[EN] Vulnerability Report
CLOUDRON 7.6.3

In the context of this report presenting two distinct vulnerabilities, I would like to be mentioned as the source/author in the corresponding CVE entries and in the patch notes. This would be a valuable recognition of my contribution and greatly assist my professional ambitions.

Critère	Description
Impact on Confidentiality	GDPR Risk Depending on Use of Solution
Impact on Integrity	Potential Data Alteration
Impact on Availability	Possibility of Service Interruption
Exploitation Complexity	Proof of Concept (Private)
Required Privileges	Requires CLI Access on the Machine
User Interaction	No User Interaction Required
Context of the Solution Version	Latest Version as of Report Date (7.6.3)

[EN] Context

While setting up the solution, I closely examined the components installed by the installation script to ensure the security of the data to be stored. This analysis allowed me to detect certain security flaws. The combined exploitation of these vulnerabilities can lead to the takeover of the "owner" account and the obtaining of root rights on the server hosting the solution.

[EN] Prerequisites

- One of the fundamental criteria is access to a user account on the system hosting the solution. For the purpose of our study, we will refer to the 'www-data' account, a standard account without elevated privileges.

- It is also important to note that our analysis is based on the most recent version of the solution.

[EN] Deployment

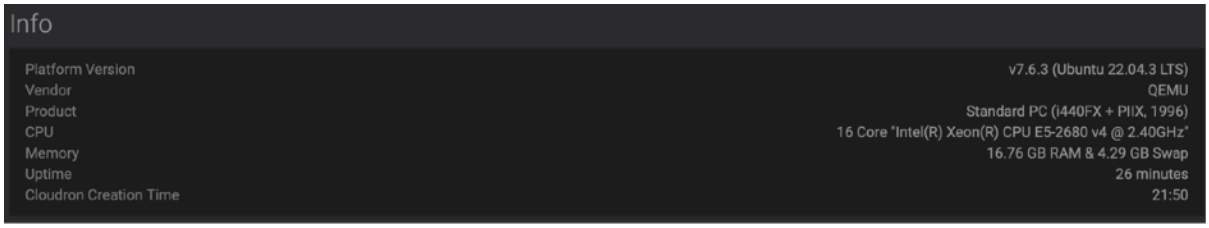
Domain setup

Domain	cloudron-poc.lan
DNS Provider	No-op (only for development)
Zone Name (optional)	cloudron-poc.lan
Certificate provider	Self-Signed
IP Configuration	Network Interface
Interface Name	ens18

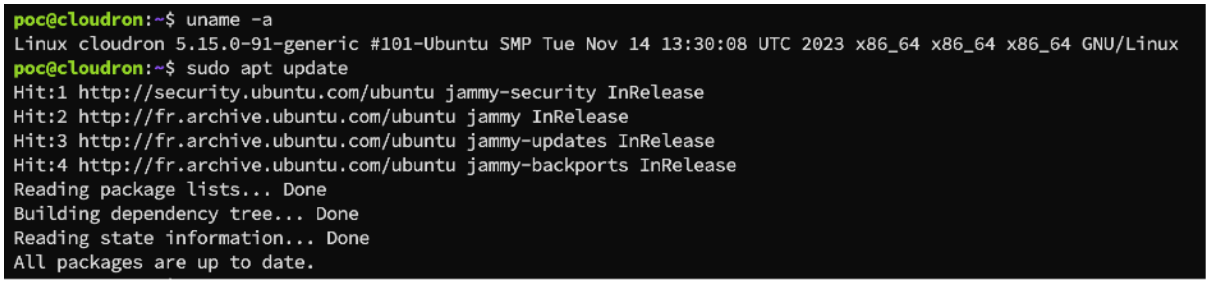
Set up Admin Account

Full Name	Proof of Concept
Email	poc@cloudron-poc.lan
Username	myuser
Password	rootroot

Information from Dashbord



Information from CLI



[EN] STEAL ‘OWNER’ ACCOUNT

Exploitation

To exploit this vulnerability, CLI access on the server hosting the solution is necessary.

The vulnerability exploits a misconfiguration in the database, present in the version 7.6.3 that I deployed. It is easily rectifiable. My script uses this vulnerability to perform actions in the database, allowing the 'owner' account to be reset. This script also disables two-factor authentication and reactivates the account if it was disabled. Once access to the 'owner' account is obtained, it is possible to take full control of the solution.

(As you are more familiar with the solution than I am, it is not necessary to detail the possibilities available once in possession of the owner account.).

```
www-data@cloudron:/tmp$ systemctl status box
● box.service - Cloudron Admin
     Loaded: loaded (/etc/systemd/system/box.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-01-05 20:32:30 UTC; 1h 37min ago
   Main PID: 1056 (node)
      Tasks: 11 (limit: 19048)
     Memory: 113.8M (max: 400.0M available: 286.1M)
        CPU: 11.939s
     CGroup: /system.slice/box.service
             └─1056 node /home/yellowtent/box/box.js
www-data@cloudron:/tmp$ ./CVE-2024-XXXXX.sh
ID de l'utilisateur : uid-f5dba46e-d35d-4230-93f8-49299acb8a87
Nom d'utilisateur : myuser
{
  "accessToken": "137197ed94b63cdc8ed77d86e0ed40bc4e552cf1bba0791ff6af943c1005bdaa"
}
credential myuser/strongpassword
```

Continuing the Analysis

It is likely feasible to exploit access to the database to insert data perceived as legitimate by box.js. This data could then be used in a shell.promises.exec() or shell.promises.sudo() command, thus enabling remote access to the 'yellowtent' account.

Proof of Concept

Please share with me your solution for secure file sharing so that I can send you the file securely. This file includes all the detailed steps of the exploitation.

Remediation

Corriger la mauvaise configuration liée à la base de donnée.

[EN] PRIVILEGE ESCALATION

Exploitation

In order to exploit this vulnerability, you must first become 'yellowtent'.

The vulnerability exploits account permissions to perform actions as sudo, thus obtaining root user rights.

```
yellowtent@cloudron:/tmp$ systemctl status box
● box.service - Cloudron Admin
     Loaded: loaded (/etc/systemd/system/box.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-01-05 20:32:30 UTC; 1h 13min ago
   Main PID: 1056 (node)
      Tasks: 11 (limit: 19048)
     Memory: 111.1M (max: 400.0M available: 288.8M)
        CPU: 10.670s
     CGroup: /system.slice/box.service
             └─1056 node /home/yellowtent/box/box.js

Jan 05 20:45:13 cloudron sudo[1579]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=808)
Jan 05 20:45:13 cloudron systemd[1]: Reloading Cloudron Admin...
Jan 05 20:45:13 cloudron sudo[1579]: pam_unix(sudo:session): session closed for user root
Jan 05 20:45:14 cloudron sudo[1594]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=808)
Jan 05 20:45:14 cloudron sudo[1594]: pam_unix(sudo:session): session closed for user root
Jan 05 20:45:18 cloudron systemd[1]: Reloaded Cloudron Admin.
Jan 05 20:51:57 cloudron sudo[3968]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=808)
Jan 05 20:51:57 cloudron sudo[3968]: pam_unix(sudo:session): session closed for user root
Jan 05 20:52:23 cloudron sudo[3989]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=808)
Jan 05 20:52:23 cloudron sudo[3989]: pam_unix(sudo:session): session closed for user root
yellowtent@cloudron:/tmp$ whoami
yellowtent
yellowtent@cloudron:/tmp$ ./priv.sh
backup file
injection ...
give shell root...
root@cloudron:/tmp# whoami
root
root@cloudron:/tmp#
```

Proof of Concept

Please share with me your solution for secure file sharing so that I can send you the file securely. This file includes all the detailed steps of the exploitation.

Remediation

Check and restrict the permissions of the 'yellowtent' account to the bare minimum to prevent the account from using its current rights to gain administrative access to the source machine.

[FR] Rapport de vulnérabilité
CLOUDRON 7.6.3

Dans le cadre de ce rapport exposant deux vulnérabilités distinctes, j'aimerais être mentionné comme source/auteur dans les entrées CVE correspondantes et dans les notes de patch. Cela constituerait une reconnaissance valorisante de ma contribution et serait d'une grande aide pour mes ambitions professionnelles

Critère	Description
Impact sur la Confidentialité	Risque RGPD selon l'utilisation de la solution
Impact sur l'Intégrité	Altération des données possible
Impact sur la Disponibilité	Possibilité d'interrompe les services
Complexité de l'Exploitation	Proof of Concept (privé)
Privilèges Nécessaires	Nécessite un CLI sur la machine
Interaction de l'Utilisateur	Aucune interaction de l'utilisateur nécessaire
Contexte de la Version de la Solution	Dernière version à date du rapport (7.6.3)

[FR] Contexte

Durant la mise en place de la solution, j'ai attentivement examiné les composants installés par le script d'installation pour assurer la sécurité des données à y stocker. Cette analyse m'a permis de détecter certaines failles de sécurité. L'exploitation combinée de ces vulnérabilités permet d'aboutir à la prise de contrôle du compte "owner" ainsi qu'à l'obtention des droits root sur le serveur hébergeant la solution.

[FR] Pré-requis

- L'un des critères fondamentaux est l'accès à un compte utilisateur sur le système hébergeant la solution. Dans le cadre de notre étude, nous

ferons référence au compte 'www-data', un compte standard dépourvu de privilèges élevés.

- Il est également important de noter que notre analyse se base sur la version la plus récente de la solution.

[FR] Déploiement

Domain setup

Domain	cloudron-poc.lan
DNS Provider	No-op (only for development)
Zone Name (optional)	cloudron-poc.lan
Certificate provider	Self-Signed
IP Configuration	Network Interface
Interface Name	ens18

Set up Admin Account

Full Name	Proof of Concept
Email	poc@cloudron-poc.lan
Username	myuser
Password	rootroot

Information from Dashbord

Information from CLI

```
poc@cloudron:~$ uname -a
Linux cloudron 5.15.0-91-generic #101-Ubuntu SMP Tue Nov 14 13:30:08 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
poc@cloudron:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://fr.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

[FR] STEAL ‘OWNER’ ACCOUNT

Exploitation

Afin d’exploitation cette vulnérabilité il est nécessaire d’avoir un CLI sur le serveur qui héberge la solution.

La vulnérabilité tire parti d'une mauvaise configuration dans la base de données, présente dans la version 7.6.3 que j'ai mise en place. Elle est aisément rectifiable. Mon script exploite cette faille pour effectuer des actions dans la base de données, permettant de réinitialiser le compte 'owner'. Ce script désactive également l'authentification à deux facteurs et réactive le compte si celui-ci était désactivé. Une fois accès au compte 'owner' obtenu, il est possible de prendre le contrôle total de la solution.

(Étant donné que vous connaissez la solution mieux que moi, il n'est pas nécessaire de détailler les possibilités offertes une fois en possession du compte owner.).

```
www-data@cloudron:/tmp$ systemctl status box
● box.service - Cloudron Admin
     Loaded: loaded (/etc/systemd/system/box.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-01-05 20:32:30 UTC; 1h 37min ago
   Main PID: 1056 (node)
      Tasks: 11 (limit: 19048)
     Memory: 113.8M (max: 400.0M available: 286.1M)
        CPU: 11.939s
     CGroup: /system.slice/box.service
             └─1056 node /home/yellowtent/box/box.js
www-data@cloudron:/tmp$ ./CVE-2024-XXXXX.sh
ID de l'utilisateur : uid-f5dba46e-d35d-4230-93f8-49299acb8a87
Nom d'utilisateur : myuser
{
  "accessToken": "137197ed94b63cdc8ed77d86e0ed40bc4e552cf1bba0791ff6af943c1005bdaa"
}
credential myuser/strongpassword
```

Poursuivant l’analyse

Il est probablement faisable d'exploiter l'accès à la base de données pour y insérer une donnée perçue comme légitime par box.js. Cette donnée pourrait ensuite être utilisée dans une commande `shell.promises.exec()` ou `shell.promises.sudo()`, permettant ainsi d'obtenir un accès à distance au compte 'yellowtent'.

Proof of Concept

Merci de me communiquer votre solution pour le partage sécurisé de fichiers, afin que je puisse vous envoyer le fichier de manière sécurisée. Ce fichier comprend toutes les étapes détaillées de l'exploitation.

Remediation

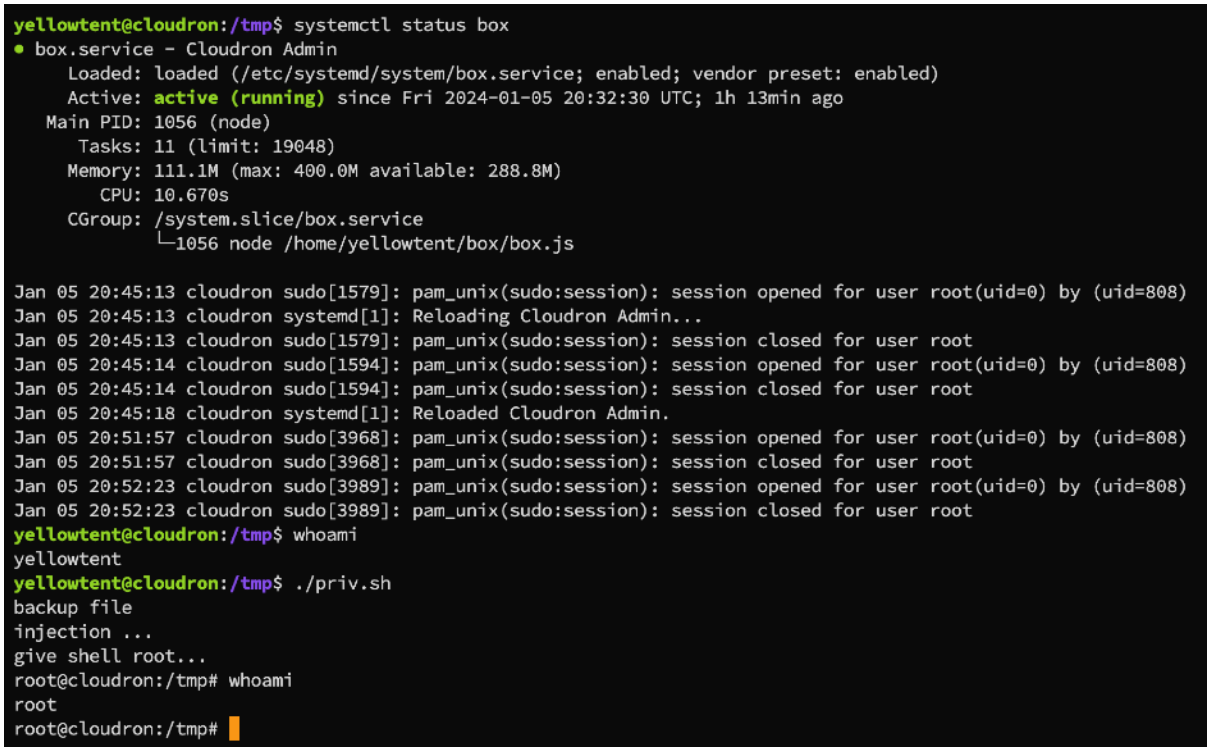
Corriger la mauvaise configuration liée à la base de donnée.

[FR] PRIVILEGE ESCALATION

Exploitation

Afin d'exploitation cette vulnérabilité il est nécessaire d'avoir au préalable réussi à devenir 'yellowtent'

La vulnérabilité exploite les permissions du compte pour effectuer des actions en tant que sudo, ce qui permet d'obtenir les droits d'utilisateur root.



Proof of Concept

Merci de me communiquer votre solution pour le partage sécurisé de fichiers, afin que je puisse vous envoyer le fichier de manière sécurisée. Ce fichier comprend toutes les étapes détaillées de l'exploitation.

Remediation

Contrôler l'ensemble des droits du compte yellowtent et les limiter au strict minimum afin d'éviter le compte à utiliser les droits actuel pour lui permettre de devenir administrateur de la machine source.