Service Misconfigurations

Misconfigurations usually happen when a system administrator, technical support, or developer does not correctly configure the security framework of an application, website, desktop, or server leading to dangerous open pathways for unauthorized users. Let's explore some of the most typical misconfigurations of common services.

Authentication

In previous years (though we still see this sometimes during assessments), it was widespread for services to include default credentials (username and password). This presents a security issue because many administrators leave the default credentials unchanged. Nowadays, most software asks users to set up credentials upon installation, which is better than default credentials. However, keep in mind that we will still find vendors using default credentials, especially on older applications.

Even when the service does not have a set of default credentials, an administrator may use weak passwords or no passwords when setting up services with the idea that they will change the password once the service is set up and running.

As administrators, we need to define password policies that apply to software tested or installed in our environment. Administrators should be required to comply with a minimum password complexity to avoid user and passwords combinations such as:

```
admin:admin
admin:password
admin:<blank>
root:12345678
administrator:Password
```

Once we grab the service banner, the next step should be to identify possible default credentials. If there are no default credentials, we can try the weak username and password combinations listed above.

Anonymous Authentication

Another misconfiguration that can exist in common services is anonymous authentication. The service can be configured to allow anonymous authentication, allowing anyone with network connectivity to the service without being prompted for authentication.

Misconfigured Access Rights

Let's imagine we retrieved credentials for a user whose role is to upload files to the FTP server but was given the right to read every FTP document. The possibility is endless, depending on what is within the FTP Server. We may find files with configuration information for other services, plain text credentials, usernames, proprietary information, and Personally identifiable information (PII).

Misconfigured access rights are when user accounts have incorrect permissions. The bigger problem could be giving people lower down the chain of command access to private information that only managers or administrators should have.

Administrators need to plan their access rights strategy, and there are some alternatives such as Role-based access control (RBAC), Access control lists (ACL). If we want more details pros and cons of each method, we can read Choosing the best access control strategy by Warren Parad from Authress.

Unnecessary Defaults

The initial configuration of devices and software may include but is not limited to settings, features, files, and credentials. Those default values are usually aimed at usability rather than security. Leaving it default is not a good security practice for a production environment. Unnecessary defaults are those settings we need to change to secure a system by reducing its attack surface.

We might as well deliver up our company's personal information on a silver platter if we take the easy road and accept the default settings while setting up software or a device for the first time. In reality, attackers may obtain access credentials for specific equipment or abuse a weak setting by conducting a short internet search.

Security Misconfiguration are part of the OWASP Top 10 list. Let's take a look at those related to default values:

- Unnecessary features are enabled or installed (e.g., unnecessary ports, services, pages, accounts, or privileges).
- Default accounts and their passwords are still enabled and unchanged.
- Error handling reveals stack traces or other overly informative error messages to users.
- For upgraded systems, the latest security features are disabled or not configured securely.

Preventing Misconfiguration

Once we have figured out our environment, the most straightforward strategy to control risk is to lock down the most critical infrastructure and only allow desired behavior. Any communication that is not required by the program should be disabled. This may include things like:

- Admin interfaces should be disabled.
- Debugging is turned off.
- Disable the use of default usernames and passwords.
- Set up the server to prevent unauthorized access, directory listing, and other issues.
- Run scans and audits regularly to help discover future misconfigurations or missing fixes.

The OWASP Top 10 provides a section on how to secure the installation processes:

- A repeatable hardening process makes it fast and easy to deploy another environment that is appropriately locked down. Development, QA, and production environments should all be configured identically, with different credentials used in each environment. In addition, this process should be automated to minimize the effort required to set up a new secure environment.
- A minimal platform without unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.
- A task to review and update the configurations appropriate to all security notes, updates, and patches as part of the patch management process (see A06:2021-Vulnerable and Outdated Components). Review cloud storage permissions (e.g., S3 bucket permissions).
- A segmented application architecture provides effective and secure separation between components or tenants, with segmentation, containerization, or cloud security groups (ACLs).
- Sending security directives to clients, e.g., security headers.
- An automated process to verify the effectiveness of the configurations and settings in all environments.