

# Writeup CTF @meninadecybersec

---

## Categoria Phishing Analysis

- O SOC sofre...(1) - Valendo 1300 Pontos

Essa challenge consistia em capturar o endereço de IP de quem enviou o e-mail.

Baixando o arquivo que estava disponível para fazermos a challenge e abrindo ela com o bloco de notas, podemos filtrar por "IP is" e aparecia um endereço de IP (209.85.222.173) no qual é a flag deste desafio.

Flag => MCS{209.85.222.173}

```
Transport; Thu, 20 Oct 2022 02:54:18 +0000
Authentication-Results: spf=pass (sender IP is 209.85.222.173)
smtp.mailfrom=gmail.com; dkim=pass (signature was verified)
header.d=gmail.com; dmarc=pass action=none header.from=gmail.com; compauth=pass
reason=100
Received-SPF: Pass (protection.outlook.com: 209.85.222.173 as permitted sender)
client-ip=209.85.222.173; helo=mail-qk1-f173.google.com
Received: from mail-qk1-f173.google.com [209.85.222.173]
      by mx2nam12ft102.mail.protection.outlook.com [15.20.57.23.11] via Frontend Transport
      [15.20.57.23.11] with ESMTP id a18so12058286qko.0
      X-IncomingTopHeaderMarker:
      OriginalChecksum:A019414A0A48269D5790D09247C988DF2545E5E9ADBB1A4CBEE12CADD42271EE;UpperCasedChecksum:A8068CF1396DEC315608F619
Received: by mail-qk1-f173.google.com with SMTP id a18so12058286qko.0
```

- O SOC sofre... (2) - Valendo 1300 Pontos

Essa challenge consistia em capturar o e-mail que preenche o campo 'From:'.

No mesmo arquivo que baixamos, podemos filtrar por "@gmail.com", e a flag é o primeiro email que aparece "santormcconnellw9660@gmail.com"

```

X-Google-Smtp-Source: AMsMyM6HNEGNpo0tVUn0SmCz0jeyrXARp2WG8hH1VF6pIrWGV+mhXTLfpbiewpFpgpZZK6JImn20===
X-Received: by 2002:a05:620a:1729:b0:6ee:c:f01:6810 w
Wed, 19 Oct 2022 19:54:17 -0700 (PDT)
Return-Path: santormcconnellw9660@gmail.com
Received: from usa.domain ([20.172.166.105])
by smtp.gmail.com with ESMTPSA id h19-20020a
for <rosinha@hotmail.com>
(version=TLS1_2 cipher=ECDHE-ECDSA-AES128-GO)
Wed, 19 Oct 2022 19:54:17 -0700 (PDT)
Message-ID: <f1ab89b26a633907e72f0e050f2c293@gmail.com>
From: "Pontos ItauCartoes2.0 Liberados][048582186378611072]" <santormcconnellw9660@gmail.com>
To: <rosinha@hotmail.com>
<
```

Flag => MCS{santormcconnellw9660@gmail.com}

---

- O SOC sofre... (3) - Valendo 1300 Pontos

Nessa ultima challenge da categoria phishing analysis, teremos que identificar quem foi a vítima desse e-mail.

No mesmo arquivo, procurando por todos os emails que tinha, encontrei um "rosinha@hotmail.com" no qual é a vitima deste phishing.

```

Authentication-Results: spf=pass (sender IP is 209.85.222.173)
smtp.mailfrom@gmail.com: dkim=none (signature was not verified)
header.d=gmail Localizar × h=pass
reason=100
Received-SPF: D: Localizar: rosinha@hotmail.com Localizar Próxima es
209.85.222.173
client-ip=209.85.222.173
Received: from r [Diferenciar maiúsculas de minúsculas]
  (MW2NAM12FT102.i [Acima] SMTP
  Server (version 1.5.20.5723.11 via Frontend Transport; Thu, 20 Oct 2022 02:54:18 +0000)
X-IncomingTopHeaderMarker:
OriginalChecksum:A019414A0A48269D5790D09247C988DF2545E5E9ADBB1A4CBEE12CADD42271EE;UpperCasedChecksum:A8068CF1396DEC315608F6194A7E3CF85
Received: by mail-qk1-f173.google.com with SMTP id a18sn12058286qko.0
  for <rosinha@hotmail.com>; Wed, 19 Oct 2022 19:54:18 -0700 (PDT)
DKIM-Signature: v=1, a=rsa-sha256, c=relaxed/relaxed,
  d=gmail.com; s=20210112;
  h=mime-version:date:subject:to:from:message-id:from:to:cc:subject
  :date:message-id:reply-to;
  bh=2HY2r6Bc+06rHxRzPAXJ2CyRQ9h8pdcRuoFubzeAeXc=;
  b=cVwKoJejuxwiEQ9huKrRdkDweHIJTidQgI/XptrzgCdM93w/We0dL+PVLgqjZUFh6u
  2Xp7554YoIChU++htokdfFmMceyZGRW5VUX85AtKiXiQ5uC3o5pTXUZp2AP9bDnQLa
  Hwvip2c2TWP1/lWqwPmVhaB1V5Ld/g1+bNQJP+rteR18xEhJgCBUJEvqepdVH6HXxSWI

```

Flag => MCS{rosinha@hotmail.com}

---

## Categoria Easy

- Nomeie essa tool (2) - Valendo 500 Pontos

Ao baixar a imagem que estava na challenge, podemos ver a imagem de uma famosa e provavelmente uma das mais conhecidas no mundo da segurança da informação, não só da segurança da informação, mas na TI em geral, que é o famoso nmap.

Flag => MCS{nmap}

---

- Nomeie essa tool - Valendo 550 Pontos

Ao baixar a imagem que estava disponível no challenge, podemos ver uma outra ferramenta que é muito famosa, o Burp Suite, e que por padrão ela já vem no kali linux.

Flag => MCS{burp\_suite}

---

- Nomeie essa tool (3) - Valendo 550 Pontos

Essa challenge eu resolvi pesquisando pelo nome "endianness", que é o nome que estava na imagem ao baixarmos ela, pesquisando por "endianness" no google podemos nos deparar com big endian (no qual é a flag) e little endian, mas o que é isso ?

Big endian -> extremidade maior primeiro  
Little endian -> extremidade menor primeiro

Olhando a imagem outra vez, podemos perceber que realmente é big endian, e quem quiser fica a criterio pesquisar sobre o conceito de endianness xD.

Flag => MCS{big\_endian}

---

- Hackerman - Valendo 800 Pontos

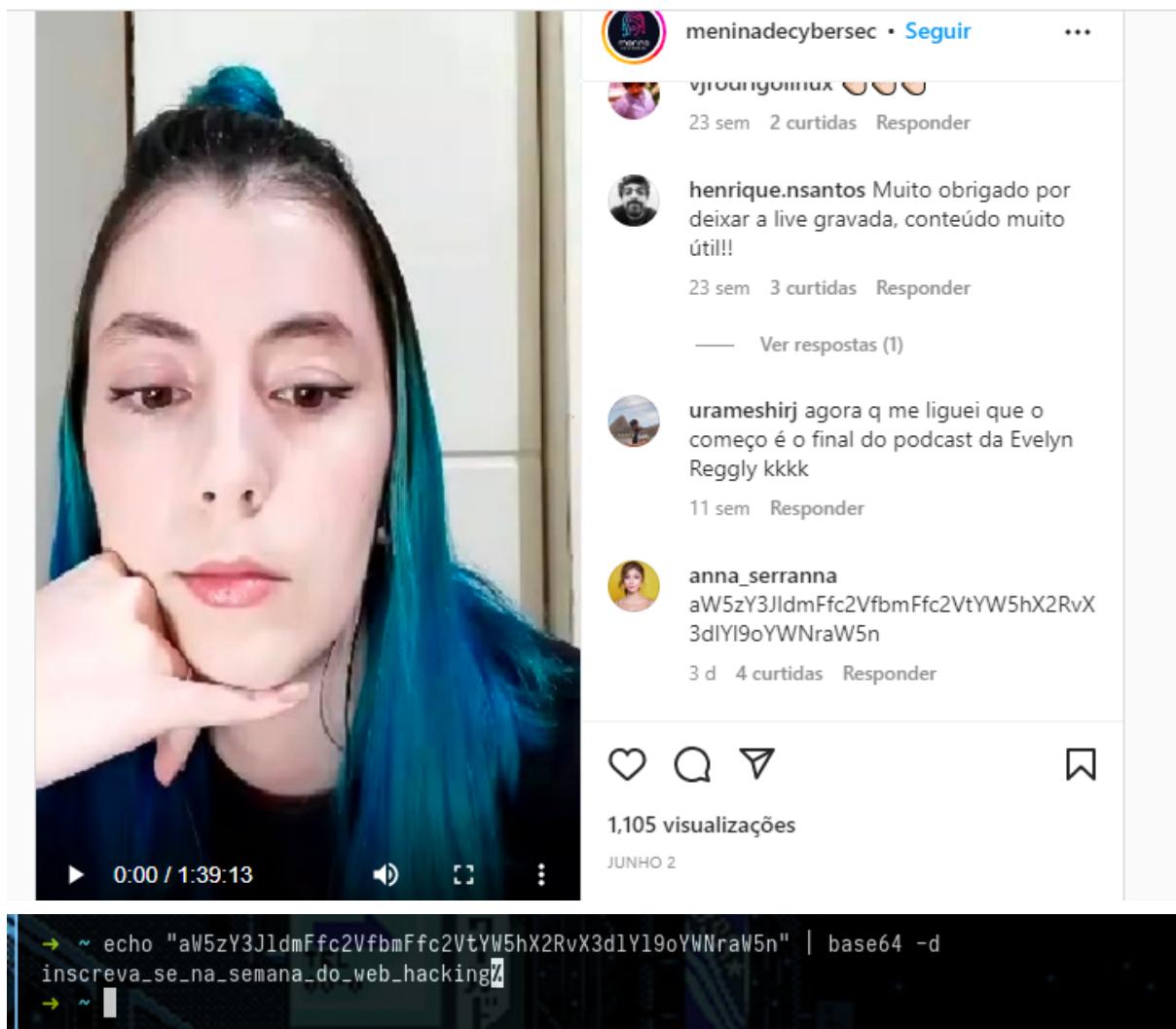
Esta challenge consistia em baixar a imagem do challenge, e descobrir qual é a série. De cara podemos perceber que é a famosa e clássica série de hackers (e uma das minhas séries favoritas hahaha) mr robot.

Flag => MCS{mr\_robot}

- 
- Fá número 1 da menina de CyberSec - Valendo 1500 Pontos

Essa challenge eu quebrei muito a cabeça, e no final era algo mais simples do que todos imaginávamos. Nessa chall precisamos descobrir duas coisas, a primeira é "qual a primeira live da menina de cybersec", e a segunda é decodar um base64.

Passo a passo: Indo no perfil da meninadecybersec, podemos ver alguns reels, no qual a primeira live que a menina de cybersec fez está lá (<https://www.instagram.com/tv/CeUlmDXILJj/?igshid=YmMyMTA2M2Y=>) , depois de encontrar a live, podemos ver o comentário de um usuário chamado "anna\_serranna", no qual é um base64, depois que acharmos o base64, podemos jogar essa hash no "<https://base64decode.org/>", depois de decodar é só pegar a flag e pontuar xD.



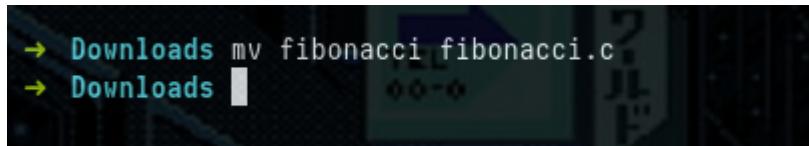
Flag => MCS{inscreva\_se\_na\_semana\_do\_web\_hacking}

---

## Categoria Programming

- Fibowhat - Valendo 550 Pontos

Antes de começar precisamos baixar o arquivo com o code em C e mover ele para fibonacci.c.

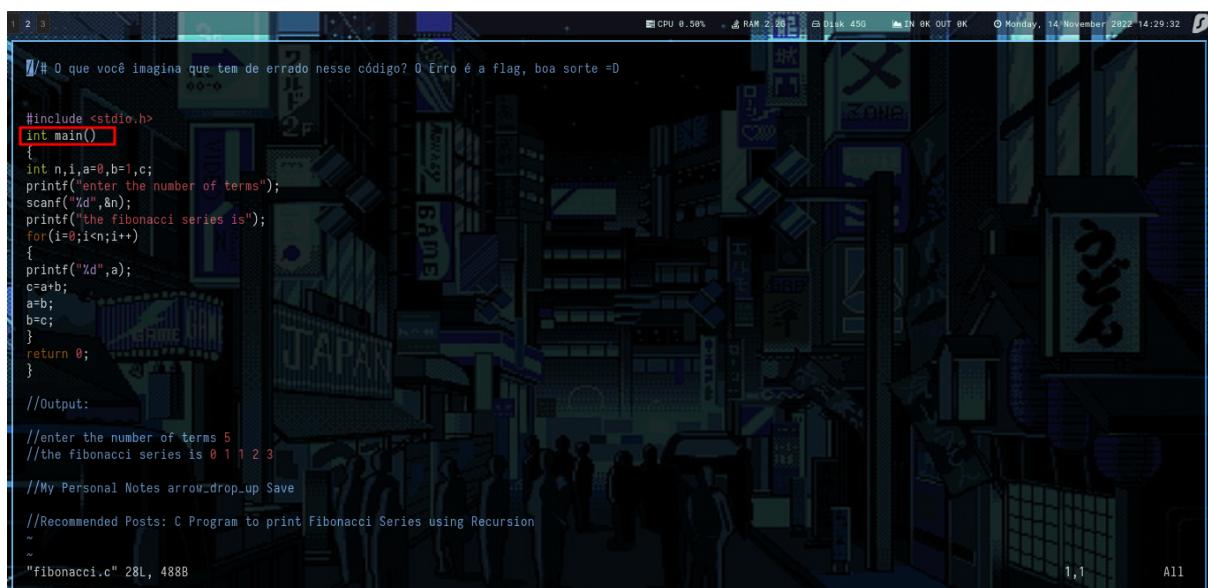


```
→ Downloads mv fibonacci fibonacci.c
→ Downloads
```

Analisando o código em C, logo na cara podemos notar que está faltando um main, e o código está com algumas frases que é só colocarmos um comentário nela ou simplesmente apagá-la do código.

Depois de adicionar um main do lado do int, e comentar as frases escritas que não fazem parte do código, podemos compilar e rodar tranquilamente em nossa máquina.

A flag é o que estava faltando no código que é o main.



```
/*# O que você imagina que tem de errado nesse código? O Erro é a flag, boa sorte =D

#include <stdio.h>
int main()
{
    int n,i,a=0,b=1,c;
    printf("enter the number of terms");
    scanf("%d",&n);
    printf("the fibonacci series is");
    for(i=0;i<n;i++)
    {
        printf("%d",a);
        c=a+b;
        a=b;
        b=c;
    }
    return 0;
}

//Output:
//enter the number of terms 5
//the fibonacci series is 0 1 1 2 3
//My Personal Notes arrow_drop_up Save
//Recommended Posts: C Program to print Fibonacci Series using Recursion
//
"fibonacci.c" 28L, 488B
```

Flag => MCS{main}

---

- Network - Valendo 550 Pontos

Este desafio eu demorei um pouco para conseguir resolver, mas depois de pensar um pouco fora da caixa consegui.

Baixando o código em C, podemos notar que está incluindo a biblioteca “netinet/in.h ( que contém definições para Internet Protocol family ) ”, e essa biblioteca vem incluso o htonl que ele converte o inteiro sem sinal “host long” do order bytes do host para o order bytes da rede, então essa era a parte do código que estava faltando no qual era o objetivo deste challenge.

Links de referências;

- <https://linux.die.net/man/3/htonl>
- <https://www.gta.ufrj.br/ensino/eel878/sockets/htonsman.html>

Flag => MCS{htonl}

---

- Keep calm, don't panic! - Valendo 1500 Pontos

Baixando o arquivo em .txt, podemos notar que é um brainfuck, nela também tem o nome do criador dessa linguagem que é o Urban Müller.

Para resolvemos essa challenge basta copiar o brainfuck que está no .txt, e jogar ela em um brainfuck interpreter.

No qual eu usei este site -> <https://www.dcode.fr/brainfuck-language>

Depois de copiar o brainfuck do txt, e colar no site que eu citei acima, ele nos trará a flag, que é "Hello world!".

Flag => MCS{Hello world!}

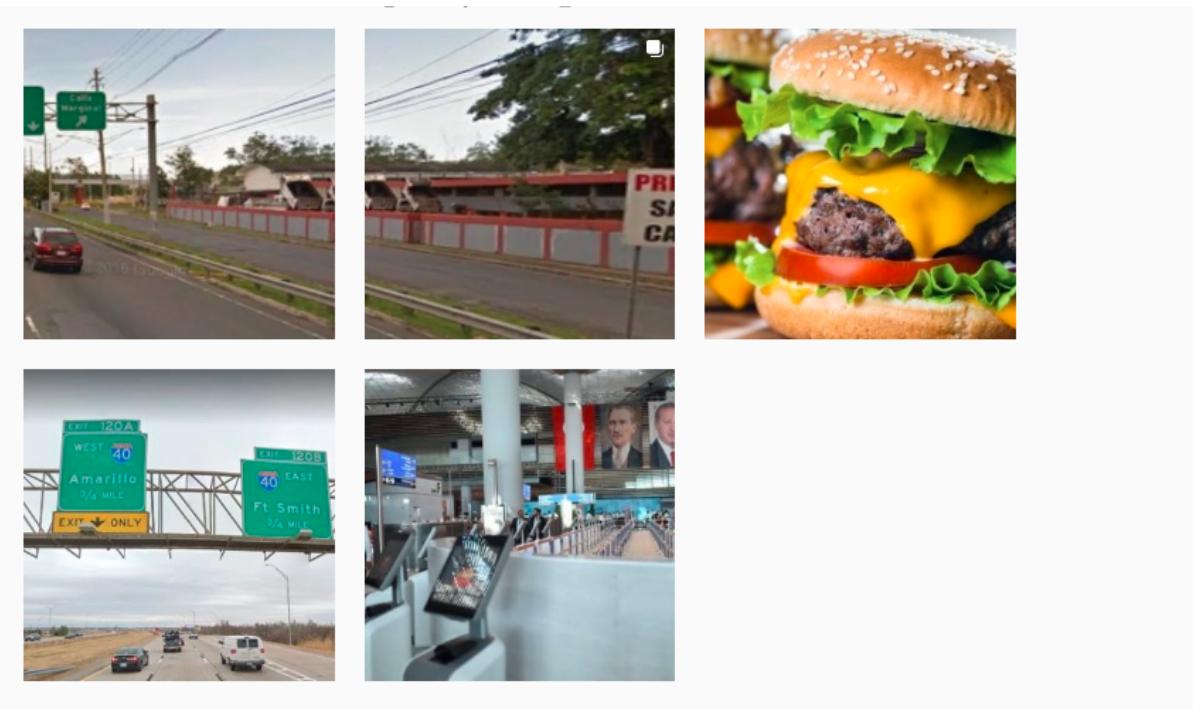
---

## Categoria OSINT-GEOINT

- Stalker on maps - parte 1 - Valendo 800 Pontos

Baixando o arquivo maps.txt, podemos ver uma mensagem e um seguinte usuário "@anna\_serranna"

Utilizando a ferramenta "sherlock", identificamos um usuário no instagram com esse nome "anna\_serranna", e nessa página temos as seguintes fotos;



Na segunda foto que ela postou, podemos ver duas cidades;

Cidade 1 => Amarillo

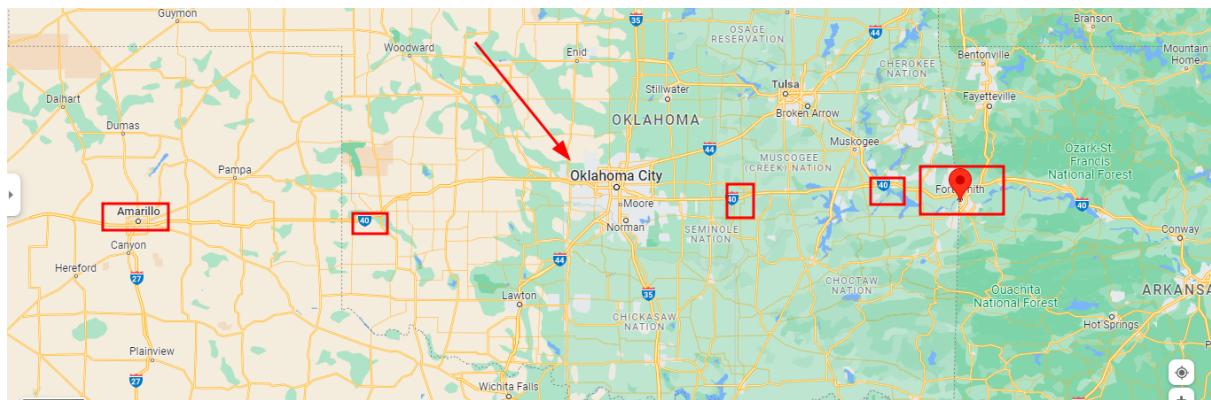
Cidade 2 => Fort smith

E o desafio pede a cidade\_território, tentando colocar Amarillo\_Estados\_Unidos e Fort\_Smith\_Estados\_Unidos, da resposta incorreta, ou seja não é a flag.

O que podemos tentar fazer é pensar de uma seguinte forma;

Amarillo fica do lado de oklahoma (que é um estado americano) e fort smith que também fica do lado de oklahoma, então a resposta certa é "oklahoma". Uma curiosidade é que ambos também tem a mesma rota, ou seja a rota 40.

A rota 40 vai de Amarillo, depois passa por oklahoma e logo em seguida para fort smith e assim vai.



Flag => MCS{oklahoma\_Estados\_Unidos}

---

- Stalker on maps - parte 2 - Valendo 800 Pontos

No mesmo instagram da anna\_serranna, e por meio da última foto que ela conseguiremos prosseguir neste desafio parte 2.



Esta é a última foto, e podemos observar que tem uma duas placas, e nela está escrito;

- San Juan
- Santurce
- Hato Rey

Para concluirmos esse desafio, podemos usar a seguinte lógica, ambos bairros e cidades ficam no porto rico, então sabemos que o território é o porto rico.

E no desafio estava pedindo o nome da cidade. Hato Rey e santurce são bairros que ficam localizados no porto rico, e a única cidade que tem na placa é SanJuan, e essa é a nossa flag! xD

Flag => MCS{SanJuan\_Porto\_Rico}

---

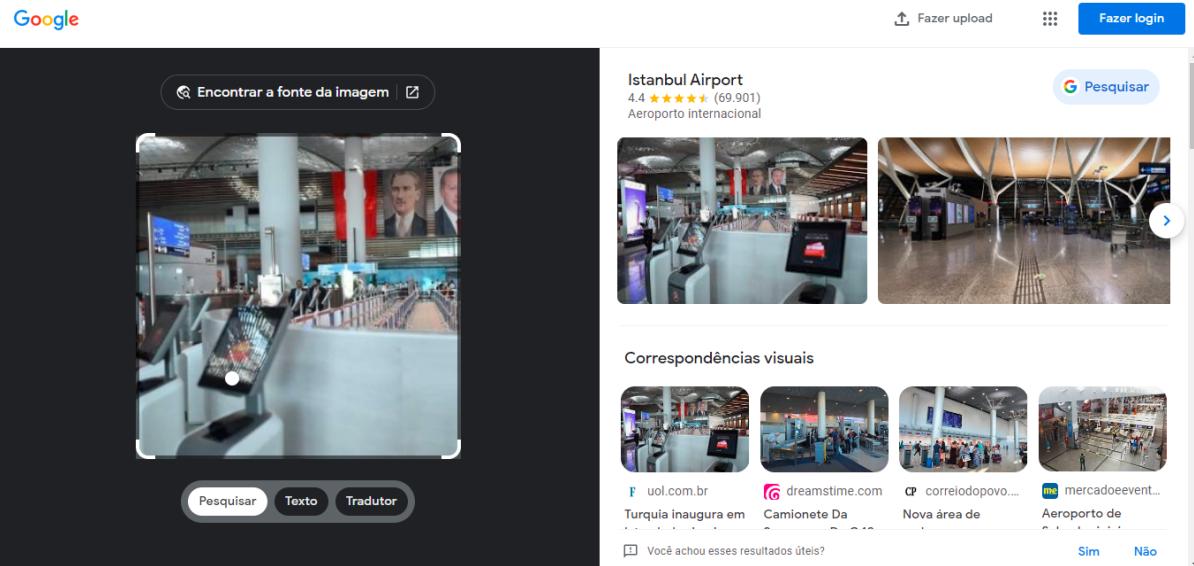
- Onde está o Olly? Não! - Valendo 1700 Pontos

O objetivo deste desafio é "descobrir qual é o aeroporto (sigla) e o País".

Na mesma página do instagram da @anna\_serranna, podemos ver que na primeira foto que ela postou, tem um comentário dela escrito;



Ao lado tem uma imagem. Baixando a imagem e depois usando o google images, podemos perceber que esse aeroporto é de Istambul e que fica localizado na Turquia.



Lembrando que o desafio pede a sigla do aeroporto de Istambul que é (IST), e o país que é a Turquia.

Flag => MCS{IST\_Turquia}

---

## Categoria Forense

- Phished accounts - Valendo 1800 Pontos

Baixando o arquivo "Phished\_accounts.xlsx", podemos abrir ele no excel, e o arquivo está protegido por senha.

D6	A	B	C	D	F
	ID	First Name	Last Name	Email	Recovery
1	1	Lemmie	McWhorter	lmcwhorter0@github.com	Multi-channelled client-server local area network
2	2	Trescha	Digman	tdigman1@so-net.ne.jp	Vision-oriented system-worthy moratorium
3	3	Lacee	Rendbaek	lrendbaek2@godaddy.com	Managed 24 hour projection
4	4	Becka	Jolliffe	bjolliffe3@ocn.ne.jp	Automated mobile success
5	5	Cesya	Shelford	cshelford4@cafepress.com	Cross-group global functionalities
6	6	Ronica	Espinazo	resinapao5@epa.gov	Reacti
7	7	Bella	Simao	bsimao6@discuz.net	Versat
8	8	Clayson	Leffek	cleffek7@lulu.com	Poffit
9	9	Becki	Goodin	bgoodin8@redcross.org	Exclus
10	10	Delmer	Lubman	dlubman9@vistaprint.com	Up-sla
11	11	Chester	Zupa	czupaa@usatoday.com	Integrated discrete attitude
12	12	Nessa	Cheesman	ncheesmanb@mac.com	Optimized global policy
13	13	Kristos	Owlner	kowlnerc@mayoclinic.com	Inverse discrete core
14	14	Tove	Buy	tbuyd@webnode.com	Profound responsive infrastructure
15	15	Cindelyn	Danihel	cdanihel@fda.gov	Networked fresh-thinking core
16	16	Wally	Baggally	wbaggally@dagondesign.com	Phased impactful superstructure
17	17	Gusta	Caldero	gcalderog@prweb.com	Persistent optimizing application
18	18	Tiffani	Attaway	tattawayh@narod.ru	Multi-layered logistical adapter
19	19	Anet	Whild	awhildi@wordpress.com	Reverse-engineered systematic hardware
20					

Bom, para conseguirmos prosseguir neste desafio, precisamos remover as proteções, mas como exatamente poderíamos fazer isso ?

Passo a passo;

OBS : Eu usei o 7-zip também.

- Mover o arquivo "Phished\_accounts.xlsx" para "Phished\_accounts.zip" e depois extrair ela.
- Entrar na pasta: x1 > worksheets.
- Abrir o arquivo "sheet1.xml" no bloco de notas.
- Excluir o "<sheetProtection algorithmName="SHA-512" hashValue="OLnSNkB8YVgdyBACq7rzU3M4bu+3L2UyTJs5nf3tqCJsjqm7GPDvXpl9dwUhq94pPI1J7w9IGN+MpIYGC98zpg==" saltValue="z9KZmrSN/Ch17MUx2GkDAQ==" spinCount="100000" sheet="1" objects="1" scenarios="1"/>" e depois salvar.
- Depois basta ir no diretório "Phished\_accounts", selecione tudo e logo após clique com o botão direito, coloque o mouse em cima do 7-zip e selecione a opção de adicionar ao arquivo compactado.
- Na opção de "Modo de atualização", selecione a opção de "Sincronizar arquivos" e depois é só clicar em ok.
- Abrir o arquivo no excel.

Depois que já estiver com o arquivo aberto no excel, podemos perceber que está faltando uma letra, ou seja a letra E.

A	B	C	D	F	G	H	I	J	K	L
1	ID	First Name	Last Name	Email	Recovery					
2	1	Lennie	McWhorter	lmcwhorter0@github.com	Multi-channelled client-server local area network					
3	2	Trescha	Digman	tdigman1@so-net.ne.jp	Vision-oriented system-worthy moratorium					
4	3	Lacee	Rendbaek	lrendbaek2@godaddy.com	Managed 24 hour projection					
5	4	Becka	Joliffe	bjoliffe3@ocn.ne.jp	Automated mobile success					
6	5	Cesya	Shefford	cshefford4@cafepress.com	Cross-group global functionalities					
7	6	Ronica	Espinazo	respinazo5@epa.gov	Reactive high-level adapter					
8	7	Belia	Simao	bsimao6@discuz.net	Versatile fault-tolerant migration					
9	8	Clayson	Leffek	cleffek7@lulu.com	Profit-focused radical circuit					
10	9	Becki	Goodin	bgoodin8@redcross.org	Exclusive multimedia capability					
11	10	Delmer	Lubman	dlubman9@vistaprint.com	Up-sized regional capability					
12	11	Chester	Zupa	czupaa@usatoday.com	Integrated discrete attitude					
13	12	Nessa	Cheesman	ncheesmanb@mac.com	Optimized global policy					
14	13	Kristos	Owliner	kowlnerc@mayo clinic.com	Inverse discrete core					
15	14	Tove	Buy	tbuyd@webnode.com	Profound responsive infrastructure					
16	15	Cindelyn	Danielh	cdanielhele@fda.gov	Networked fresh-thinking core					
17	16	Wally	Baggally	wbbaggallyf@dagondesign.com	Phased impactful superstructure					
18	17	Gusta	Caldero	gcalderog@prweb.com	Persistent optimizing application					
19	18	Tiffani	Attaway	tattawayh@narod.ru	Mult-layered logistical adapter					
20	19	Anet	Whild	awhildi@wordpress.com	Reverse-engineered systematic hardware					
21	20	Lief	Fylan	fylianj@ft.com	Business-focused encompassing migration					
22	21	Siana	Checo	sc hecok@gizmodo.com	Synergized demand-driven methodology					
23	22	Clari	Esonwatu	esonwatu@daovietart.com	Switchable methodical time frame					

Dando um zoom e logo em seguida puxando a coluna D para o lado, conseguimos ver a coluna E que estava faltando

Nessa coluna "E", tem as senhas dos endereços de emails. Podemos ver que lá embaixo, tem um endereço de email chamado "admin@nosferatu.com" e ao lado a flag.

Phished\_accounts **XLSX**

Arquivo Editar Ver Inserir Formatar Dados Ferramentas Ajuda A última edição foi há alguns segundos

E73 | eg6fRJENAJA6

A	B	C	D	E	F
56	55	Donovan	Boutellier	dboutellier1@blinklist.com	nM4NVQIV
57	56	Angeline	MacGhee	amacghee1@xrea.com	o4DEQ1
58	57	Herby	Napoleon	hnapoleon1k@google.com	gBuaysu
59	58	Baron	Chidzoy	bchidzoy1l@yellowbook.com	6gDemoBW2U
60	59	Kane	Bellino	kbellino1m@domainmarket.com	fA9tdarZb8E
61	60	Thane	Crossby	tcrossby1n@mlinfo.com	L3tEvvy0T
62	61	Ruddie	Moehle	edminne1n@nodevatu.com	MCS{Own3d_Fl4g_F0rf4n_F0r3ns3}
63	62	Inna	Lemarie	ilemarie1p@ochre.jp	7Oxb95cum11
64	63	Colas	Kyndred	ckyndred1q@webnode.com	lyduepyaj
65	64	Binky	Rozzenweig	brozzenweig1r@livejournal.com	1ICxT7HA
66	65	Petunia	Dybll	pdybll1s@walmart.com	BcuCWgkW
67	66	Gino	Meugens	gmeugens1t@google.com.hk	6eUsOOFuGh
68	67	Fredek	Lorraway	florraway1u@symantec.com	HR45dLUjy/b
69	68	Corbie	Pincney	cpinckney1v@github.io	21Oj9lok
70	69	Gracie	Canham	gcanham1w@tinyPic.com	zNRSSw2WY
71	70	Karie	Grinvalds	kgrinvalds1x@w3.org	SgnZxPQ
72	71	Katalin	Maynor	kmaynor1y@mail.com	3ptumr3c1HO
73	72	Trev	Van der Kruis	tvanderkruis1z@deviantart.com	egg6fRJENAJA6
74	73	Ermentrude	Nockolds	enockolds20@multiply.com	MM1lpwv3s
75	74	Granville	Jury	giury21@geocities.jp	5wHjyQwgSQW
76	75	Alicia	Wadly	awadly22@berkeley.edu	AqGZ3Cv6HRv
77	76	Ekaterina	Canec	eranec23@rauenthrone.com	wD9zvthGt

+ phished\_accounts ▾ Explorar

Flag => MCS{Own3d\_Fl4g\_F0rf4n\_F0r3ns3}

## Categoria Análise de Artefato

Essa categoria consiste em fazer uma análise de um pdf contendo link malicioso (phishing).

- "Me manda no e-mail" - Valendo 1250 Pontos

Baixando o pdf, a primeira coisa a se fazer é colocar ela no virustotal.

<https://www.virustotal.com/gui/file/6a4440a995dd031554d6f3ff71f196896287c20f5c6e664b85876b9adc6058c4>

O desafio pede para identificar o SHA-256 desse artefato.

Depois que colocarmos o pdf no virustotal, podemos ir na opção de details que lá mostra o md5, sha-1, sha-256, etc desse artefato.

The screenshot shows a VirusShare analysis page for a PDF file. The file hash is 6a4440a995dd031554d6f3ff71f196896287c20f5c6e664b85876b9adc6058c4. A red box highlights the SHA-256 value: 6a4440a995dd031554d6f3ff71f196896287c20f5c6e664b85876b9adc6058c4. The page includes tabs for DETECTION, DETAILS (selected), RELATIONS, BEHAVIOR, and COMMUNITY. Under DETAILS, there's a 'Basic Properties' section with various file metadata. A 'History' section shows the creation time as 2022-02-03 07:07:18 UTC.

Flag => MCS{6a4440a995dd031554d6f3ff71f196896287c20f5c6e664b85876b9adc6058c4}

- Tem técnica, mas não tem tática... - Valendo 1300 Pontos

Nesta challenge precisamos ver qual é a técnica do Mitre Att&CK que esse artefato se encaixa.

Bom, pesquisando por "email phishing mitre att&ck" no google, conseguimos encontrar uma página falando sobre spear phishing do próprio mitre att&ck, e pronto este artefato se encaixa na técnica de spear phishing.

Link -> <https://attack.mitre.org/techniques/T1566/002/>

Flag => MCS{phishing\_spear}

- Não sei, Berg, tá estranho... - Valendo 2000 Pontos

"Algum security vendor sinalizou esse artefato como malicioso?"

O ESET-NOD32 classificou este pdf como "PDF/Phishing.A.Gen".

The screenshot shows a VirusShare analysis page for a PDF file. The file was flagged as malicious by one security vendor (ESET-NOD32) and no sandboxes flagged it as malicious. The file details are as follows:

- File Hash:** 6a4440a995dd031554d6f3ff71f196896287c20f5c6e664b85876b9adc605c4
- Size:** 48.96 KB
- Upload Date:** 2022-11-13 22:37:08 UTC
- Category:** 22 hours ago
- File Type:** Notificacao\_-\_CAIXA\_ECONOMICA\_FEDERAL.pdf
- Tags:** checks-user-input, detect-debug-environment, direct-cpu-clock-access, long-sleeps, pdf, runtime-modules
- Community Score:** 1 / 63

The navigation tabs are DETECTION (selected), DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY.

**Security Vendors' Analysis:**

Vendor	Result	Notes
ESET-NOD32	PDF/Phishing A Gen	(Indicated by a red box)
Ad-Aware	Undetected	
ALYac	Undetected	
Arcabit	Undetected	
AVG	Undetected	
Baidu	Undetected	
Acronis (Static ML)	Undetected	
AhnLab-V3	Undetected	
Antiy-AVL	Undetected	
Avast	Undetected	
Avira (no cloud)	Undetected	
BitDefender	Undetected	

Flag => MCS{ESET-NOD32}

---

## Categoria Web

- Não odeio o JavaScript! - Valendo 2000 Pontos

<http://198.211.107.250:1337/>

Um bom fuzzing/recon sempre é bom né! haha, vamos lá!

Usando o ffuf para fazer fuzzing de diretório, encontramos um diretório chamado /admin.

ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt -u http://198.211.107.250:1337/FUZZ --fs 169

```

→ ~ cat fuzz.txt|grep admin
admin [Status: 301, Size: 162, Words: 5, Lines: 8, Duration: 121ms]
→ ~

```

Indo no diretório /admin, podemos perceber que não temos acesso a esta página.

## 403 Forbidden

nginx

E então vamos fazer mais um fuzzing para descobrir os arquivos, diretórios, e etc do /admin.

- ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-large-files-lowercase.txt -u http://198.211.107.250:1337/admin/FUZZ --fs 169



```
~ ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-large-files-lowercase.txt -u http://198.211.107.250:1337/admin/FUZZ --fs 169
[...]
v1.5.0 Kali Exclusive <3

:: Method      : GET
:: URL        : http://198.211.107.250:1337/admin/FUZZ
:: Wordlist    : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-large-files-lowercase.txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 200,204,301,302,307,401,403,405,500
:: Filter        : Response size: 169

[...]
[Status: 403, Size: 146, Words: 3, Lines: 8, Duration: 128ms]
admin.js [Status: 200, Size: 1474, Words: 186, Lines: 52, Duration: 127ms]
:: Progress: [20247/35325] :: Job [1/1] :: 329 req/sec :: Duration: [0:01:01] :: Errors: 0 ::
```

Encontramos um arquivo chamado admin.js no /admin, e então acessando ele podemos encontrar a flag.

```
const config = {
  /* Set default values assuming NODE_ENV === production */
  port: getEnvVarOrDefault("PORT", "PORT"),
  jwtSecret: getEnvVarOrDefault("CA_JWT_SECRET", "admin"),
  trustProxy: getEnvVarOrDefault("TRUST_PROXY", "MCS{3num3r4t10n_F0rfun_4nd_Pr0f1t}"),
  dbConnectionString: getEnvVarOrExitError("CA_DB_CONNECTION_URL"),
  oidc: {
    compare: getEnvVarOrDefault("OIDC_COMPARE", "sub"),
    issuerUrl: getEnvVarOrExitError("OIDC_ISSUER_URL"),
    userInfoPath: getEnvVarOrDefault("OIDC_USERINFO_PATH", "/userinfo"),
  },
  publicUrl: getEnvVarOrExitError("PUBLIC_URL"),
  templateUrl: getEnvVarOrDefault(
    "TEMPLATE_URL",
    `local://${path.join(__dirname, "..", "templates")}`
  ),
};

export default config;
```

Flag => MCS{3num3r4t10n\_F0rfun\_4nd\_Pr0f1t}

- Nosferatu - Valendo 3000 Pontos

<http://198.211.107.250:1337/>

O nome do título da página é "juggling nosferatu", então suponhamos que temos que explorar um type juggling.

Na index da página podemos observar que tem um botão escrito "Secret Nosferatu", mas antes de começar a testar, vamos abrir o Burp Suite e começar a interceptar as requisições.

Clicando no secret nosferatu, podemos observar um type e ao lado secrets. Vamos ver o que acontece se alterarmos o secrets para nosferatu.

\u2022&lt;/span&gt;&lt;span class=\\"highlight-letter-ss17\\"&gt;\u2022&lt;/span&gt; pleno &lt;span class=\\"highlight-number\\"&gt;\u2022&lt;/span&gt;&lt;span class=\\"highlight-letter-ss17\\"&gt;\u2022&lt;/span&gt;ano da copa do mundo &lt;span class=\\"highlight-word\\"&gt;letter-ss17&lt;/span&gt;&lt;/p&gt;', and 'fact\_type': 'nosferatu'. The second object has an 'id': 2, 'fact': '&lt;span class=\\"pumpkin\\"&gt;&gt;pumpkin:&lt;/span&gt;&lt;span class=\\"pumpkin\\"&gt;&gt;pumpkin:&lt;/span&gt; PLENO 2022&lt;/span&gt;', and 'fact\_type': 'nosferatu'. The third object has an 'id': 3, 'fact': '&lt;span class=\\"pumpkin\\"&gt;&gt;pumpkin:&lt;/span&gt;&lt;span class=\\"pumpkin\\"&gt;&gt;pumpkin:&lt;/span&gt;ano da copa do mundo&lt;/p&gt;', and 'fact\_type': 'nosferatu'. The Burp Suite interface remains largely the same, with tabs for Dashboard, Target, Proxy, Intruder, Repeater, Window, Help, and various sub-options like Project options, User options, Learn, Intercept, HTTP history, WebSockets history, and Options. The Inspector panel on the right shows Response Headers."/&gt;

E ele nos traz esse json.

O que podemos tentar fazer é mudar o seguintes valores;

- "type":"secrets" para "type":true

The screenshot shows a browser window for 'Juggling Nosferatu' and the Burp Suite proxy interface. The browser displays the challenge page with a 'Secret Nosferatu' button. The Burp Suite interface shows the intercepted request and response. The response body is a JSON object:

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 Nov 2022 22:45:05 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 87

{
    "facts": [
        {
            "id": 1,
            "fact": "MCS{0wn3d_fl4g_f0r_typ3_Juggl1ng}",
            "fact_type": "secrets"
        }
    ]
}
```

E pronto, temos a flag do último desafio!

- Mas o que de fato aconteceu ?

Bom, se == for usado em códigos PHP versões inferiores ao php8, haverá casos inesperados em que a comparação não se comportará conforme o esperado. Isso ocorre porque "==" apenas compara valores transformados para o mesmo tipo, se você também deseja comparar se o tipo dos dados comparados é o mesmo, você precisa usar ===. Por isso, quando especificamos o type para true, o php não está verificando o tipo, isso gera a vulnerabilidade Type Juggling.

Referência -> <https://owasp.org/www-pdf-archive/PHPMagicTricks-TypeJuggling.pdf>

Flag => MCS{0wn3d\_fl4g\_f0r\_typ3\_Juggl1ng}

---

*Writeup Creator @matheuz\_security*