx5C2sVq+uSb6GH2eIdAs8ZmxnzLIpmvmekxIL9oqNjjG5tsjIEMMu06VVWdZDc84Ma9m99DlmbGhwc28RPdxqUHFYOFNH58OD3iF8Fdu5qkq6IUaKgbhAl6Oqyg26v7C4yKmUG04UI9AAh022NIKg5+gdQIDAQAB

RSA Key Type: ☒ Public key ☐ Private Key

Select Cipher Type

RSA/ECB/PKCS1Padding

Encrypt

Encrypted Output (Base64):

VZ8MBwS6/ks53CkDpDs7qlfZWZw4rNRjE+FDDeOSVLcQqNIagKzUnGfGmcntV5k8UEDjZjkua7BDU+jMsPi50/8wuzP6M/WTCsMvoZXAL5lajejvf8rISpVRsnj9qFt2XE1kC0lzU0dSM+gGPRWrY0hiJeBE2J5qjxEPdHspNizI=

RSA Decryption

Enter Encrypted Text to Decrypt (Base64)

VZ8MBwS6/ks53CkDpDs7qlfZWZw4rNRjE+FDDeOSVLcQqNIagKzUnGfGmcntV5k8UEDjZjkua7BDU+jMsPi50/8wuzP6M/WTCsMvoZXAL5lajejvf8rISpVRsnj9qFt2XE1kC0lzU0dSM+gGPRWrY0hiJeBE2J5qjxEPdHspNizI=

Enter Public/Private key

MIICdQIBADANBgkqhkiG9w0BAQEFAASCALBgwggJbAgEAAoGBAJXq8qx8xtPyr70N6I/HkLaxWr65JvoYfZ4h0CzymbGfMuWma+Z6TGUV2io2OMbm2yPUQwy6jpVVZ1kNzzgxr2b30OWZsaHBzbxE93GpQd9g4U0fnw4PeIXwV27mqSrqVRoqBuEAjo6rKDbq/sLjIqZQbThTCHThY0ggDn6B1AqMBA4AECqYAgAKWJGKH

RSA Key Type: ☐ Public key ☒ Private Key

Select Cipher Type

RSA/ECB/PKCS1Padding

Decrypt

Decrypted Output:

9311

- Повідомлення для за кодування - 9311
- За кодоване повідомлення -
VZ8MBwS6/ks53CkDpDs7qlfZWZw4rNRjE+FDDeOSVLcQqNIagKzUnGfGmcntV5k8UEDjZjkua7BDU+jMsPi50/8wuzP6M/WTCsMvoZXAL5lajejvf8rISpVRsnj9qFt2XE1kC0lzU0dSM+gGPRWrY0hiJeBE2J5qjxEPdHspNizI=
- Public key -
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCV6vKsfMbT8q+

9DeiPx5C2sVq+uSb6GH2eIdAs8ZmxnzLlpmvmekxlL9oqNjjG5tsj1EMMu06
VVWdZDc84Ma9m99DlmbGhwc28RPdxqUHfYOFNH58OD3iF8Fdu5qkq6l
UaKgbhAI6Oqyg26v7C4yKmUG04U19AAh022NIKg5+gdQIDAQAB

- Private key -

MIICdQIBADANBgkqhkiG9w0BAQEFAASCAl8wggJbAgEAAoGBAJXq8q
x8xtPyr70N6I/HkLaxWr65JvoYfZ4h0CzxmbGfMuWma+Z6TGUv2io2OMb
m2yPUQwy6jpVVZ1kNzzgxr2b30OWZsaHBzbxE93GpQd9g4U0fnw4PeIXw
V27mqSrqVRoqBuEAjo6rKDbq/sLjIqZQbThTX0ACHTbY0gqDn6B1AgMB
AAECgYAgAKWJGKHBAN9u9hcdCjbkKPv8Fr4xSSUXqpQr4+8xMZDF4T
7LVnQUbaCevjBklUdWYKtnfjONwUtsgTwo/MI6XyDdI8yyFJBz3AMhkDN
5kgP4tGILW1bg6PtflcmLDOUllnpakaQQFq/Q3QnBzW1DFXaFj6UMBnud
KbalNuugyQJBAMp6/5hS6uUAPJ/hvpg1TYHeV5+jrSvJY0e6GQH+eCq6xv
pQvojHA4IvdXfb1+kXdj2x0SKCexjRqdIDBfJHg7MCQQC9i0C39VDE0lJd
e1a04J7MpeDQt1kLwRBKCpPCLfVZFuTsvS3G0ZOG9Lz+nuBzmflQSYeA
4+AJ6QAHsbZnPdc3AkA/TU7lR18KcxWBAql8moV9yY5paV11bAOu4/53g
h/9c+FLVr0Ks/Vj2QSFisoqRFyCEzqH6HUloD7QWoOcaEFDaKANEdgRlz
nNil8jYQjWihKJG/sHiUz7kYF1CYusvQyI6xo39Md+SR86FBAGIoZpkjJtXJQ
yYANysg0uHF6Zvo7AkBA1TLwH55Yo4z+p875S2xXp92Wzkcm6LIDgdN
2pcLsxYx1Z2bsI0zShXg7LzaBtPIFBJXrXb4PAcziKcjCc+T8

- Односторонні хеш функції. Згенеруйте хеш для повідомлення, яке містить наступний текст: емейл-адреса.

SHA-256 найкраще розуміти як набір криптографічних хеш-функцій. Хеш-функція, яку також називають дайджестом або відбитком пальця, схожа на унікальний підпис для файлу даних або тексту.

Він не може бути прочитаний або розшифрований, оскільки допускає лише односторонню криптографічну функцію. Це дозволяє використовувати хешування для перевірки файлів, цифрових підписів, захищених повідомлень та інших додатків.

SHA-256 hash calculator

SHA-256 produces a 256-bit (32-byte) hash value.

Data

Valentyn2202@gmail.com

SHA-256 hash

1a49632fb3e6d3769f2d674dd8852fd0c9f00a32003fd51e0a3810e74439c9c0

- Повідомлення для заcodування - Valentyn2202@gmail.com
- Закодоване повідомлення -
1a49632fb3e6d3769f2d674dd8852fd0c9f00a32003fd51e0a3810e74439c9c0

4. Опишіть, що таке uuid та для чого його використовують (4 бали):

Універсально унікальні ідентифікатори (UUIDS) - це 128-розрядні числа, що складаються з 16 октетів і представлені у вигляді 32 символів з основою 16, які можуть бути використані для ідентифікації інформації в комп'ютерній системі. Ця специфікація була спочатку створена компанією Microsoft і стандартизована як IETF, так і MCE.

UUID зазвичай використовуються для ідентифікації інформації, яка повинна бути унікальною в межах системи або мережі. Їх унікальність та низька ймовірність повторення робить їх корисними для використання в якості асоціативних ключів в базах даних та ідентифікаторів для фізичного обладнання в організації. Однією з переваг UUID є те, що вони не повинні видаватися центральним органом влади, а можуть генеруватися незалежно, а

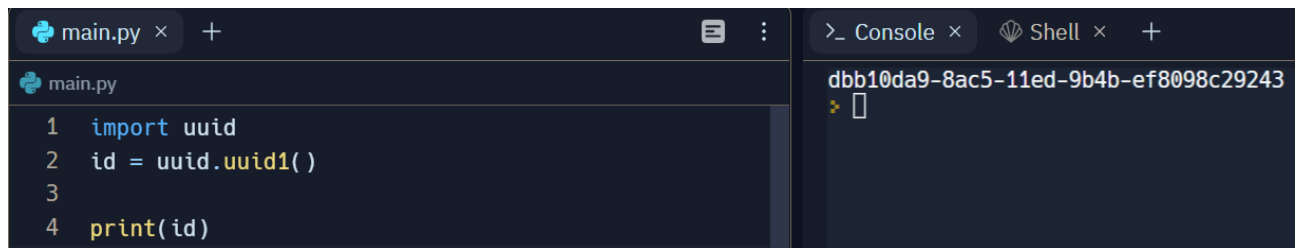
потім використовуватися в даній системі без підозри, що дублікат або зіткнення UUID був згенерований в іншому місці. Apple, Microsoft, Samsung та інші використовують UUID, або визначені специфікацією IETF, або власний варіант, для ідентифікації та відстеження обладнання як всередині компанії, так і для продажу споживачам.

- Які версії знаєте?

Загалом є 5 версій UUID:

- Версія 1. Ця версія генерується із зазначеного часу та вузла і являє собою унікальний ідентифікатор хоста, заснований на мітці часу
- Версія 2. Тут усе відбувається аналогічно до версії 1, однак, замінюються менш значущі біти. А саме, вісім біт послідовності годинника замінюються на номер локального домену, а 32 біти мітки часу замінюються на номер для вказаного локального домену. Ці біти зарезервовані для DCE Security UUID.
- Версія 3. Цей варіант створюється шляхом хешування ідентифікатора простору імен та імені. Версії 3 і 5 побудовані аналогічно, однак у версії 3 в якості алгоритму хешування використовується алгоритм дайджесту повідомлень 5 (MD5).
- Версія 4. Генерація проходить випадковим чином. Хоча даний UUID використовує випадкові байти, чотири біти використовуються для позначення версії 4, а два-три біти використовуються для позначення варіанту. Вони можуть бути створені за допомогою генератора випадкових або псевдовипадкових чисел. У цій версії використовується більше бітів, тому комбінацій UUID менше. Однак комбінацій UUID все ще достатньо, щоб уникнути можливості колізії.
- Версія 5 генерується так само, як і версія 3. Однак вона створюється з використанням алгоритму Secure Hash Algorithm 1, або SHA-1, на відміну від MD5, який версія 3 використовує для хешування. Версії 3 і 5 добре підходять для використання в якості унікальних ідентифікаторів для інформації та даних в межах простору імен системи

- Згенеруйте приклади
- Версія 1:

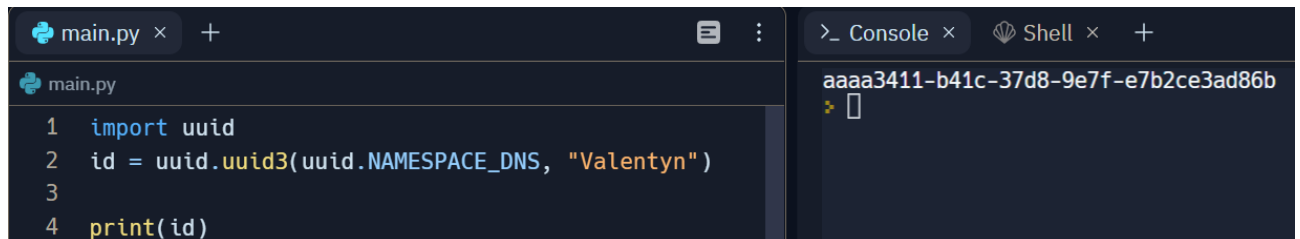


The screenshot shows a code editor with a file named `main.py`. The code is as follows:

```
1 import uuid
2 id = uuid.uuid1()
3
4 print(id)
```

The output in the console is a UUID: `dbb10da9-8ac5-11ed-9b4b-ef8098c29243`.

- Версія 3:

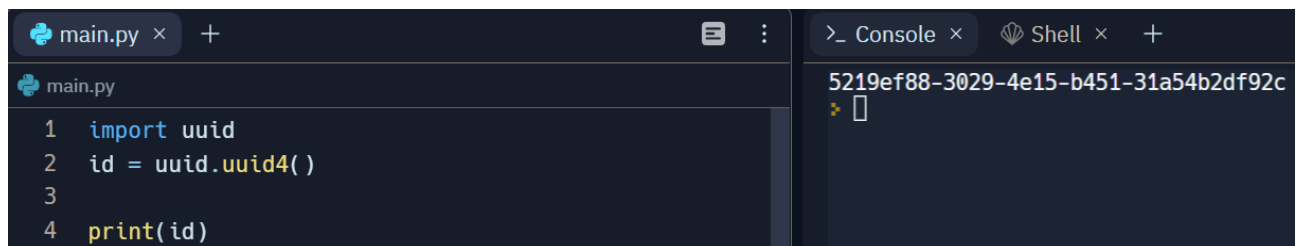


The screenshot shows a code editor with a file named `main.py`. The code is as follows:

```
1 import uuid
2 id = uuid.uuid3(uuid.NAMESPACE_DNS, "Valentyn")
3
4 print(id)
```

The output in the console is a UUID: `aaaa3411-b41c-37d8-9e7f-e7b2ce3ad86b`.

- Версія 4:



The screenshot shows a code editor with a file named `main.py`. The code is as follows:

```
1 import uuid
2 id = uuid.uuid4()
3
4 print(id)
```

The output in the console is a UUID: `5219ef88-3029-4e15-b451-31a54b2df92c`.

- Версія 5:



The screenshot shows a code editor with a file named `main.py`. The code is as follows:

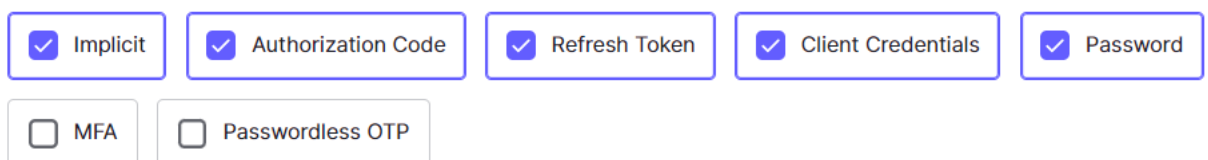
```
1 import uuid
2 id = uuid.uuid5(uuid.NAMESPACE_DNS, "Valentyn")
3
4 print(id)
```

The output in the console is a UUID: `d286b36a-80fd-5f67-bb02-df389922fe65`.

5. Перерахуйте основні grant types у OAuth2 протоколи (4 бали).

Якщо зайти на сайт `auth0` у налаштування Application, то можна побачити ось такі grant types:

Grants



The screenshot shows the 'Grants' configuration interface. It contains five checkboxes, all of which are checked:

- ☒ Implicit
- ☒ Authorization Code
- ☒ Refresh Token
- ☒ Client Credentials
- ☒ Password

Below these, there are two more checkboxes, both of which are unchecked:

- ☐ MFA
- ☐ Passwordless OTP

- Дайте стислий опис, для чого вони використовуються.
- Authorization – Клієнтська програма і служба OAuth спочатку використовують перенаправлення для обміну серією HTTP-запитів на основі браузера, які ініціюють потік. Користувача запитують, чи погоджується він на запитований доступ. Якщо він погоджується, клієнтській програмі надається "код авторизації". Потім клієнтська програма обмінюється цим кодом зі службою OAuth, щоб отримати "маркер доступу", який вона може використовувати для здійснення викликів API для отримання відповідних даних користувача.

Вся комунікація, яка відбувається з моменту обміну кодом/токеном і далі, надсилається від сервера до сервера по захищеному, попередньо налаштованому зворотному каналу і, отже, невидима для кінцевого користувача. Цей захищений канал встановлюється, коли клієнтська програма вперше реєструється в службі OAuth. В цей час також генерується `client_secret`, який клієнтська програма повинна використовувати для аутентифікації при відправці цих запитів між сервером і сервером

- Implicit – Неявний тип гранту є набагато простішим. Замість того, щоб спочатку отримати код авторизації, а потім обміняти його на маркер доступу, клієнтська програма отримує маркер доступу відразу після того, як користувач дає свою згоду.

Ви можете задатися питанням, чому клієнтські програми не завжди використовують неявний тип гранту. Відповідь відносно проста - він набагато менш безпечний. При використанні неявного типу надання вся комунікація відбувається через перенаправлення браузера - немає захищеного зворотного каналу, як у потоці коду авторизації. Це означає, що конфіденційний токен доступу та дані користувача є більш вразливими до потенційних атак

- Refresh Token – Тип надання Refresh Token використовується клієнтами для обміну токена оновлення на токен доступу, коли термін дії токена доступу закінчився.

Це дозволяє клієнтам продовжувати мати дійсний токен доступу без подальшої взаємодії з користувачем

- Client Credentials – Тип надання "Облікові дані клієнта" використовується клієнтами для отримання маркера доступу поза контекстом користувача.

Зазвичай це використовується клієнтами для доступу до ресурсів про себе, а не для доступу до ресурсів користувача

- Password – Грант "Пароль" є одним з найпростіших грантів OAuth і передбачає лише один крок: додаток представляє традиційну форму входу в систему з ім'ям користувача та паролем для збору облікових даних користувача і робить POST-запит до сервера для обміну пароля на токен доступу.

- Наведіть приклади.

- Authorization:

POST https://YOUR_DOMAIN/oauth/token

Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&client_id=YOUR_CLIENT_ID&client_secret=YOUR_CLIENT_SECRET&code=AUTHORIZATION_CODE&redirect_uri=https://YOUR_APP/callback

- Client Credentials:

POST https://YOUR_DOMAIN/oauth/token

Content-Type: application/x-www-form-urlencoded

audience=API_IDENTIFIER&grant_type=client_credentials&client_id=YOUR_CLIENT_ID&client_secret=YOUR_CLIENT_SECRET

- Refresh Token:

POST https://YOUR_DOMAIN/oauth/token

Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&client_id=YOUR_CLIENT_ID&client_secret=YOUR_CLIENT_SECRET&refresh_token=YOUR_REFRESH_TOKEN

- Password:

POST https://YOUR_DOMAIN/oauth/token

Content-Type: application/x-www-form-urlencoded

grant_type=password&username=USERNAME&password=PASSWORD&audience=API_IDENTIFIER&scope=SCOPE&client_id=YOUR_CLIENT_ID&client_secret=YOUR_CLIENT_SECRET

6. Назвіть який формат даних використовується для передачі інформації в SAML протоколі (2 бали)?

Мова розмітки тверджень безпеки (Security Assertion Markup Language, або SAML) - це стандартизований спосіб повідомити зовнішнім додаткам і службам, що користувач є тим, за кого себе видає. SAML робить технологію єдиного входу (SSO) можливою, надаючи спосіб автентифікації користувача один раз, а потім передавати цю автентифікацію декільком додаткам. Найновіша версія SAML - SAML 2.0.

Уявіть собі автентифікацію SAML як ідентифікаційну картку: короткий, стандартизований спосіб показати, ким є людина. Замість того, щоб, скажімо, проводити серію тестів ДНК для підтвердження чиєїсь особи, можна просто поглянути на її посвідчення особи.

- Стисло опишіть, як ви розумієте різницю SP-initiated, IDP-initiated підходах SAML.

IdP визначає, чи існує сеанс Windows, і отримує облікові дані користувача, який увійшов в систему. Він генерує відповідь SAML.

Постачальник ідентифікаційних даних управляє ідентифікацією та атрибутами користувача (IdP). А додаток, до якого користувач хоче увійти та отримати доступ, є вашим постачальником послуг (SP).

- Ініційований SP вхід

Запит на вхід ініціюється через ваш додаток, до якого ви хочете отримати доступ.

Користувач перенаправляється на IdP, де ви можете побачити сторінку входу для аутентифікації користувачів

IdP визначає, чи існує сеанс Windows, і отримує облікові дані користувача, який увійшов в систему. Він генерує SAML-відповідь.

За допомогою SAML-відповіді від IdP служба реєструє користувача в додатках.

- Ініційований IdP вхід

При вході, ініційованому IdP, користувач входить безпосередньо в IdP, а не в інформаційну панель програми.

За допомогою SAML-відповіді від IdP відбувається вхід користувача в додатки.