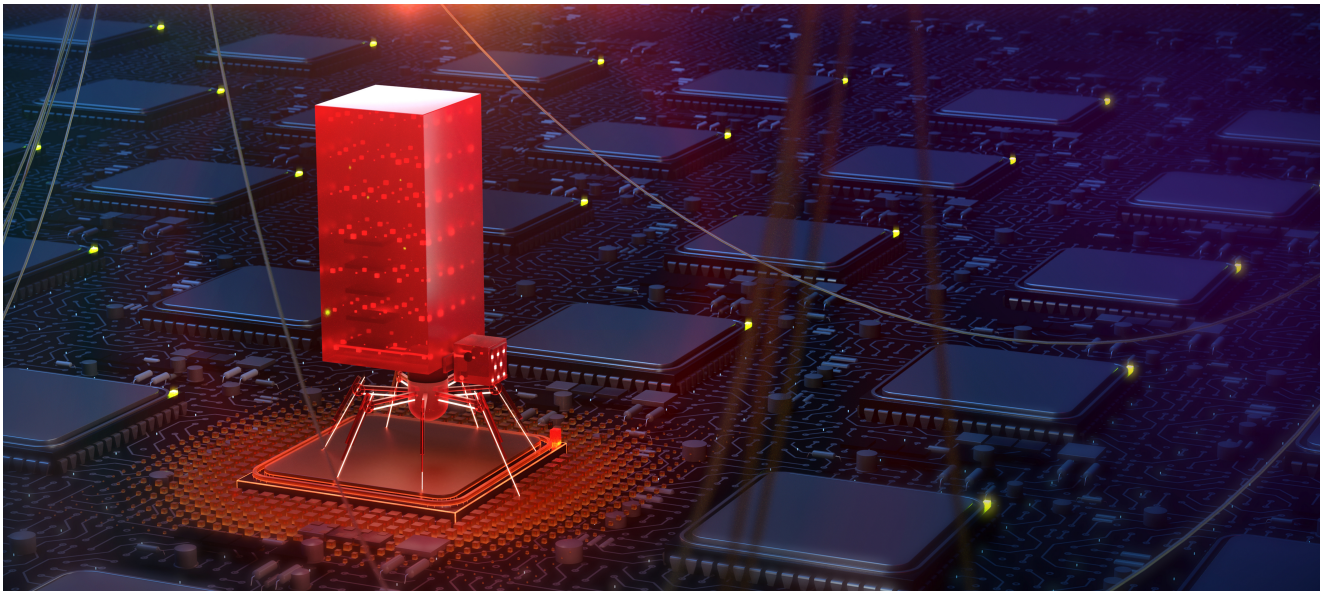
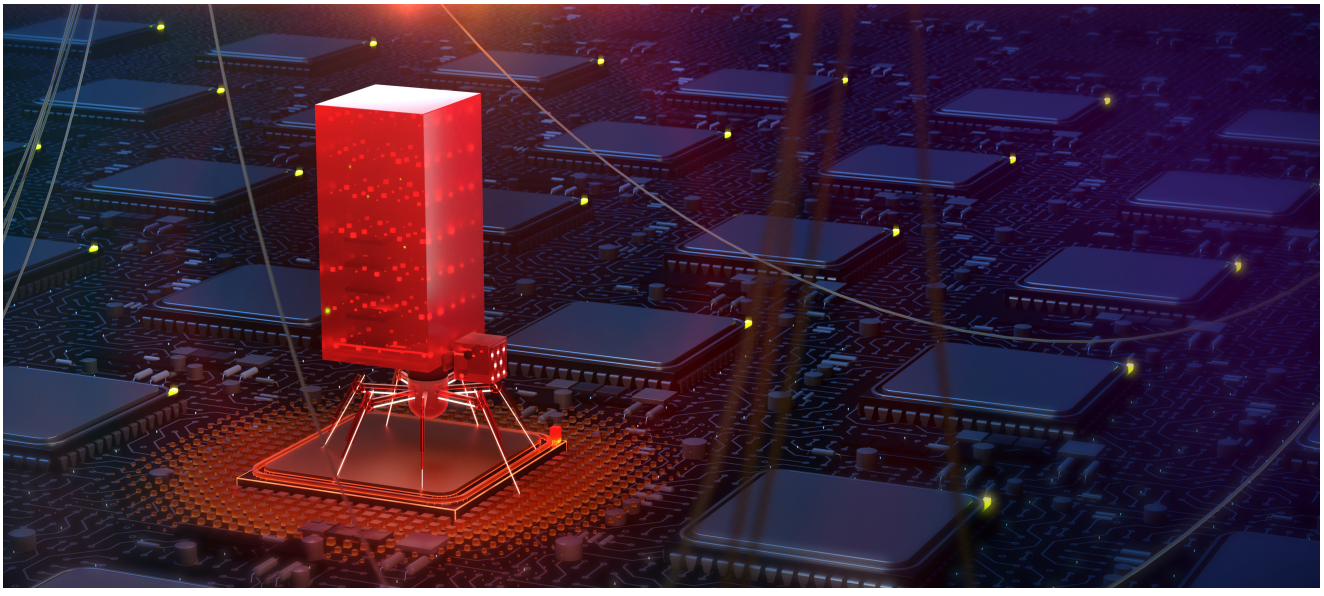
Taurus Loader: User-Guided Infection

blog.minerva-labs.com/taurus-user-guided-infection





- Tweet
-

In recent months we have seen a spike in events associated with Taurus loader. Although Taurus has already been covered extensively by researchers, we think its spreading method was left untouched by the security community. This method allows this malware to generate new samples and infect new devices continuously.

Our research saw Taurus being downloaded using promoted cracked software sites. An individual trying to download pirated installations, using Google search engine, will discover sites containing this informative GIF:

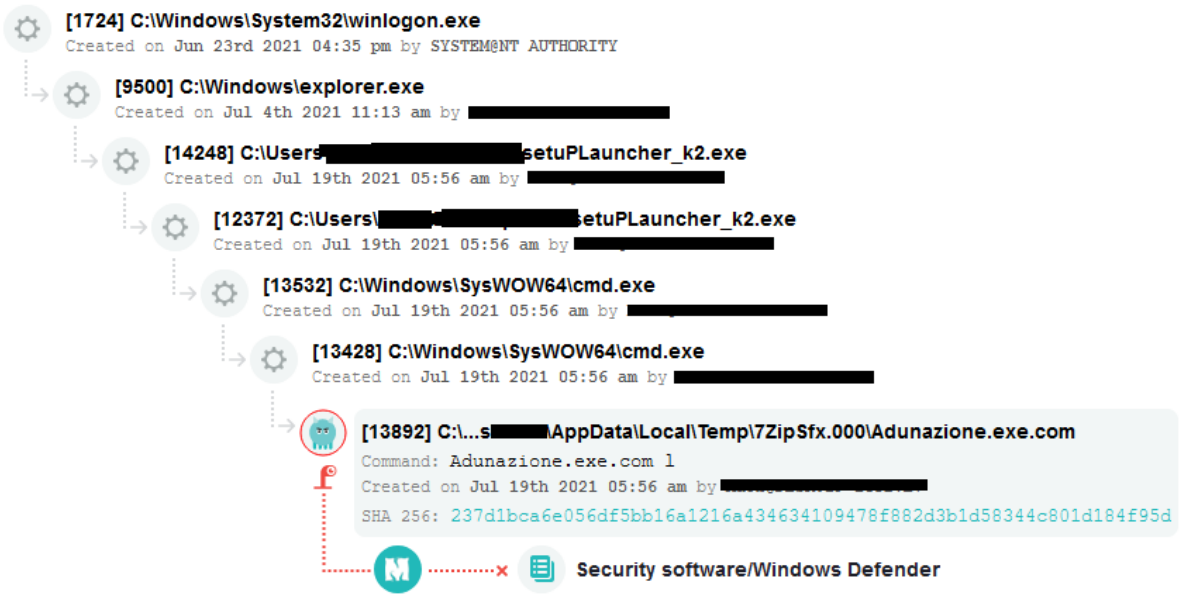
It will instruct the unsuspecting user through the installation process of the illicit software of his or her choosing. Alas, following this interactive guide will result in an installation of Taurus loader. This social engineering trick increases the infection rate by allowing even non-technical users to successfully execute this evasive malware. To top it all, the download page is protected from bots by captcha, effectively preventing researchers from automatically extracting Taurus' payloads.

Another less documented aspect of Taurus is its final payload decryption, which is achieved using a specially crafted machine code snippet. As detailed in our previous piece, Taurus uses AutoIt to perform various evasion techniques, and if a machine is deemed "safe" a payload will be decrypted in memory and executed. Instead of implementing its decryption algorithm in AutoIt, the malware developers have chosen to write an assembly implementation of their chosen stream cipher, RC4.

The RC4 encryption routine, as seen in Hex-Ray's decompiler:

```
result = a1;
v5 = *a1;
v6 = a1[1];
if ( a3 > 0 )
{
  v7 = 0;
  v10 = *a1;
  do
  {
    v10 = (unsigned __int8)(v10 + 1);
    v8 = *((_BYTE *)a1 + v10 + 8);
    v6 = (unsigned __int8)(v8 + v6);
    v9 = *((_BYTE *)a1 + v6 + 8);
    *((_BYTE *)a1 + v10 + 8) = *((_BYTE *)a1 + v6 + 8);
    *((_BYTE *)a1 + v6 + 8) = v8;
    *(_BYTE *)(a2 + v7++) ^= *((_BYTE *)a1 + (unsigned __int8)(v9 + v8) + 8);
  }
  while ( v7 != a3 );
  v5 = v10;
}
*a1 = v5;
a1[1] = v6;
return result;
```

Minerva prevents Taurus loader with our hostile environment simulation, preventing the malware from using its own code:



IOCs:

Hashes:

c0cf7e674127446f47f5fe7b86497ecf229e16f73aa2115991d09716006de668

62d7b4a61f5d4b7ca24e76455cb41d2aa797f2ea3927b3bb08b1b48712be4fe2

dd9a3a5efe274380e34fd498215c8e1c4f88a45081a73b023444ddf04295d903

d6a6df7af65dd7b36b5e3fc93e39a9502305d8de61e3d9d85b8a509fb4698002

eea97d0c1c57a11b629c3b235cbafe52cd41f5c0d8ec6f1f9c7a5b6a7281ece0

03110bdd6c0d0f027422de86d47ce39cfb8cae70c04a616cbb8c325c03853ef6

dafc932c76f461d6ad517dc1a7a125e498bdd954acd0f5f0b217f4e469b10416

2cd5cef50e109e70cba18b914ba9b4de01a7b9f005ec7e08e1ed63e963d67606

141393f1b8984349e7577ca5e12af707f4563f4c9e4fe49d41fc61aff951b1a6

DNS:

https://chcracked[.]com/ (the site redirects to the malicious website)

https://profreefiles[.]com

References:

You can learn more about how Minerva's technology can prevent even the stealthiest of malware attacks, by contacting us today.