Mitigating CVE-2022-22948: VMware vCenter Information Disclosure - Pentera

By Yuval Lazar

Archived: 2026-04-05 23:05:11 UTC

New zero-day vulnerability joins a chain of recently discovered vulnerabilities capable of operating an end-to-end attack on ESXi. Organizations should evaluate risk and apply vCenter client patches immediately.

Executive Summary

Pentera Labs' Senior Security Researcher, Yuval Lazar, discovered an Information Disclosure vulnerability impacting more than 500,000 appliances running default vCenter Server deployments. This finding is critical given its potential global impact. According to VMware – more than 80 percent of virtualized workloads are running on VMware technology, including 100 percent of Fortune 500 and Fortune Global 100 companies. The ease and convenience that vCenter offers for managing virtualized hosts in enterprise environments provides cybercriminals with centralized access and the potential to inflict widespread damage on organizations. In the full attack vector, threat actors can completely take over an organization's ESXi's deployed in a hybrid infrastructure and virtual machines hosted and managed by the hypervisor from just endpoint access to a host with a vCenter client. VMware has issued a patch for the vulnerability that can be found here.

Background

Installed in hundreds of thousands of organizations worldwide and used to manage some of their most critical asset and core systems, VMware vCenter Server is a high-priority target for cybercriminals. Once compromised, the ease and convenience that vCenter offers for managing virtualized hosts in enterprise environments will play into the adversary's hands, providing centralized access and widespread damage. Spurred by previously reported vulnerabilities, increasing demand from our customers, and threats observed in the wild, most notably recent reports of a python ransomware strain targeting ESXi, Pentera Labs has doubled down on exploring new vCenter attack vectors. The result of these explorations was the discovery of a new Information Disclosure vulnerability. The findings were proactively reported to VMware and later released under CVE-2022-22948.

Complete Attack Vector

The vulnerability described in this paper is part of a critical kill chain that leads to an ESXi takeover. In the full attack vector, which is a subject for a future article, we can see that initial endpoint access to a host with a vCenter client lends itself to a complete takeover of the organization's ESXis. The devil and his merry band of demons are in the details of the chained vulnerabilities exploited as part of this attack vector.

Complete Attack Operation

From initial access to complete server takeover

Image 1 – Chaining multiple vulnerabilities for a complete view of cybersecurity risk

---

Impact

Pentera Labs firmly believes security readiness is not measured by a single vulnerability or the security team's ability to discover or patch it. Rather, an organization's security posture is determined by whether that vulnerability can be exploited directly or chained to preceding vulnerabilities across the complete attack operation. In the case of VMware vCenter vulnerabilities, the adversary can take over the host where the vCenter application is running and all of its ESXi servers deployed in a hybrid infrastructure and virtual machines hosted and managed by the hypervisor. Recent reports of ransomware gangs encrypting large amounts of virtual machines highlight the criticality of ransomware strains adding this vector to their arsenal of TTPs.

Vulnerability Walkthrough

With our customers' increased focus on virtualization infrastructure, Pentera Labs researchers set out on a mission to further enhance its existing attack frameworks used to identify, elevate, exploit and progress attack operations. As always, Pentera's methods are to assume the attacker's view and identify ways in which the adversary can take advantage of critical weaknesses – those known and those yet to be discovered.

The Credentials Disclosure

As our starting point, we gained shell access to a fresh instance of a VMware vCenter using a low-privileged user named '*vpshere-ui*', a member of the "cis" group. So far, so good. During our research on the vCenter client, we located a file containing plaintext login credentials for the client's postgresDB: "*/etc/vmware-vpx/vcdb.properties*". This file is accessible to any of the users that are part of the "cis" group. In other words, any user that is a member of the "cis" group can connect to the vCenter's Postgres database.

Credentials to PostgresDB in vcdb.properties

Exploring the vCenter internal DB

Armed with the PostgresDB credentials, we proceeded to conduct an examination of the PostgresDB. We used the following command to connect to the DB, placing the credentials found in *vcdb.properties* to good use:

```
/opt/vmware/vpostgres/current/bin/psql -d VCDB -U vc -w <Password>
```

Here we discovered that it contains a wealth of information about the ESXi and vCenter, including information about the datacenters and virtual machines that are stored on them. For example:

VCDB query for information about virtual machines located on the ESXi Another table is the table ‘vpx_host’ containing details for a user called ‘vpxuser’ and its password phrase. The vpx_host table holds a record for each managed ESXi, each containing a user called “vpxuser” and a unique password phrase. So we retrieved the password phrase, using the command:

```
/opt/vmware/vpostgres/current/bin/psql -d VCDB -U vc -w <Password> -c 'SELECT user_name, password FROM vc.vpx_
```


Text Description automatically generated

The vpxuser password phrase Next, we dived in to investigate the “*vpxuser”*.

vpxuser – who are you?

Turns out that “*vpxuser*” is a high-privileged user automatically created on the first connection between the ESXi to the vCenter. The “*vpxuser*” user is created on the ESXi by default and is designed according to the principle of least-privilege, so it can be managed by the vCenter without the use of root. The user is created through a process called “*vpxd*” which is responsible for communication between the ESXi and the vCenter. There isn’t a lot of information about this user, but we found a comment about it on the “*passwd”* file on the ESXi. The *“passwd”* file is usually used to keep track of users that have access to a system and it contains information about the user accounts. We were able to gather a lot of information about the user “vpxuser” from the file. Eventually, it was the description that led us to our next finding – “*VMware VirtualCenter administration account*”.

We started investigating the “*vpxd*” process using IDA in an effort to better understand how the “vpxuser” and its password are created and recorded in the PostgresDB that began this journey. Our first step was to search for the string “*VMware VirtualCenter administration account*” using IDA’s *Strings* tab to see where else this string is referenced. This search led to the function related to the creation of the user on the ESXi. From there, we could identify the function where the password is created as the figure below shows.

Password creation function call Diving into the “create_random_password” function, we identified the use of Random_Crypto (open-vm-tools), which suggests that the password for this user is randomly created.


Graphical user interface, text, application Description automatically generated

Random_Crypto function in “create_random_password” We knew that the password must be saved to the vCenter’s PostgresDB since it is randomly generated upon the first connection and is required for further communication. Yet, as we previously saw in the DB, the password is not saved in plaintext. We immediately noticed that the password phrase is encoded in base64, yet the encoded phrase was not readable, so we assumed it is also encrypted.


Text Description automatically generated

The vpxuser password phrase

Decrypting the vpxuser password

We started exploring the processes the password goes through, as it is created, encrypted, encoded and finally, saved in the DB. Eventually, we found the encryption function – only to discover that it uses OpenSSL Symmetric EVP common encryption method.

A call to the function that encrypts the password

OpenSSL Symmetric Encryption EVP

Symmetric encryption uses the same secret key both to encrypt and decrypt the information. By using symmetric encryption algorithms, data is converted to a form that cannot be understood by anyone who does not possess the secret key. The EVP functions provide a high level interface to OpenSSL cryptographic functions. In order to perform symmetric encryption or decryption using EVP you will need to know:

- The algorithm – There are several types of encryption algorithms that can be used, such as block or stream algorithms. As we later discovered, AES_256_CBC is the algorithm used here.
- The key – You need to possess the secret key to decrypt the message.
- The Initialization Vector (IV) – An Initialization Vector is a random number used in combination with a secret key as a means to encrypt data. This number is sometimes referred to as a **nonce**. An IV is used to avoid repetition during the data encryption process, making it impossible for hackers who use dictionary attacks to decrypt the exchanged encrypted message by discovering a pattern.

So now that we know what a symmetric encryption looks like, let's go back and identify the password creation method. Since the *IV* is randomized, we deduced that it's probably also saved to the DB. And it is! Once we found the function where the *IV* is created, this led the road to the decryption function!


Graphical user interface, text Description automatically generated

The call to the OpenSSL decryption function We were ready to proceed, having found all 3 prerequisites needed to decrypt a symmetric encryption:

1. The algorithm – We could just tell that the algorithm is AES_256_CBC.
2. The IV – We realized that the first 16 bytes of the password phrase are the *IV*.
3. The key – We found that the encryption key is created in advance and saved to the file "*/etc/vmware-vpx/ssl/symkey.dat*".

There was still one missing piece to this puzzle. The "*/etc/vmware-vpx/ssl/symkey.dat*" file is only accessible to high-privileged users, so we needed a method that would give us root privileges to read it. Luckily, we had a method ready on stand-by thanks to our previous research and discovery of CVE-2021-22015, a privilege escalation vulnerability. Once we extracted the key, we had all the ingredients necessary to decrypt the password. We wrote a python script to extract the cleartext password and got ourselves high-privilege credentials to ESXi!

Analysis Summary

1. Logged into the vCenter's PostgresDB.
2. Found the encoded and encrypted password for vpxuser, a high-privileged user automatically created on the first connection between the ESXi to the vCenter.
3. Reverse-engineered the creation of vpxuser to devise a method for decrypting the vpxuser password.
4. Leveraged vScalation (CVE-2021-22015), a Privilege Escalation method also discovered by Pentera Labs to gain root privileges necessary to decrypt the vpxuser password.
5. Once decrypted, the compromised root account vpxuser confirms complete takeover of the ESXi server and a new zero-day is born.

To remediate CVE-2022-22948, apply the updates listed in VMware's Advisory site. There is no known workaround.

MITRE Technique

Exploitation for Credential Access (T1212) Exploitation for Privilege Escalation (T1068)

Acknowledgments

I wanted to say thanks to everyone at VMware involved with the patch for this vulnerability. They were kind, prompt in their responses, and very easy to work with.

Proof of Concept

We were ready to verify that we had enough to decrypt the password. Using a script we created for this purpose, we entered the encoded-encrypted vpxuser password phrase and the key as arguments. This is the result we received:

Our Python script decrypted the password! We used the vpxuser credentials to connect via SSH to the managed ESXi with high privileges and gained full control of the ESXi. We could control the ESXi as we wish – extract the memory of virtual machines, list inventory, get sensitive files, access sensitive information, you name it. Watch Yuval's 20-minute online session covering this vulnerability or read the PDF version of this article.
