WhisperKill vs WhiteBlackCrypt: un petit soucis de fichiers…

sebdraven.medium.com/whisperkill-vs-whiteblackcrypt-un-petit-soucis-de-fichiers-9c4dcd013316

Sebdraven February 1, 2022

Sebdraven

Jan 31

.

2 min read

Fin de semaine dernière, le CERT UA publie un article détaillant que WhisperKill utilisé pour détruire les disques lors de l'attaque du #WhisperGate de ses victimes serait un copy cat de WhiteBlackCrypt.

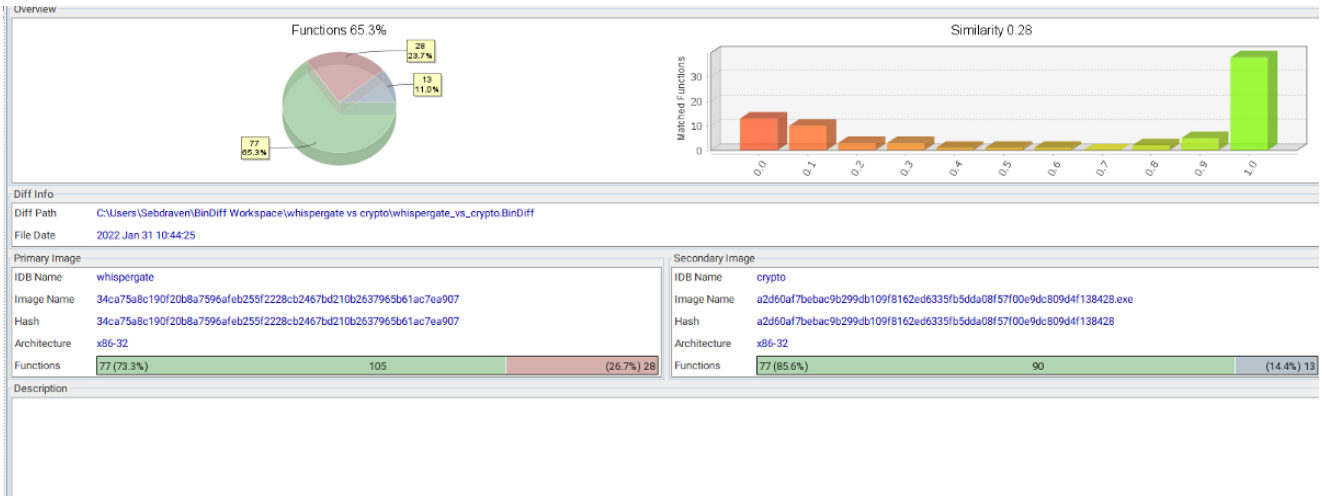
CERT-UA

Урядова команда реагування на комп'ютерні надзвичайні події України, яка функціонує в складі Державного центру…

cert.gov.ua

avec une similarité des fonctions de 91 %

Le soucis est quand on reprend les mêmes hashes que l'étude et qu'on refait l'expérience, nous tombons à 28 %.



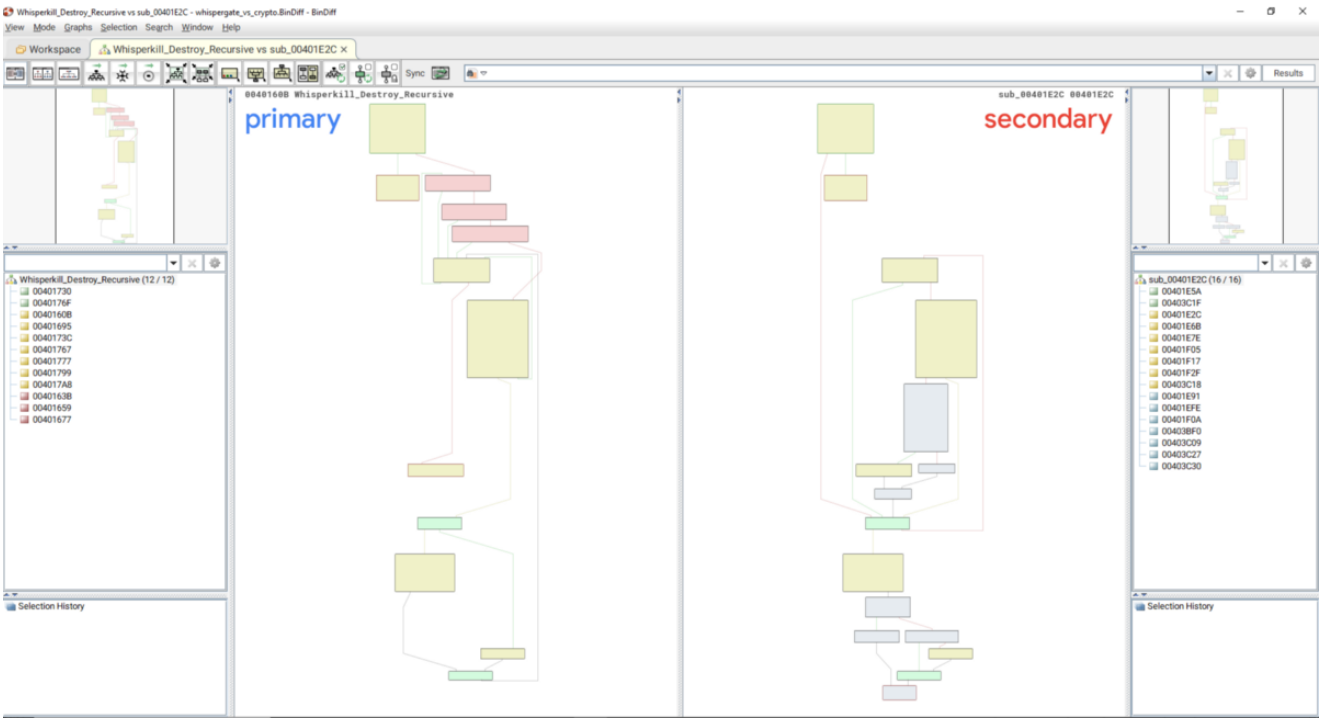
la similarité sur la fonction isDirectory est forcée, car si l'on fait une recherche sur le masque utilisé:

```
bool __cdecl isDirectory(undefined4 param_1)

{
  int iVar1;
  bool bVar2;
  undefined local_30 [6];
  ushort local_2a;

  iVar1 = _wstat(param_1,local_30);
  bVar2 = false;
  if (iVar1 == 0) {
    bVar2 = (local_2a & 0xf000) == 0x4000;
  }
  return bVar2;
}
```

(local_2a & 0xf000) == 0x4000;

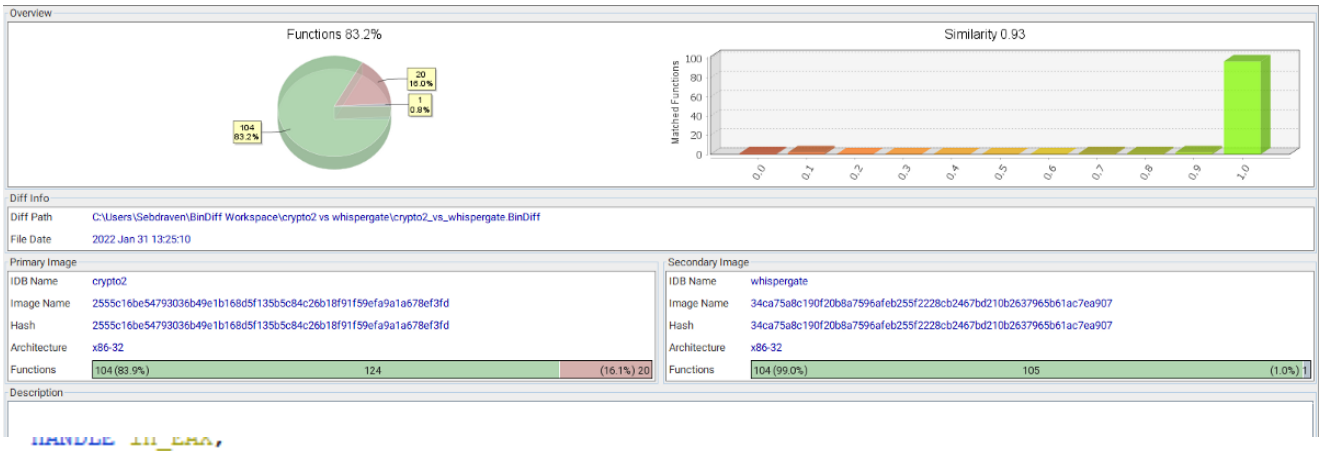
Il y a beaucoup de codes source qui l'utilisent. Donc en terme de discriminant, ce n'est pas suffisant.

Il n'y a vraiment qu'une similarité intéressante, c'est la fonction de destruction/chiffrement. Mais idem, les mécanismes restent proche d'un ransomware.



Après avoir contacté le CERT UA, les hashes qui ont été publiés ne sont pas les bons. https://twitter.com/_CERT_UA/status/1488138913818554372?s=20&t=wl6hKRIEhc-zgVd50q6bxw

Lorsque l'on refait les expériences du papier avec ceux cités ci-dessus, nous nous retrouvons bien avec les bonnes valeurs et les bonnes fonctions




```
HANDLE in_EAX;
BOOL BVar3;
DWORD DVar4;
int *piVar5;
char *pcVar6;
_WIN32_FIND_DATAA local_14c;

BVar3 = FindNextFileA(in_EAX,(LPWIN32_FIND_DATAA)&local_14c);
if (BVar3 == 0) {
  DVar4 = GetLastError();
  if (DVar4 != 0x12) {
    piVar5 = _errno();
    *piVar5 = 2;
    return 0;
  }
}
else {
  pcVar6 = (char *)(param_2 + 0xc);
  *(undefined2 *)(param_2 + 6) = 0;
  uVar2 = 0;
  while (cVar1 = local_14c.cFileName[uVar2], *pcVar6 = cVar1, cVar1 != '\0') {
    uVar2 = *(short *)(param_2 + 6) + 1;
    *(ushort *)(param_2 + 6) = uVar2;
    pcVar6 = pcVar6 + (uVar2 < 0x104);
  }
  if (0x10 < (local_14c.dwFileAttributes & 0xffffff58)) {
    *(undefined4 *)(param_2 + 8) = 0x18;
    return BVar3;
  }
  *(DWORD *)(param_2 + 8) = local_14c.dwFileAttributes & 0xffffff58;
}
```

Encrypt3D_DestroyRecursive

```
hFindFile = FindFirstFileW(param_1,(LPWIN32_FIND_DATAW)&local_26c);
if (hFindFile != (HANDLE)0xffffffff) {
  do {
    iVar2 = wcscmp(local_26c.cFileName,L".");
    if (iVar2 != 0) {
      iVar2 = wcscmp(local_26c.cFileName,L"..");
      if (iVar2 != 0) {
        iVar2 = wcscmp(local_26c.cFileName,L"$RECYCLE.BIN");
        if (iVar2 != 0) {
          sVar3 = wcslen(local_26c.cFileName);
          sVar4 = wcslen(param_1);
          iVar2 = sVar3 + sVar4;
          _Dest = (wchar_t *)malloc((iVar2 + 4) * 2);
          wcscpy(_Dest,param_1);
          _Dest[sVar4 - 1] = L'\0';
          wcscat(_Dest,local_26c.cFileName);
          pwVar5 = L"A:\\Windows";
          pwVar8 = local_282;
          for (iVar7 = 0x16; iVar7 != 0; iVar7 = iVar7 + -1) {
            *(undefined *)pwVar8 = *(undefined *)pwVar5;
            pwVar5 = (wchar_t *)((int)pwVar5 + 1);
            pwVar8 = (wchar_t *)((int)pwVar8 + 1);
          }
```

WhisperKill_DestroyRecursive

Il y a donc une vrai tentative de copycat pour ce virus dont le but est de reprendre la structure et les fonctionnalités.

Par ailleurs, beaucoup d'articles ont été publiés avec la première version des hashes sans vérification. Ce qui est dommageable quand cela vient d'équipe de Threat Intelligence ou d'analyse de malware.