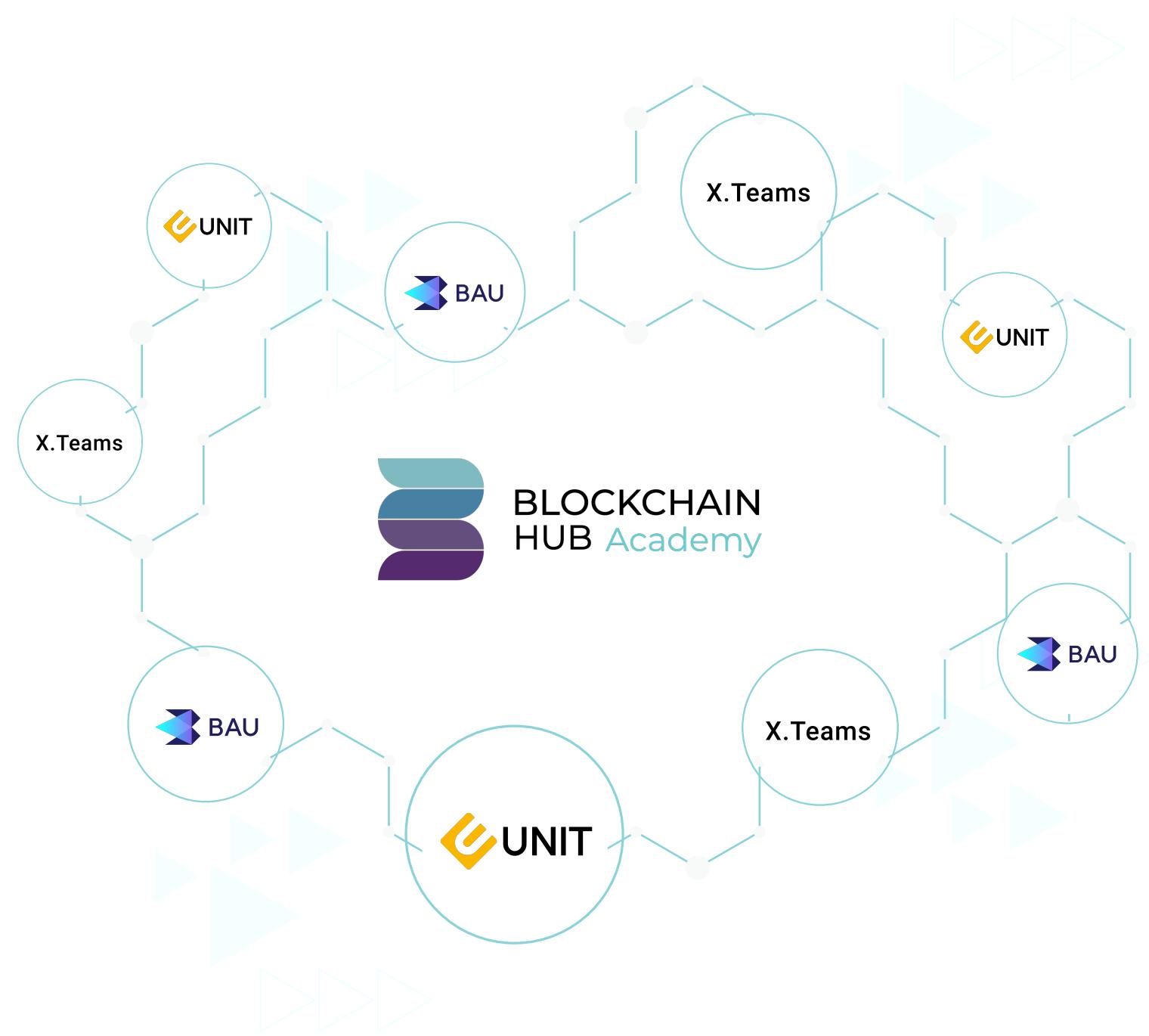




BLOCKCHAIN  
HUB Academy



PS: 1  
Created: 2019

```
static bool init(CURL *&conn, char *url)
{
    CURLcode code;
    conn = curl_easy_init();
    if (conn == NULL)
        fprintf(stderr, "Failed to create CURL connection\n");
    exit(EXIT_FAILURE);

    code = curl_easy_setopt(conn, CURLOPT_ERRORBUFFER,
                           errorBuffer);
    if (code != CURLE_OK)
        fprintf(stderr, "Failed to set error buffer [%d]\n",
                code);

    return false;

    code = curl_easy_setopt(conn, CURLOPT_URL, url);
    if (code != CURLE_OK)
        fprintf(stderr, "Failed to set URL [%s]\n", errorBuffer);

    return false;

    code = curl_easy_setopt(conn, CURLOPT_FOLLOWLOCATION,
                           1);
    if (code != CURLE_OK)
        fprintf(stderr, "Failed to set redirect option [%s]\n",
                errorBuffer);

    return false;

    code = curl_easy_setopt(conn, CURLOPT_WRITEFUNCTION,
                           write),
    if (code != CURLE_OK)
```



```
static void StartElement(
    Context *context,
    const char *name,
    const char **attributes,
    void *userData)
{
    if (COMPARE(name, "title"))
        context->title = context->addTitle();
    else if (COMPARE(name, "description"))
        context->addDescription();
}

// libxml end element
static voidEndElement(
    Context *context,
    const char *name,
    void *userData)
{
    if (COMPARE(name, "title"))
        context->addTitle();
}

// Text handling
static void handleCharacterData(
    Context *context,
    const char *data,
    int length,
    void *userData)
{
    if (context->addTitle())
        context->title.append(data, length);
}

// libxml PCDATA callback
static void CharacterDataHandler(
    void *userData,
    const XML_Char *data,
    int length)
```

# CONTENTS

I	General Instructions	4
I	PART 00: Introduction	5
II	PART 01: Ethernaut	6

# GENERAL INSTRUCTIONS

## **Technical organization:**

Submit your finished tasks to the GIT repository.  
<https://xteams-gitlab.unit.ua/module-N-login>

Only work submitted to the repository will be considered for mentor-evaluation. Any extraneous files will be considered against you if this is not justified by any serious cause.

## **Deadline:**

Access to the repository for making records closes after 7 days since tasks were presented at 09:01:00 AM GMT+2

XBIP: 6 - 04:03:2018 at 09:01:00 AM GMT+2

# INTRODUCTION

**“With great power comes great responsibility”**

This week will be connected with smart contracts security in Ethereum blockchain. It's a practice by the exploring Ethernaut. This is a Web3/Solidity based wargame, to be played in the Ethereum Virtual Machine. Each level is a smart contract that needs to be 'hacked'.

Please make sure you use the Ethereum testnet and only ONE address under your control. Peer-review will be includes validation by this address. Don't forget to add the address to your repo.

Don't be shy to read Ethereum Smart Contract Best Practices - <https://consensys.github.io/smart-contract-best-practices/>

# PART 01: Ethernaut

This week can be completed by different levels of difficulty.

## Izi

Complete all levels of Ethernaut game - <https://ethernaut.zeppelin.solutions>

## Mid

Complete all levels of Ethernaut game and create one personal vulnerable contract in the same way as previous. The reviewer will be try to ‘hack’ your contract during the review process

## Hard

Complete all levels of Ethernaut game, create your own vulnerable contract and deploy full Ethernaut project with your additional personal contract from previous level (include it to the Ethernaut architecture with corresponding game description, patterns etc.) to the Ethereum testnet. The source code is here - <https://github.com/OpenZeppelin/ethernaut>. Please add to your repo all necessary addresses of deployed contracts.