**Міністерство освіти і науки України Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Факультет інформатики та обчислювальної техніки Кафедра обчислювальної техніки**

# Модульна контрольна робота

з дисципліни

«Безпека програмного забезпечення»

Виконав:

студент групи ІП-05

Гапій Денис Едуардович

Номер залікової: 0504

Перевірив:

доц. Волокита А. М.

Київ 2023

# Тема: «Системи безпеки програм і даних.»

**Виконання:**

1. Згенеруйте authorization хедер для Basic Authentication flow (3 бали).
   - В якості пароля використовуйте цифри заліковки ABCD.
   - В якості логіна - прізвище латиницею.

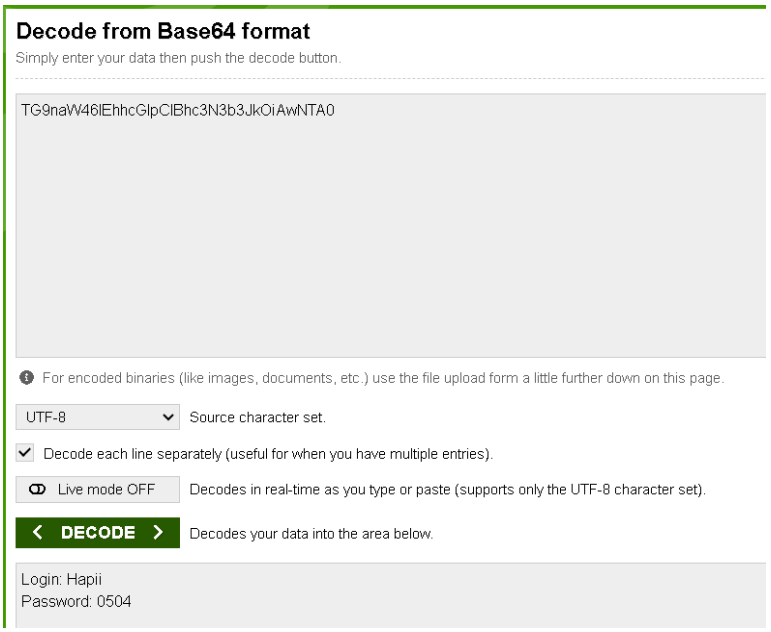
Лістинг для генерації, мовою JS:

```
const login = "Hapii";

const password = "0504";

const credentials = `Login: ${login}\nPassword: ${password}`;

const base64Credentials = btoa(credentials);

const authorizationHeader = `Basic ${base64Credentials}`;

console.log(authorizationHeader);
```

Результат:

```
PS S:\Dev\Studying\KPI-Studying\7th semester\Software Security\lab1\jwt_auth> node index.js
Basic TG9naW46IEhhcGlpClBhc3N3b3JkOiAwNTA0
PS S:\Dev\Studying\KPI-Studying\7th semester\Software Security\lab1\jwt_auth>
```

Результат декодування:



2. За допомогою сайту jwt.io створить JWT токен з наступними клеймами (3 бали):

   ● name - ім’я та прізвище (латиниця);

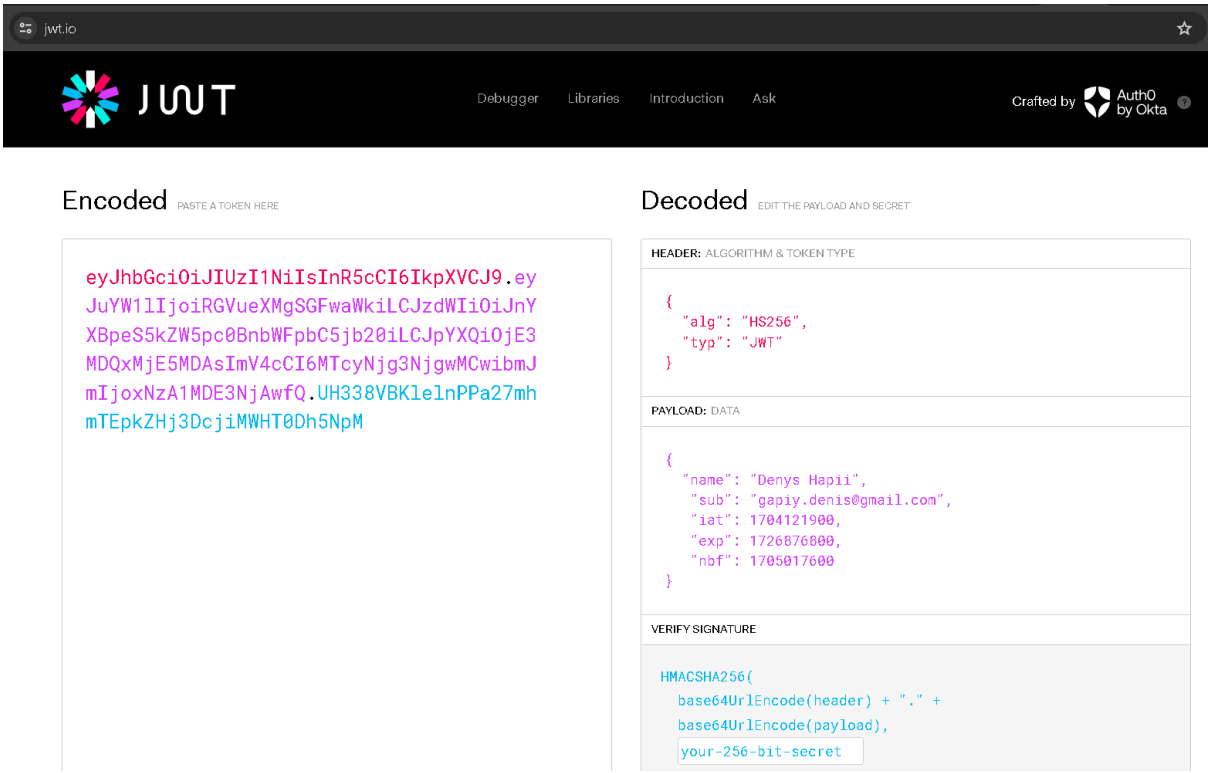
   ● sub (subject) – email;

   ● iat (issued at time) – час, коли токен було створено (поточний час);

   ● exp (expiration time) – час іспиту в системі Кампус + кількість днів до дня народження з початку року.

   ● nbf (not before time) – час іспиту в системі Кампус.

Результат сайту:




Токен:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiRGVueXMgSGFwaWkiLCJzdWIiOiJnYXBpeS5kZW5pc0BnbWFpbC5jb20iLCJpYXQiOjE3MDQxMjE5MDAsImV4cCI6MTcyNjg3NjgwMCwibmJmIjoxNzA1MDE3NjAwfQ.UH338VBKlelnPPa27mhmTEpkZHj3DcjiMWHT0Dh5NpM

3. Назвіть приклади алгоритмів та використайте для шифрування \ кодування наступного тексту: номер заліковки, name - ім’я та прізвище (латиниця), емейл-адреса (4 бали):

● Вільне кодування\декодування інформації.

base64:
Z3JhZGVib29rOiAwNTA0LCANCm5hbWU6IOKAnEhhcGlp4oCdLCANCmVtYWlsOiDigJxnYXBpeS5kZW5pc0BnbWFpbC5jb23igJ0

● Симетричні алгоритми шифрування (додайте ключ до відповіді).

AES (Advanced Encryption Standard):
P8fAVuwlzDCikf7ax6gBalOp83RCsWyDrrPbjdiPx0NVvy7fNh7ehydS0MImsIcp7vOye8WftuevNG9PWJG7+FZS3cNJ6/Qh2kgMJG5MyOo=

ключ: security

● Асиметричні алгоритми шифрування. Згенеруйте відкритий і секретні ключі (додайте ключі до відповіді).

RSA (Rivest–Shamir–Adleman):
MV/a1SgWhfJg67p/0lXvQHl/a3e5Uj1cTtBuYU7BlCbn2G3t+MU9iNmskAUq0LxeR/HW5AzJozhtJmEGYG9ZzYWcI+DgoJy/XY2wYyAc27eKV92axKgpdg6kdB8gy9oWS6VuYfwpsohFSThZzgwdlDSkxfz3rLTdsbbkjz0/tc4=

private key:
MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBALlMByly2xuXyo11cetW1f0E412LS57s+s93Weuo76y8JfWVqrYI+3GGMNkudXuQp9lo7loQM6KrbvsmLeRlR1r9pu3Q4rScR5uNk8pHqAa74IHqzzquHMzZhFsTx0BmRxxptFphQNkqye43x8jsYdjQ9QUE+JQT8imqIrhfXfedAgMBAAECgYAqROOlWuQMqVW1a0MvckGiVEkhf7Mib+DPDuTeU01JVCo8mYW1vNrPDNN715NJOrhvafX7kLKWuxC4Df+OCGR8Q9HSKOF8aqHb26x04KLhjCThyAYfhbWv0TnB1jzw0mhdNrLy3pEkUNV+BkRxvNU57CGF8oHLwmYwzHLxGN89qQJBAODHIlTQ3RhluYu2WMPmp1enyTxNSKXCdAUIg63/Zxl4AR+hfKurR4Z4hgdp8XGmYo+QnYs9Kfns4GEPNGa3xWcCQQDTCP609fn3U3BqlqT4Tb/8cb6o0vuXLbJE0qP9/1iKZmJecrtvlQpDDvPAZ0s/kgJxAXY8VKrcPTEhSPvOc1RbAkEA27LHY48iCw6iZGQ+LnvrZEmi70W6lAmTzLedvRu9ca9RqdD1QfWPW2fB2M08KJEwFJKM6eNGe/mC9lseJKfKUQJAMlVP4rtUPkAhbNq3SB3LL5u1fLCtnxYgEbgnsvoHFSNsTNiSif+DLhFP49D/Ko9Zk7hkiekrw1G4+RZeMJRjRwJAeuHllS5q24KXEHikUM0qOhtOPhfDUJnXtwMO7c3vTLd6japg/afVCyGyL35C3o8p5IT6BTkNc/7w3Sb7LjSluQ==

public key:
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC5TAcpctsbl8qN
dXHrVtX9BONdi0ue7PrPd1nrqO+svCX1laq2CPtxhjDZLnV7kKfZaOyKE
DOiq277Ji3kZUda/abt0OK0nEebjZPKR6gGu+CB6s86rhzM2YRbE8dAZ
kccabRaYUDZKsnuN8fI7GHY0PUFBPiUE/IpqiK4X133nQIDAQAB

- Односторонні хеш функції.

  SHA-512:
  285f08c35784552f342f4fca5eee81c058883cd9b2935901b77dbbf6
  7f95d953483ca1cb2c123c04ad5f7ca855ae64e5bccfe064e62424cf
  2e342cb8925c677a

текст - gradebook: 0504, name: “Hapii”, email: “gapiy.denis@gmail.com”

4. Опишіть, що таке uuid та для чого його використовують (3 бали):

   - Згенеруйте унікальні приклади для кожної версії. Надайте стислий опис

UUID - Universally Unique Identifier (Універсальний унікальний ідентифікатор), що складається з 32 символів (128 біт)

UUID використовуються в різних цілях, наприклад:

- Як первинні ключі в базах даних. UUID забезпечують унікальність ідентифікаторів записів у базі даних, що важливо для забезпечення цілісності даних.
- Як імена для файлів і об'єктів. UUID можна використовувати для створення унікальних імен для файлів, об'єктів і ресурсів в інфраструктурі.
- Як ідентифікатори для користувачів і пристроїв. UUID можна використовувати для ідентифікації користувачів і пристроїв у розподілених системах.

Приклади:

- time + mac:
  - 550e8400-e29b-41d4-a716-446655440000
- system identificator:
  - 6ba7b810-9dad-11d1-80b4-00c04fd430c8
- MD5 hash:

    - 4124bc0a-11e1-11e1-9ab4-0002a5d5c51b
- SHA-1 hash:
    - 886313e1-3b8a-5372-9b90-0c9aee199e5d

5. Перерахуйте основні grant types у OAuth2 протоколи (3 бали).

   ● Наведіть приклади запитів до Auth0 identity server (використовуйте аккаунт, який використовували під час лабораторних робіт). Надайте стислий опис.

Основні grant types: client_credentials, password, refresh_token,

authorization_code.

1) Запит на отримання application access token, повертається також час життя токену та scope дій, дозволених для виконання, наприклад робота з користувачами та їхніми ролями. `method: 'POST',`

```
  url: `${process.env.AUTH0_URL}/oauth/token`,

  headers: { 'content-type': 'application/json' },

  form: {

    client_id: process.env.AUTH0_CLIENT_ID,

    client_secret: process.env.AUTH0_CLIENT_SECRET,

    audience: `${process.env.AUTH0_URL}/api/v2/`,

    grant_type: "client_credentials",

  }
```

2) Запит використовує метод PATCH для зміни пароля користувача в системі Auth0

```
method: "PATCH",

  url:
`${process.env.AUTH0_URL}/api/v2/users/auth0%7C12344443434`,//auth0|1234444343
4

  headers: {

    "content-type": "application/json",
```

```
    authorization: `Bearer ${process.env.AUTH0_TOKEN}`
  },
  form: {
    //client_id: process.env.AUTH0_CLIENT_ID,
    //client_secret: process.env.AUTH0_CLIENT_SECRET,
    //audience: `${process.env.AUTH0_URL}/api/v2/`,
    //grant_type: "client_credentials",
    //email: 'gapiyka@gmail.com',
    "connection": "Username-Password-Authentication",
    "password": "new!Password--1-12313123"
  }
```

3) Отримання user access token, refresh token, а також час життя та scope

```
method: 'POST',
    url: `${process.env.AUTH0_URL}/oauth/token`,
    headers: { 'content-type': 'application/json' },
    form: {
        refresh_token: process.env.AUTH0_REFRESH,
        client_id: process.env.AUTH0_CLIENT_ID,
        client_secret: process.env.AUTH0_CLIENT_SECRET,
        code: process.env.AUTH0_CODE,
        audience: `${process.env.AUTH0_AUDIENCE}`,
        grant_type: "refresh_token",
        scope: "offline_access"
   }
```

6. Формат даних для передачі інформації в SAML протоколі (4 бали).

   ● Згенеруйте та наведіть SAML request та responce, де issuer - домен auth0 (який використовували під час лабораторних робіт), також додайте Assertion з Вашим email та номером заліковки.

SAML Request:

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"

    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"

    ID="_f01e2928-e6e8-496f-875c-44353976d68d"

    Version="2.0"

    IssueInstant="2024-01-03T12:00:00Z"

Destination="https://dev-gvsgio1zxq8w4q33.us.auth0.com/saml2/idp/SSOSer
vice"

    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST">

 <saml:Issuer>https://dev-gvsgio1zxq8w4q33.us.auth0.com</saml:Issuer>

    <samlp:NameIDPolicy AllowCreate="true"
Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"/>

    <samlp:RequestedAuthnContext>


<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Passw
ordProtectedTransport</saml:AuthnContextClassRef>

    </samlp:RequestedAuthnContext>

</samlp:AuthnRequest>
```

SAML Response:

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"

    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
```

```
    ID="_222e3928-e6e8-496f-875c-44353976d68d"
    Version="2.0"
    IssueInstant="2024-01-03T12:00:00Z"
    Destination="https://example.com/saml2/sp/acs"
    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST">
    <saml:Issuer>https://dev-gvsgio1zxq8w4q33.us.auth0.com</saml:Issuer>
    <samlp:Status>
        <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
    <saml:Assertion ID="_00000000-0000-0000-0000-000000000000"
        Version="2.0"
         IssueInstant="2024-01-03T12:00:00Z"
         Issuer="https://dev-gvsgio1zxq8w4q33.us.auth0.com">
        <saml:Subject>
            <saml:NameID
Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">gapiy.denis@gmai
l.com</saml:NameID>
            <saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                <saml:SubjectConfirmationData
NotOnOrAfter="2024-01-03T13:00:00Z"
Recipient="https://example.com/saml2/sp/acs"/>
            </saml:SubjectConfirmation>
        </saml:Subject>
        <saml:Conditions NotBefore="2024-01-03T12:00:00Z"
NotOnOrAfter="2024-01-03T13:00:00Z">
            <saml:AudienceRestriction>
                <saml:Audience>https://example.com/saml2/sp</saml:Audience>
```

```
            </saml:AudienceRestriction>
        </saml:Conditions>
        <saml:AttributeStatement>
            <saml:Attribute Name="gradebook"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
               <saml:AttributeValue>0504</saml:AttributeValue>
           </saml:Attribute>
           <saml:Attribute Name="email"
Format="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml:AttributeValue>gapiy.denis@gmail.com</saml:AttributeValue>
      </saml:Attribute>
       </saml:AttributeStatement>
   </saml:Assertion>
</samlp:Response>
```