# Ventoy Ring 0/Kernel Malware Driver and Malicious Root Cert: A Full Report and Analysis

Ventoy: A Full Report and Analysis
By Nicholas Soulliere
May 20, 2025

Ventoy is a popular software for installing or running live multiple operating system ISOs on various systems. You simply take a USB flashdrive, install Ventoy to it (acts as a simple UEFI boot loader + simple GUI), load up any and all ISOs you'd like to use on a single volume, be it Windows 11, Ubuntu, etc., then you select and boot whichever. It is a very, very unique piece of software, and largely the only one in its position that accomplishes all the requirements you'd need for a "single multiboot USB". I have personally used it for tons of projects and installs; I'd keep one 124GB 3.0 USB loaded with Windows, HBCD (Hiren's BootCD PE), Caine, Linux Rescue ISO, Mint, Ubuntu, etc. on a USB on my keychain. It works fantastically.

There are not really any equal competitors; for example GLIM is 100% FOSS, loads ISOs, and lets you choose any of those ISOs during boot. However, it is limited to a FAT32 file system (i.e. no ISOs over 4GB in size, which is a TON of them now. You can technically use another file system, but FAT32 has the highest probability of actually working), and some flat out wont boot/work, such as Windows; you'd need something like Rufus here, or well, Ventoy.

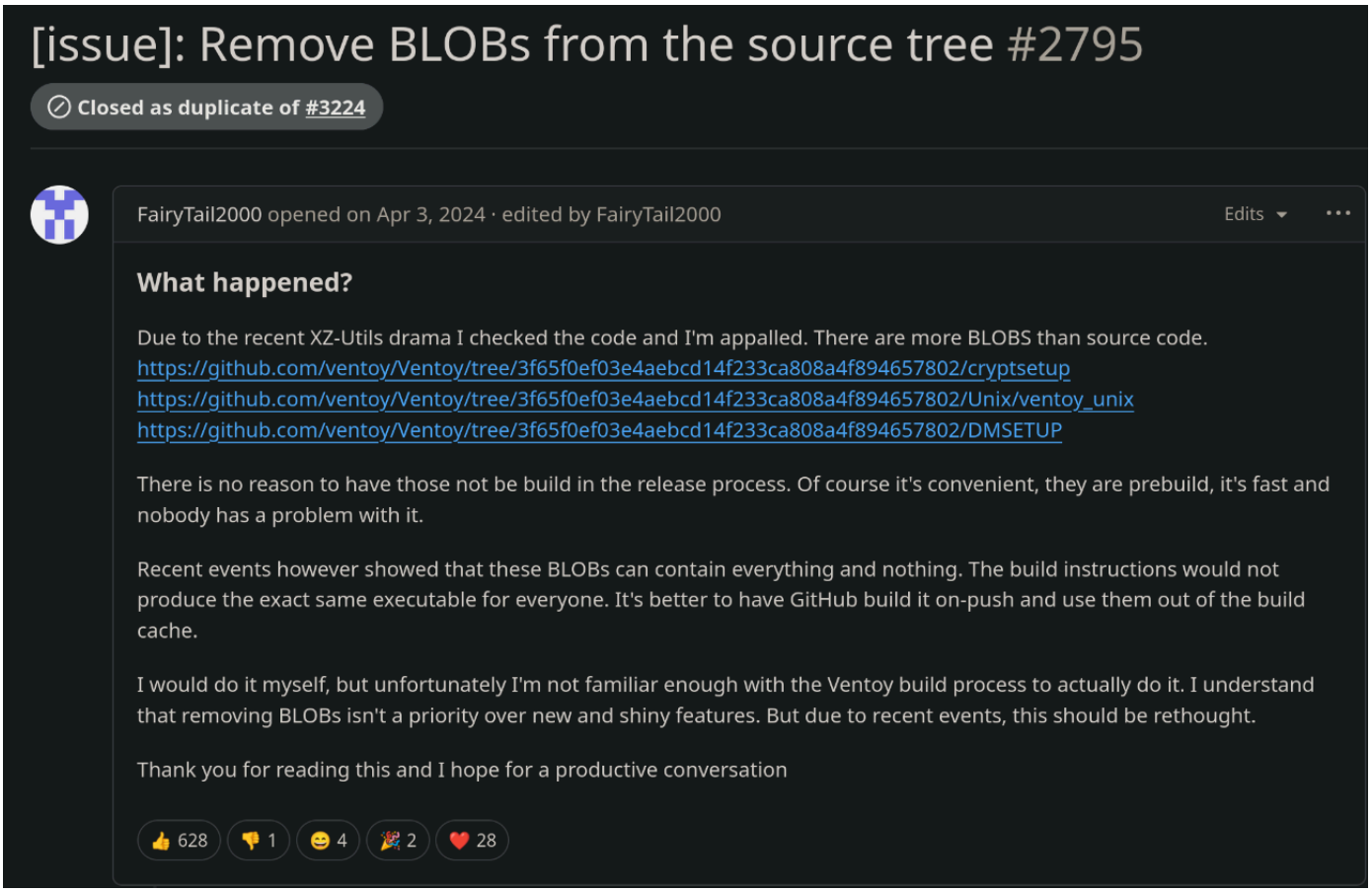
The only other option is to install one ISO per USB drive the old fashioned way, and that is obviously inconvenient if you're rotating through ISOs frequently. You would typically use something like Rufus to flash a ISO to a drive, but you'd have to reflash that USB for every OS ISO you'd need, or carry 5+ USBs at a time.

---

It was recently brought up that Ventoy had a great many precompiled blobs by user FairyTail2000 on GitHub, as he mentions, he is doing a review after the XZ-Utils backdoor fiasco, which attained a whopping CVSS score of 10.0 out of 10.0.

FairyTail2000 noticed there was an much larger than average amount of precompiled blobs that, for the most part, weren't reproducible, nor had any build instructions to go with them.

In fact, the original 39 blobs out of 153 the repo had grown to were simply missing build instructions entirely.

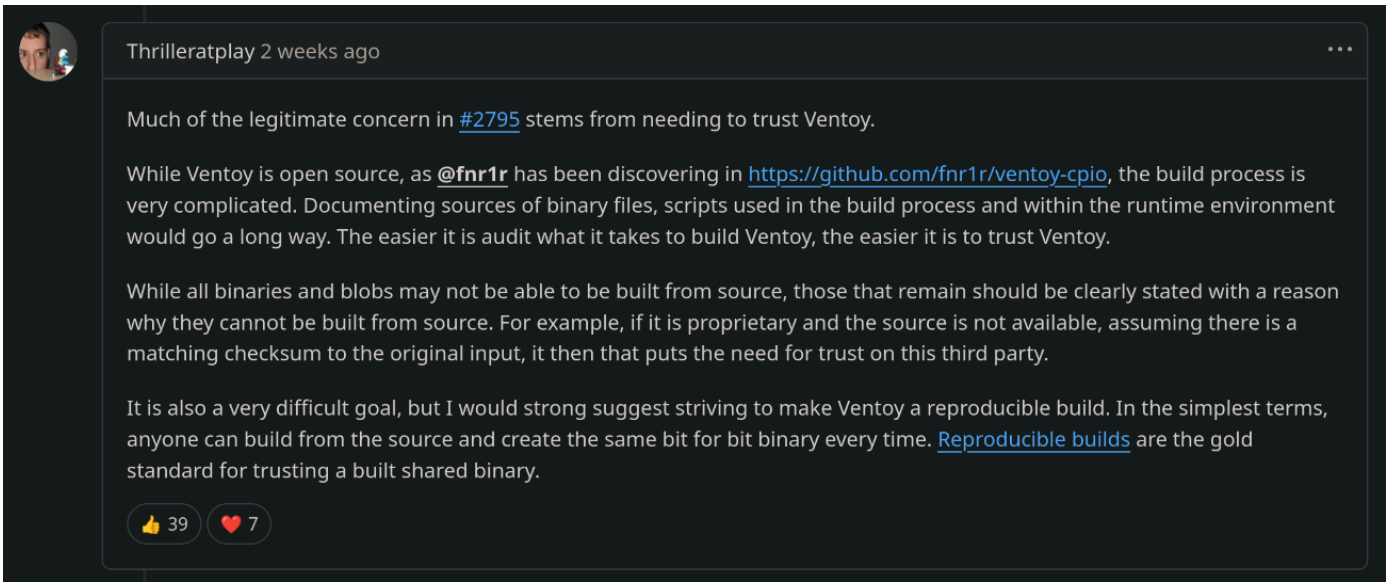



Strange things started happening to the developer at this point. He went MIA for quite some time, and other shady accounts pretending to be him popped up on other forums such as Lemmy, and elsewhere.

At the crux of the issue, the blobs themselves don't constitute an inherent security risk; in fact they are very common. What is questionable, besides the shear quantity of them, is that they need to be reproducible, which they are not in this case.

The repo itself has scripts that if ran should reproduce the blobs, however, many of them don't align after the build process. Doing a simple SHA256 against the same blob from the repo vs following the (if there are even any) build instructions and scripts often results in two different hashes.

GitHub user Thrilleratplay puts it quite succinctly, where he explains the core FOSS ethos and why FOSS is so trustworthy, and more importantly, why this actually matters:




Additionally, it is very difficult to keep track of 153 blobs. It is very unlikely someone is going to manually go through the work of rebuilding/compiling all of these blobs and actually verify any of this; not to mention the fact that at any point in time they could be changed or updated with a single commit; you would actually need to go in and manually audit each of the 153 blobs again, because any of them could've been changed. Threat actors love to exploit developer challenges like these.

Typically, this is where the non-FOSS naysayers will cleverly bring up:
"Ah yes but what about XZ-utils as you mentioned! Someone social engineered the solo dev on the project, and over the course of three years, pushed in a backdoor to the repo!"

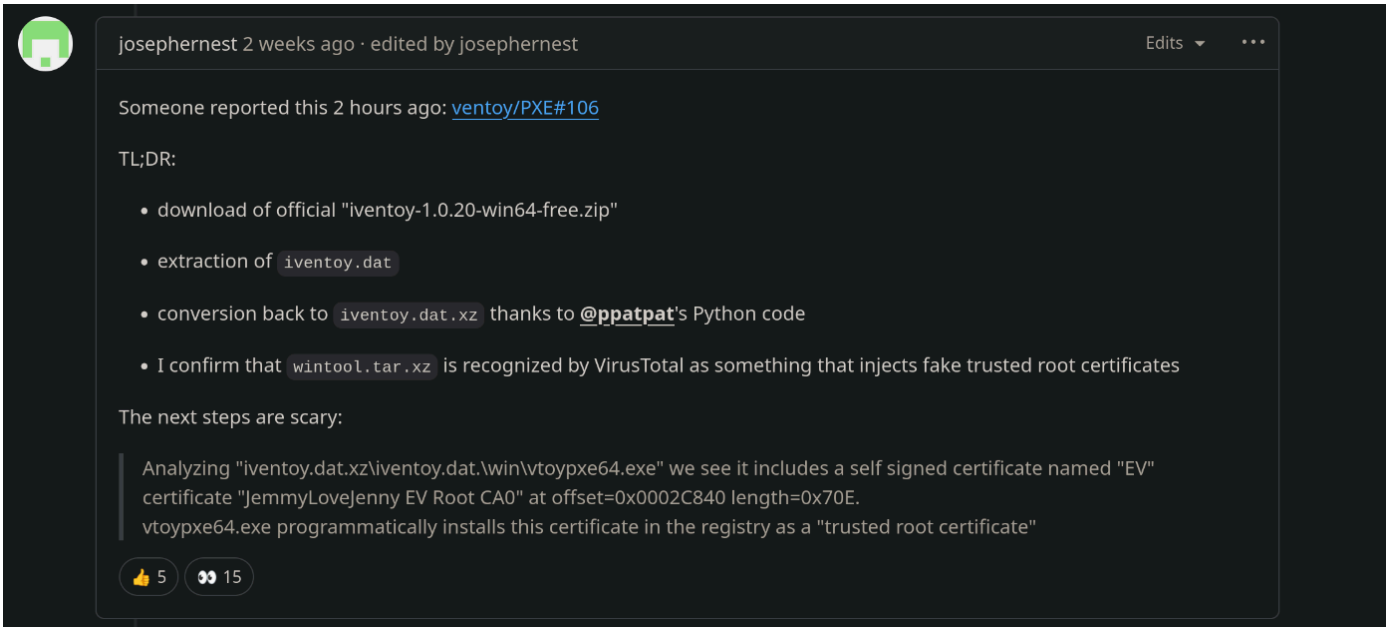
Yes, absolutely, and its a valid point. The entire point being it was caught, and fairly quickly, precisely *because* its FOSS. You can audit and probe as much as you want without any needless reverse engineering with Ghidra on proprietary code, not to mention all of the RE obfuscation put in place now for protecting IP.

---

Five months elapses with no resolution.

This is where things get bad.

User ppatpat on Github reports that iVentoy (PXE booting focused, made by the same dev), was installing unsafe Windows kernel-level drivers. For the laymen, this is what we in the industry would call a "huge fricken' deal."




Summarizing ppatpat's findings, he has created a Python script which decrypts some files from the repo.

Opening the decrypted file after running the py script, in this example, iventoy.dat.xz, and submitting them to VirusToal, we get tons of hits, along with Windows Defender.

For the uninitiated, a *general rule of thumb* for reading VT results goes like this:

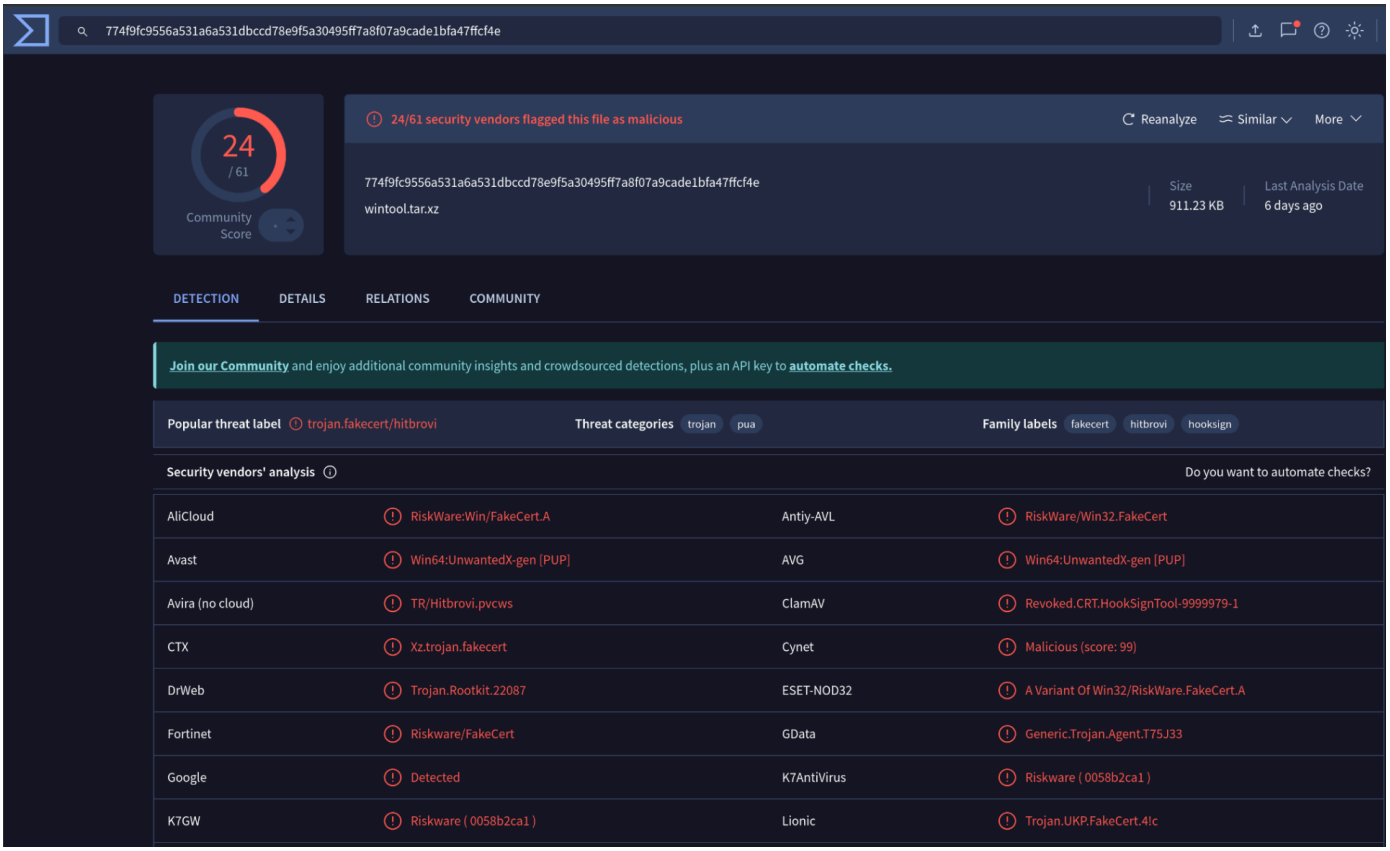
One hit or less can typically be considered a false positive.

Two hits exactly, about 50/50 odds, again, as a rule of thumb not law.

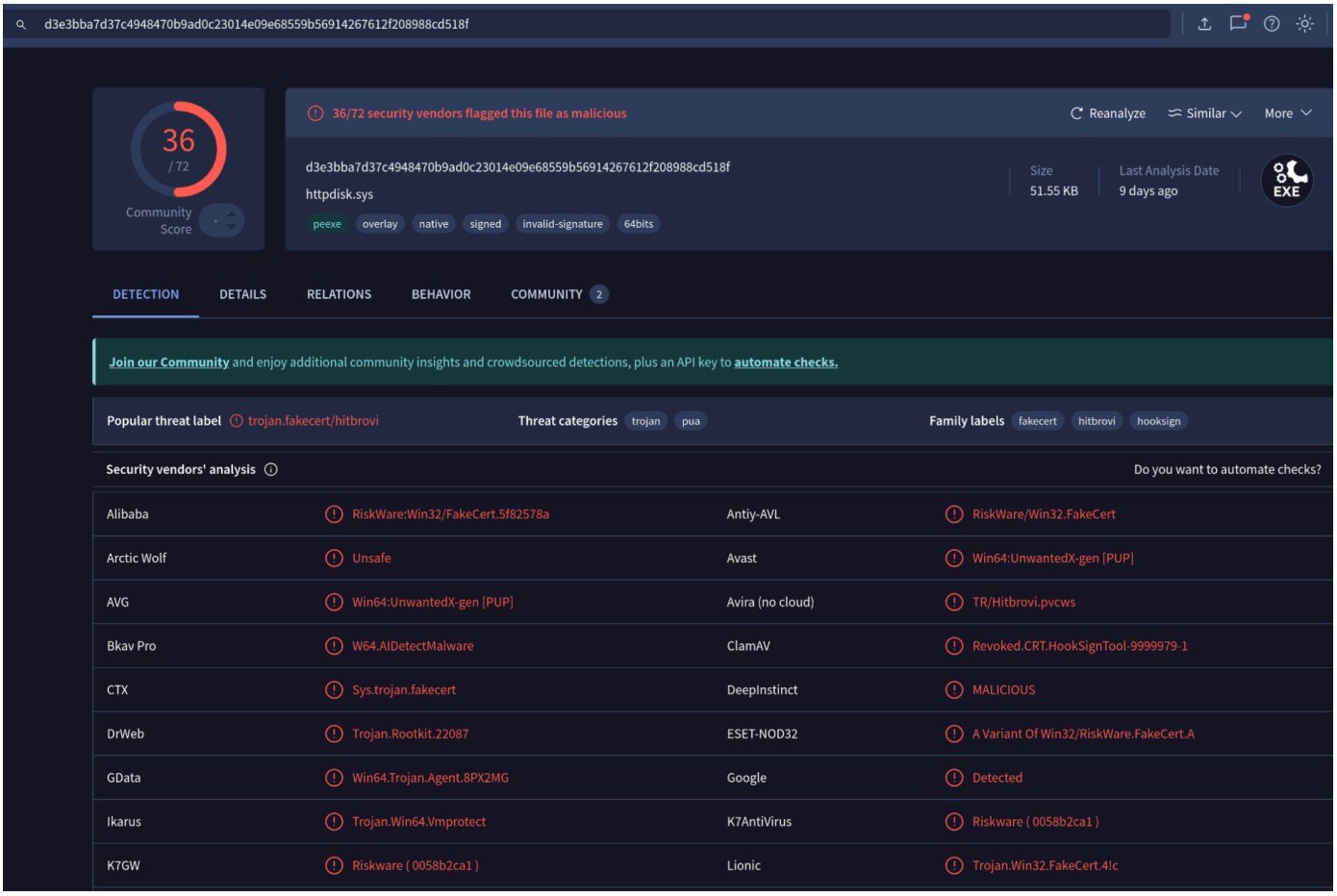
Three hits or more, and it is generally malicious. We have 24 and 36 hits respectively in the following examples.

SHA256:
774f9fc9556a531a6a531dbccd78e9f5a30495ff7a8f07a9cade1bfa47ffcf4e
File:

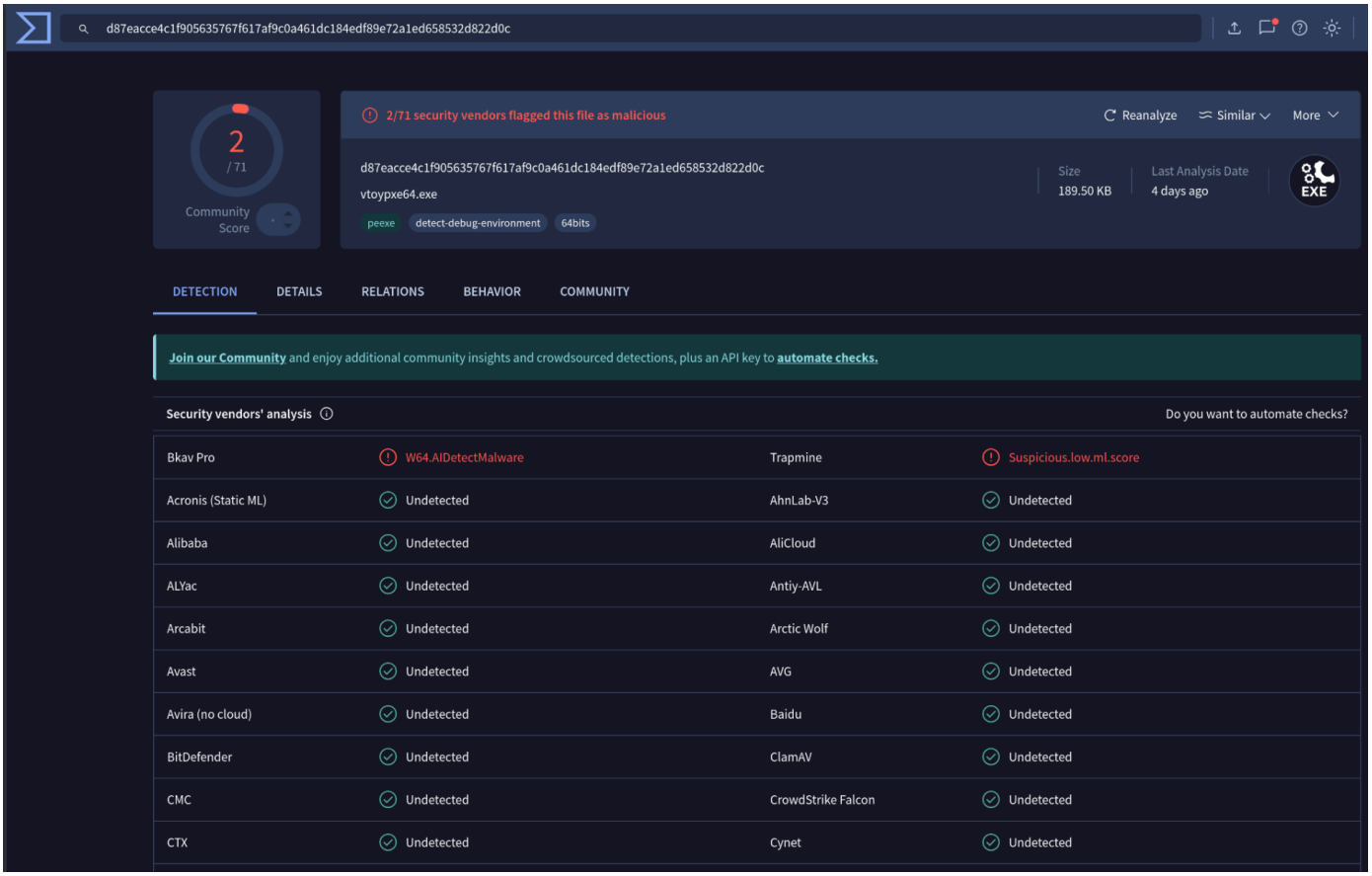
wintool.tar.xz




SHA256:
d3e3bba7d37c4948470b9ad0c23014e09e68559b56914267612f208988cd518f
File:
httpdisk.sys
Notice the threat label "fakecert"



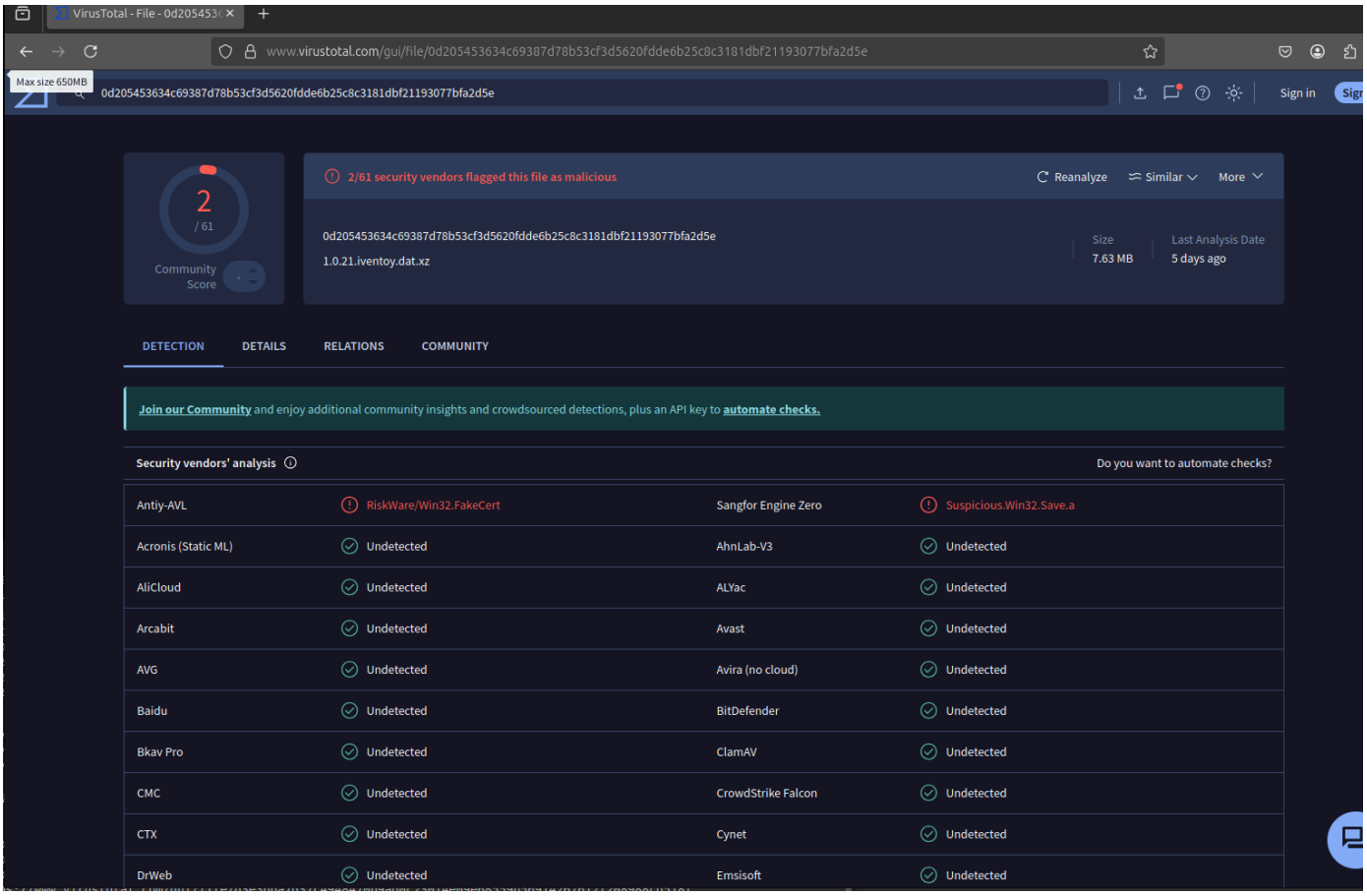
In the below case, I think its just a matter of time for other AV/EDR/XDRs and sandboxes to pick up on.
SHA256:
d87eacce4c1f905635767f617af9c0a461dc184edf89e72a1ed658532d822d0c
File:
vtoypxe64.exe



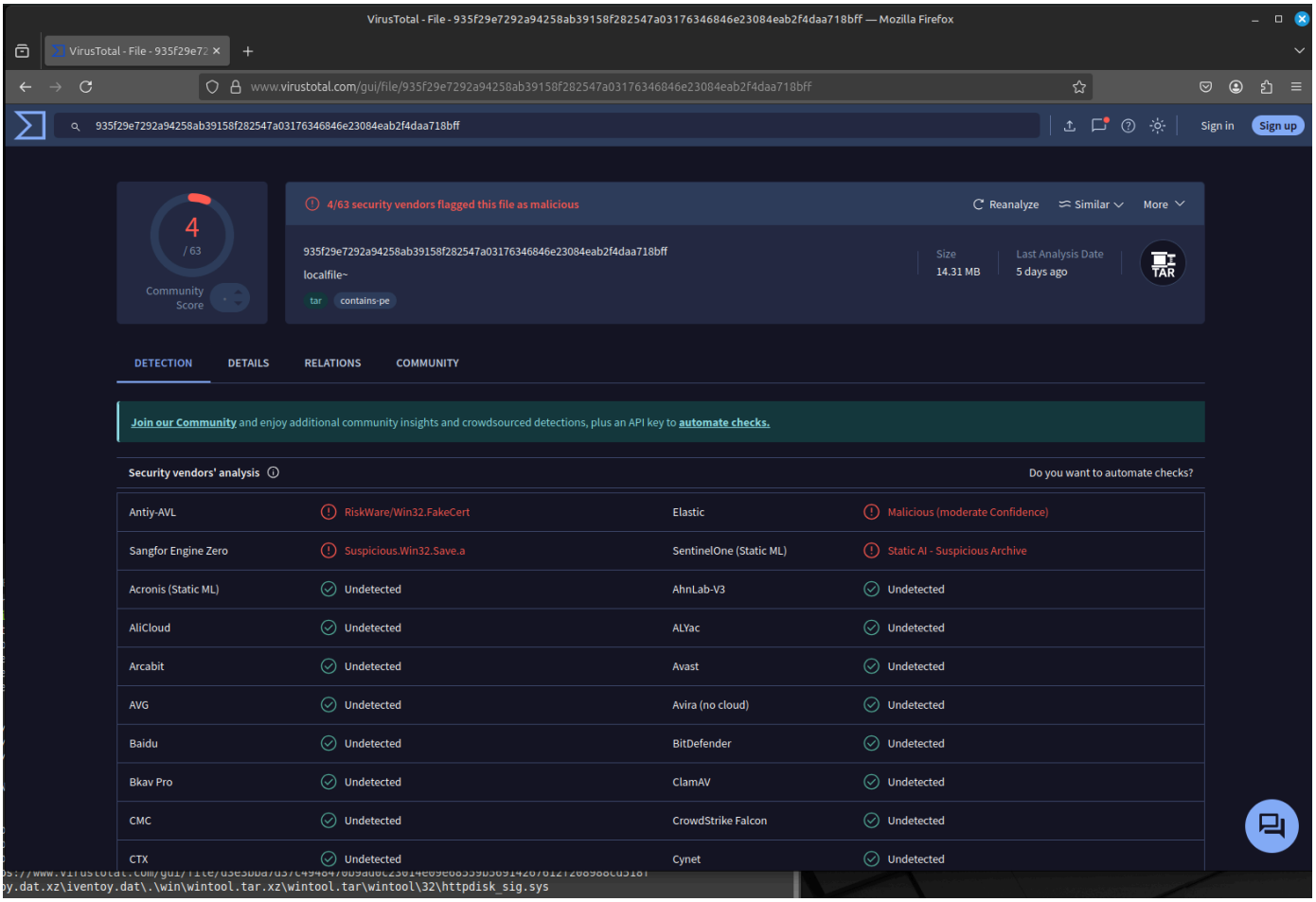

I was able to reproduce these results myself with the source code from the repo using PPatpat's python script.

```
pop@mint-vm:~/Desktop/MALWARE-iVentoy/iventoy-1.0.21/data$ python3 ../../pythonDecryptStartVentoyMalware.py
Script tested with iventoy.dat from:
  https://github.com/ventoy/PXE/releases
   iventoy-1.0.20-linux-free.tar.gz
   iventoy-1.0.20-win32-free.zip
   iventoy-1.0.20-win64-free.zip
...

File /home/pop/Desktop/MALWARE-iVentoy/iventoy-1.0.21/data/iventoy.dat has been decrypted into /home/pop/Desktop/MALWARE-iV
entoy/iventoy-1.0.21/data/iventoy.dat.xz
File /home/pop/Desktop/MALWARE-iVentoy/iventoy-1.0.21/data/iventoy.dat.xz can be opened with 7z

WARNING! File /home/pop/Desktop/MALWARE-iVentoy/iventoy-1.0.21/data/iventoy.dat.xz from iventoy-1.0.20 contain viruses/troj
ans!

iventoy.dat.xz\iventoy.dat\.\win\wintool.tar.xz\wintool.tar
  https://www.virustotal.com/gui/file/774f9fc9556a531a6a531dbccd78e9f5a30495ff7a8f07a9cade1bfa47ffcf4e
iventoy.dat.xz\iventoy.dat\.\win\wintool.tar.xz\wintool.tar\wintool\64\httpdisk_sig.sys
  https://www.virustotal.com/gui/file/d3e3bba7d37c4948470b9ad0c23014e09e68559b56914267612f208988cd518f
iventoy.dat.xz\iventoy.dat\.\win\wintool.tar.xz\wintool.tar\wintool\32\httpdisk_sig.sys

Other files could also be infected!!
pop@mint-vm:~/Desktop/MALWARE-iVentoy/iventoy-1.0.21/data$
```

Uploading the iventoy.dat.xz file




Uploading the contents of iventoy.dat.xz, iventoy.dat



PPatpat goes on to explain that included is a malicious self-signed "EV" certificate by the name of "JemmyLoveJenny EV Root CA0"

vtoypxe64.exe installs the certificate into the registry as a "trusted root certificate."

I want to reiterate the following research findings belong to PPatpat:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\ROOT\Certificates\E403A1DFC8F377E0F4AA43A83EE9EA079A1F55F2]

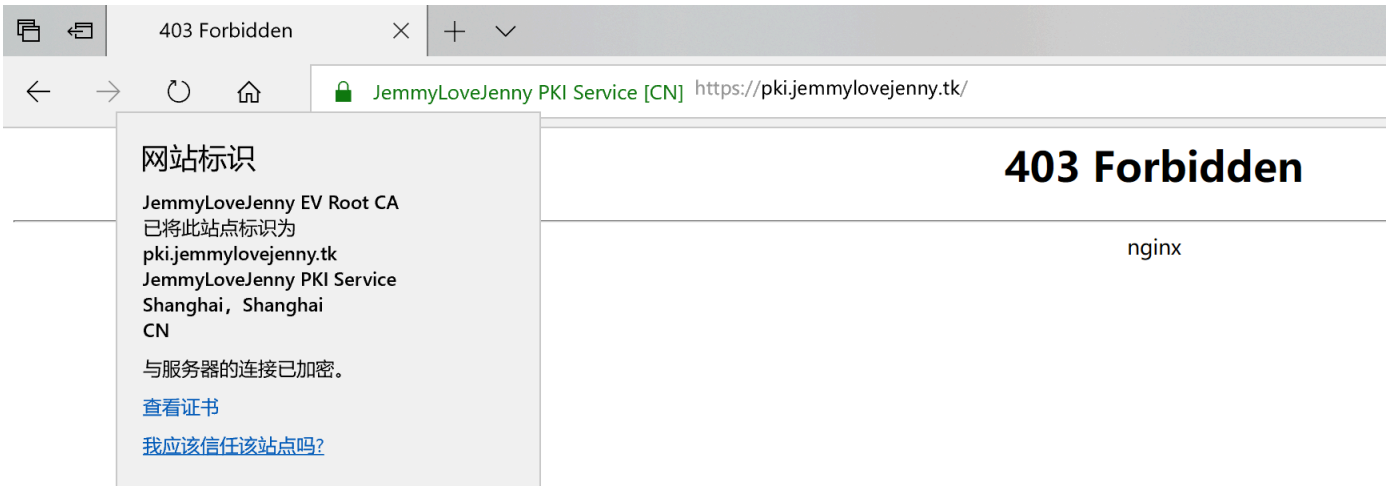
```
phkResult = 0LL;
dwDisposition[0] = 0;
v30 = RegCreateKeyExA(
        HKEY_LOCAL_MACHINE,
        "SOFTWARE\\Microsoft\\SystemCertificates\\ROOT\\Certificates\\E403A1DFC8F377E0F4AA43A83EE9EA079A1F55F2",
        0,
        0LL,
        0,
        0xF003Fu,
        0LL,
        &phkResult,
        dwDisposition);
if ( v30 )
{
  LastError = GetLastError();
  sub_140003520("Failed to create CA reg key %u %u", LastError, v30);
}
else
{
  dwDisposition[0] = 1806;
  v31 = RegSetValueExA(phkResult, "Blob", 0, 3u, &Data, 0x70Eu);
  v32 = "FAILED";
  if ( !v31 )
    v32 = "SUCCESS";
  sub_140003520("Create ca registry %s %u", v32, v31);
}
```

PPatpat:
Next vtoypxe64.exe tries to load the following ring 0 kernel drivers in sequence:

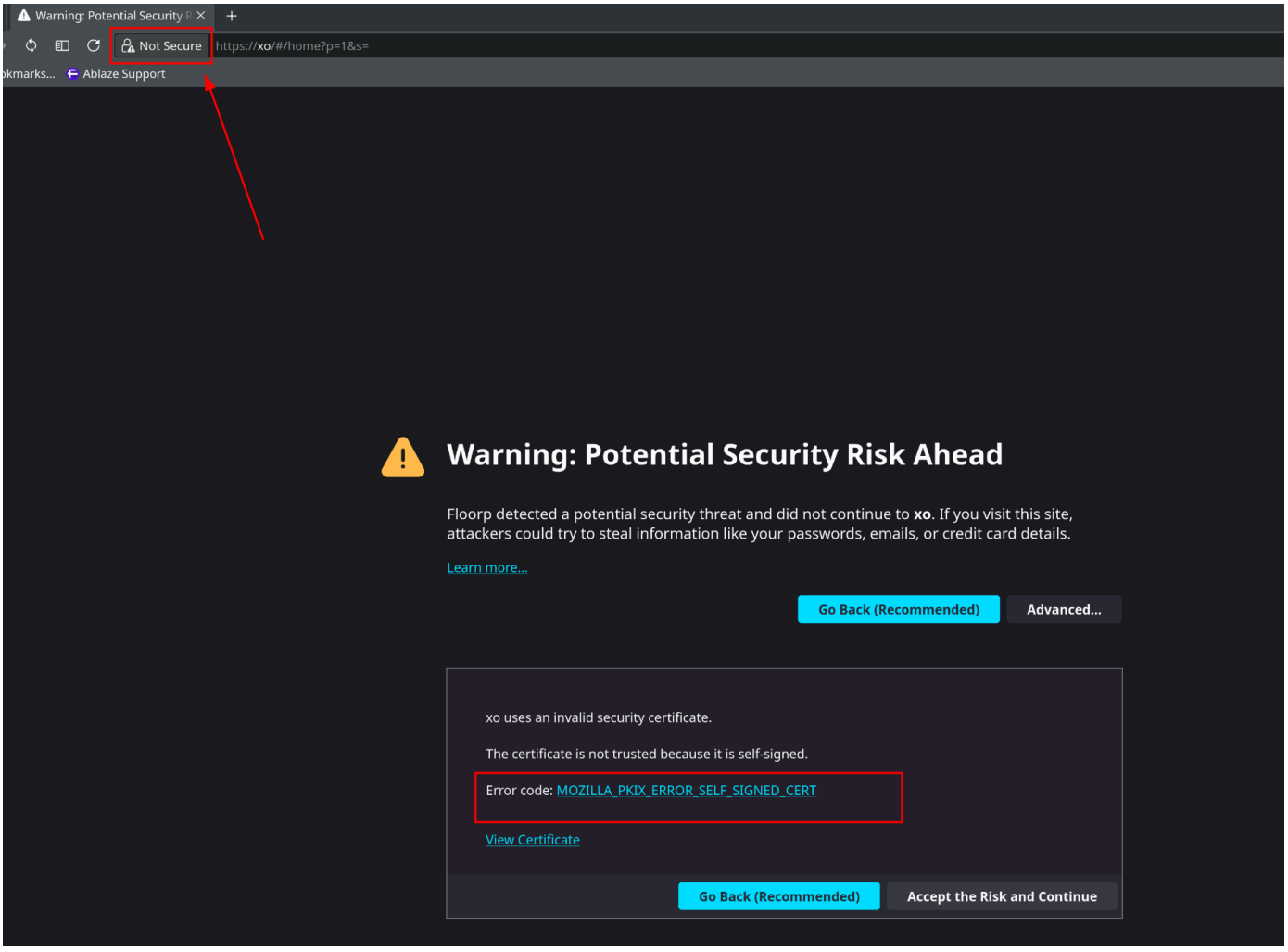
"\ventoy\httpdisk.sys", "\ventoy\httpdisk_sig.sys", "\ventoy\httpdisk_nosig.sys"

Interestingly, the exact fake root certificate creation process is broken down here by user "Jemmy1228" on Stack Exchange

The TL;DR for this post is that you can forge an EV cert and have it pass in some browsers as a legitimate one, which is very bad, to say the least.




This example provided by Jemmy1228 is a fake certificate showing as trusted and legitimate. Whereas, you should be seeing the following with a self-signed cert:




The lead dev does finally address some of these concerns, however, his reply leaves a lot to be desired.

In essence, the Ventoy dev explains that he has a really good reason for why he was using a known malicious exploit to make the software work, and furthermore, it only ever existed in memory, in the Window PE.

My heart goes out to all FOSS devs; its thankless, tedious, stressful and they do it for free.

However, if after you asked me to watch your kids for the weekend, you found out five months later I had thrown a wild house party with strippers and drugs, but I had REALLY good reason, and I made sure to cleanup and hide it from you, just trust me bro; I would say its safe to say you'd never let me watch your kids again.

---

**Conclusions and Takeaways**

- As noted by GitHub user elypter2, **Ventoy is a highly appealing vector for malware distribution**; since at its core, its designed to target the installation of *any* operating system.


- China is home to some of the most powerful Advanced Persistent Threat (APT) groups in the world; groups who have routinely reeked havoc on US and NATO allied nation's infrastructure time, and time again. APTs such as Salt Typhoon, affiliated with China's Ministry of State Security. These nation-state threat actors pose a significant risk not only to the enterprise sphere, as we've seen with almost every major US telcom provider being hacked recently, but to the US government and entire nation as a whole. The dev for Ventoy is of Chinese origin; the tactics, techniques, and procedures (TTPs) we've seen align with these APT groups based out of China.

  XZ-utils backdoor was also from a dev of Chinese origin, and with the skill set they possessed, and long term objective focus of three years to infiltrate the project and become one of the primary dev's, Jia Tan, is another example of the resources these nation-state threat actors and APT groups possess. They can, will, and do on a regular basis, perform massive, complicated operations, such as infiltrating FOSS projects and breaking in to critical infrastructure. The threat they present should never be underestimated.

- **Enterprises, and particular the public sector, should immediately cease to use Ventoy**, and **any** workstation or server that has been installed with Ventoy, wiped and reinstalled. Full forensics should be performed as well to ensure nothing remains or has spread to other endpoints. Logs should be reviewed. Enterprises should rely on more traditional or trusted methods, like the USB installation media software Microsoft provides for its Windows 11 installs, for example. If you need any other OS installed to a USB, and you work on Windows, Rufus has been the tried and true standard for a long time, and is FOSS so you can audit the source code yourself. For Linux workstations, `dd`'ing an ISO to a USB is still a tried, true, tested, and secure method.