A close up of a dragon's face

Description automatically generated

**Finding Name: API Disclosure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Team** | **Role** | **Project** | **Quality Assurance** | **Is this a re-tested Finding?** |
| DEV PATEL | PT | Team Member | EchoNet |  |  |
| Jack Perry | PT | Team Member | EchoNet |  |  |

|  |
| --- |
| **Was this Finding Successful?** |
| YES |

**Finding Description**

An API disclosure vulnerability was identified in the application, where sensitive information about the API endpoints and internal implementation details is exposed. This vulnerability occurs when the application fails to adequately restrict access to its API documentation.

Access to exposed API through port 9000 and which later leads to /docs page which is FasAPI page with all schemes of the application. So, we can Add, modify or delete the data in application through any third-party API tools.

**Risk Rating**   
Impact: Severe    
Likelihood: Certain

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact values** | | | | |
| **Very Minor** | **Minor** | **Significant** | **Major** | **Severe** |
| Risk that holds little to no impact. Will not cause damage and regular activity can continue. | Risk that holds minor form of impact, but not significant enough to be of threat. Can cause some damage but not enough to impede regular activity. | Risk that holds enough impact to be somewhat of a threat. Will cause damage that can impede regular activity but will be able to run normally. | Risk that holds major impact to be of threat. Will cause damage that will impede regular activity and will not be able to run normally. | Risk that holds severe impact and is a threat. Will cause critical damage that can cease activity to be run. |

Shape

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Likelihood** | | | | |
| **Rare** | **Unlikely** | **Moderate** | **High** | **Certain** |
| Event may occur and/or if it did, it happens in specific circumstances. | Event could occur occasionally and/or could happen (at some point) | Event may occur and/or happens. | Event occurs at times and/or probably happens a lot. | Event is occurring now and/or happens frequently. |

**Business Impact**

Through direct and unauthorised access to the API endpoints, the attacker can view, modify and delete user data. Attackers can then have a severe impact on business operations by tampering with data.

Beyond executing requests, attackers can further gather crucial information about the applications infrastructure, configurations and requirements to assist in further attacks.

When data (such animal movements and microphone positions) are accessed and altered without restriction, the Project ECHO application may include inconsistent or incorrect details.

Exposed API endpoints may lead to attackers abusing the API by sending out a lot of queries (DDoS)

**Affected Assets**

API sensitive Data

**Evidence**

Provide a step-by-step guide on how to reproduce the vulnerability with screenshots

**Step 1: Port Scanning**

**Using the Nmap to scan all open port in Application**

A screen shot of a computer

Description automatically generated

After previously scanning the host to detect all open ports, another nmap scan using ‘–A’ to perform an aggressive scan and run default scripts on port 9000 reveals a successful HTTP request:

A screenshot of a computer program

Description automatically generated

**Step 2: Navigate to Exposed API URL (127.0.0.1:9000)**

A screenshot of a computer

Description automatically generated

**Step 3: As we can see the message suggest us to navigate to /docs for more details and also, we get the same detail while scanning the port 9000 using Nmap.**

A screenshot of a computer

Description automatically generated

**Step 4: Navigate to /openapi.json**

A screenshot of a computer program

Description automatically generated

* Here we got all API schemes.
* With this scheme we can use any Api tool to modify in the application like (insomnia or postman)

**Remediation Advice**

Implement strong authentication mechanisms, such as OAuth 2.0. Implement authorization properly by defining an access control policy.

Limit access to make sure the "/docs" endpoint and "/openapi.json" schema are not available to the public.

Use rate restriction to limit the quantity of requests received in a certain amount of time from a single source. Reduce server demand by utilizing caching techniques.

**References**

Nmap

**Contact Details**

DEV PATEL: - [s222622553@deakin.edu.au](mailto:s222622553@deakin.edu.au)

Jack Perry – [s217298346@deakin.edu.au](mailto:s217298346@deakin.edu.au)

**Pentest Leader Feedback.**

The lead will provide feedback to enact on