HermeticWiper | New Destructive Malware Used In Cyber Attacks on Ukraine

sentinelone.com/labs/hermetic-wiper-ukraine-under-attack

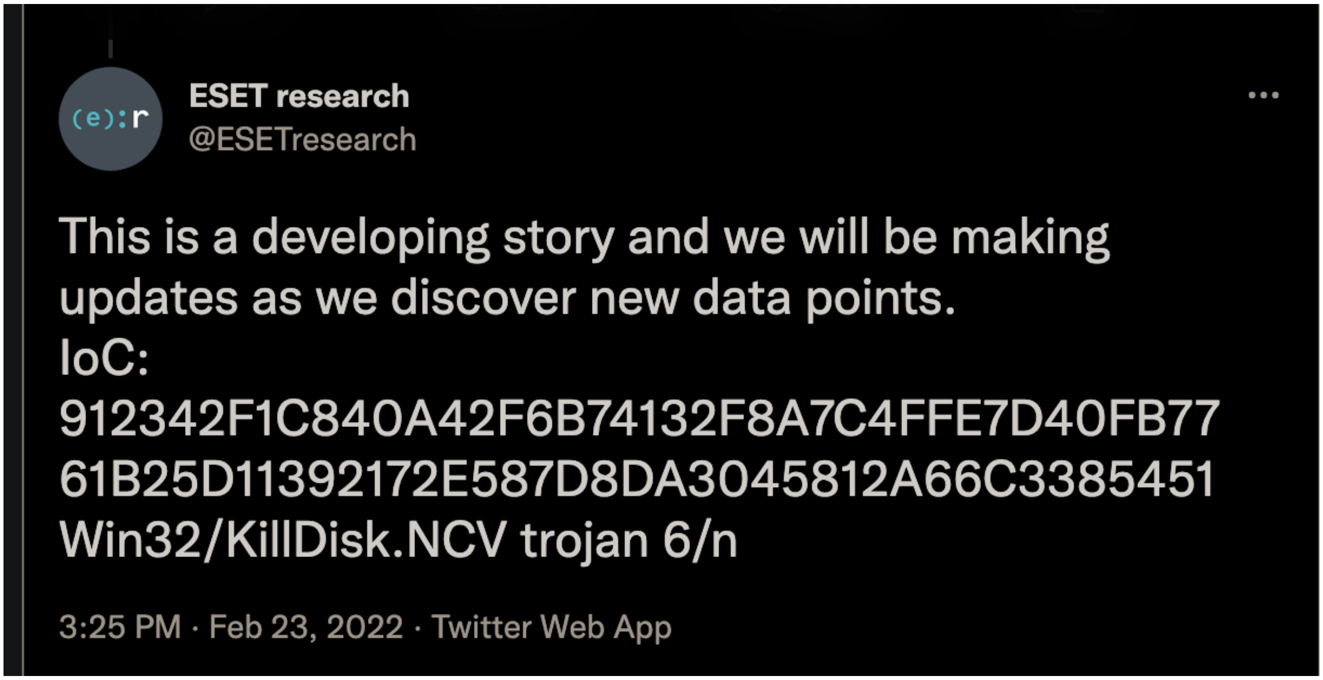
Juan Andrés Guerrero-Saade

Executive Summary

- On February 23rd, the threat intelligence community began observing a new wiper malware sample circulating in Ukrainian organizations.
- Our analysis shows a signed driver is being used to deploy a wiper that targets Windows devices, manipulating the MBR resulting in subsequent boot failure.
- This blog includes the technical details of the wiper, dubbed HermeticWiper, and includes IOCs to allow organizations to stay protected from this attack.
- This sample is actively being used against Ukrainian organizations, and this blog will be updated as more information becomes available.
- SentinelOne customers are protected from this threat, no action is needed.

Background

On February 23rd, our friends at Symantec and ESET research tweeted hashes associated with a wiper attack in Ukraine, including one which is not publicly available as of this writing.

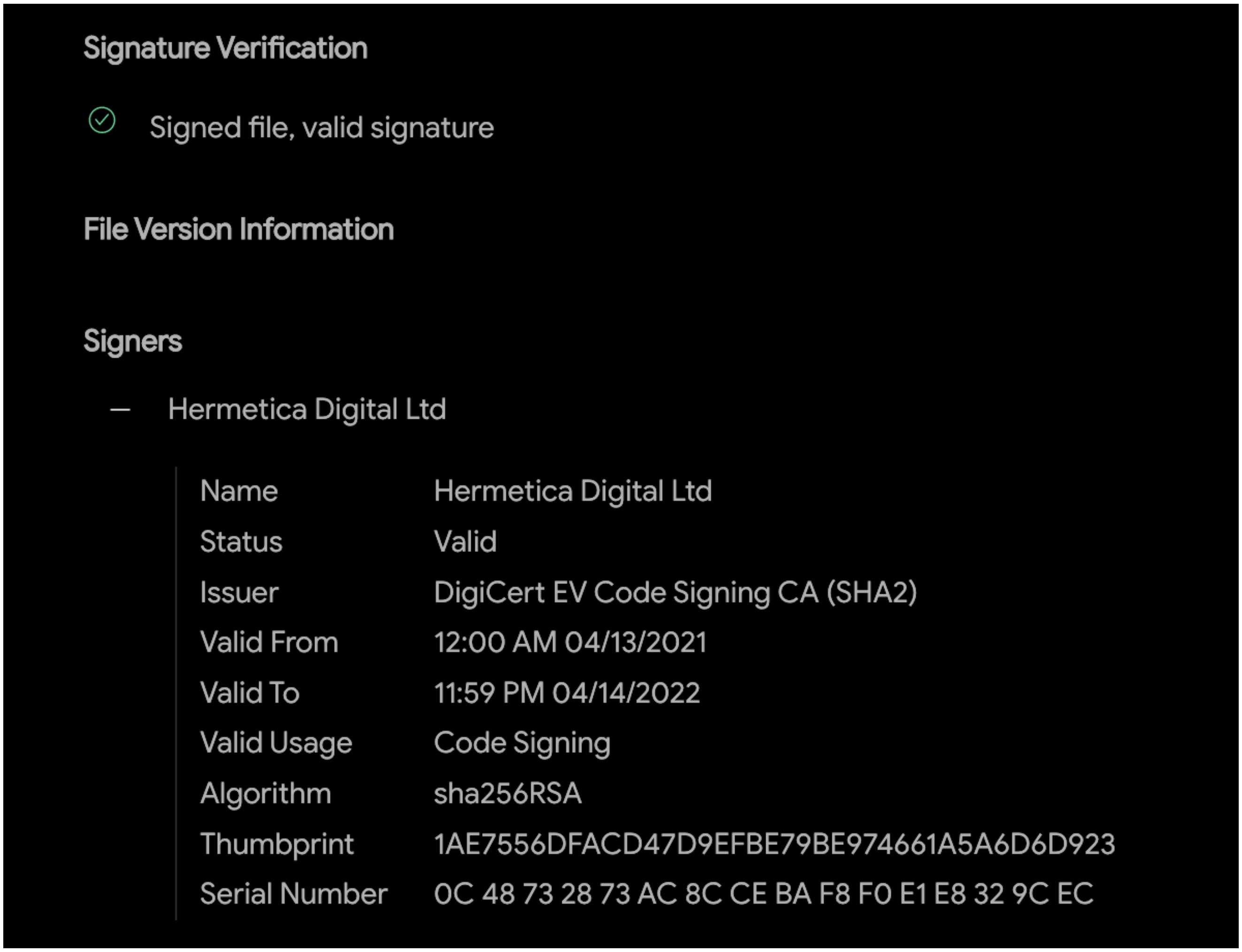


ESET research
@ESETresearch

This is a developing story and we will be making updates as we discover new data points.
IoC:
912342F1C840A42F6B74132F8A7C4FFE7D40FB77
61B25D11392172E587D8DA3045812A66C3385451
Win32/KillDisk.NCV trojan 6/n

3:25 PM · Feb 23, 2022 · Twitter Web App

We started analyzing this new wiper malware, calling it ‘HermeticWiper’ in reference to the digital certificate used to sign the sample. The digital certificate is issued under the company name ‘Hermetica Digital Ltd’ and valid as of April 2021. At this time, we haven’t seen any legitimate files signed with this certificate. It’s possible that the attackers used a shell company or appropriated a defunct company to issue this digital certificate.



Signature Verification

✓ Signed file, valid signature

File Version Information

Signers

— Hermetica Digital Ltd

Name	Hermetica Digital Ltd
Status	Valid
Issuer	DigiCert EV Code Signing CA (SHA2)
Valid From	12:00 AM 04/13/2021
Valid To	11:59 PM 04/14/2022
Valid Usage	Code Signing
Algorithm	sha256RSA
Thumbprint	1AE7556DFACD47D9EFBE79BE974661A5A6D6D923
Serial Number	0C 48 73 28 73 AC 8C CE BA F8 F0 E1 E8 32 9C EC

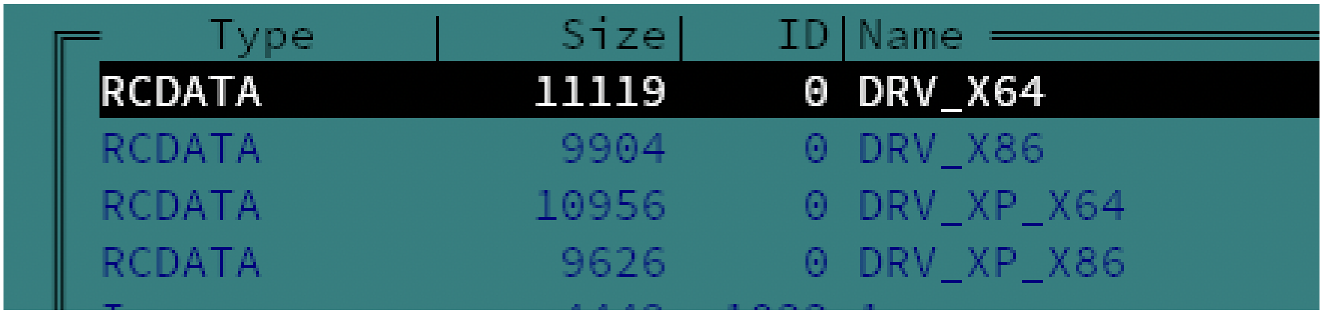
HermeticWiper Digital Signature

This is an early effort to analyze the first available sample of HermeticWiper. We recognize that the situation on the ground in Ukraine is evolving rapidly and hope that we can contribute our small part to the collective analysis effort.

Technical Analysis

At first glance, HermeticWiper appears to be a custom-written application with very few standard functions. The malware sample is 114KBs in size and roughly 70% of that is composed of resources. The developers are using a tried and tested technique of wiper malware, abusing a benign partition management driver, in order to carry out the more damaging components of their attacks. Both the Lazarus Group (Destover) and APT33

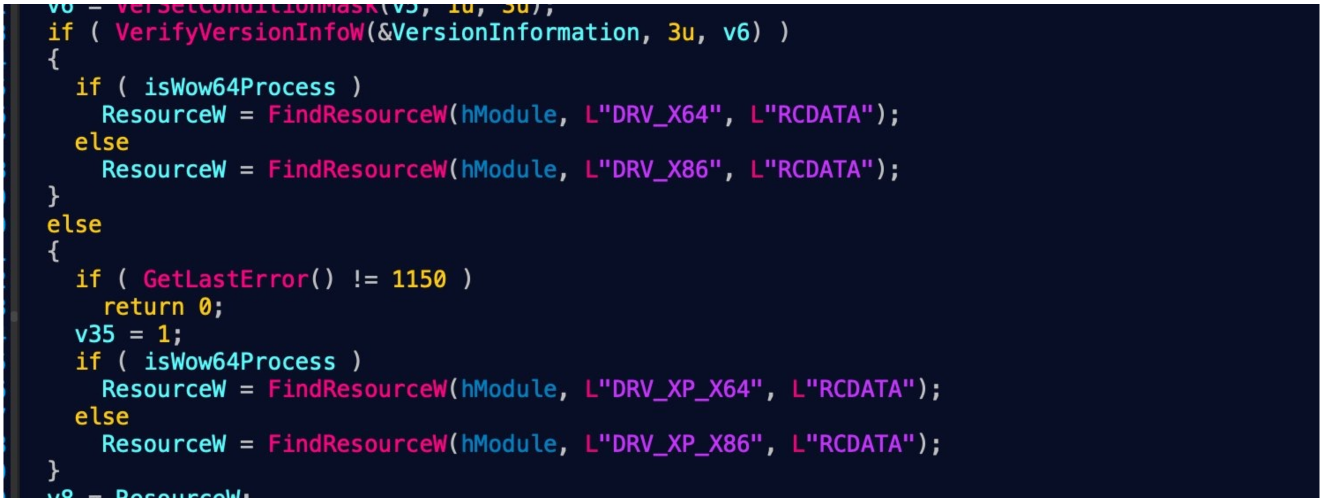
(Shamoon) took advantage of Eldos Rawdisk in order to get direct userland access to the filesystem without calling Windows APIs. HermeticWiper uses a similar technique by abusing a different driver, `empntdrv.sys`.



```
      Type              Size|     ID|Name
RCDATA                 11119       0 DRV_X64
RCDATA                  9904       0 DRV_X86
RCDATA                 10956       0 DRV_XP_X64
RCDATA                  9626       0 DRV_XP_X86
```

HermeticWiper resources containing EaseUS Partition Manager drivers

The copies of the driver are ms-compressed resources. The malware deploys one of these depending on the OS version, bitness, and SysWow64 redirection.



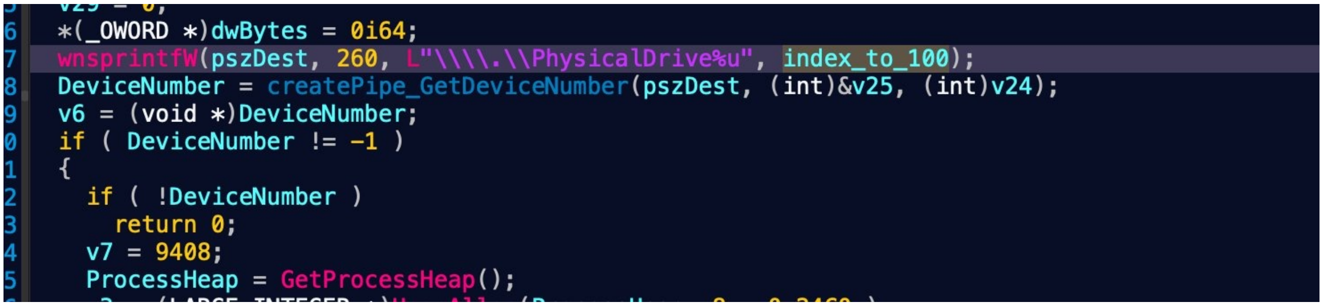
```
v6 = VerSetConditionMask(v5, 1u, 3u);
if ( VerifyVersionInfoW(&VersionInformation, 3u, v6) )
{
  if ( isWow64Process )
    ResourceW = FindResourceW(hModule, L"DRV_X64", L"RCDATA");
  else
    ResourceW = FindResourceW(hModule, L"DRV_X86", L"RCDATA");
}
else
{
  if ( GetLastError() != 1150 )
    return 0;
  v35 = 1;
  if ( isWow64Process )
    ResourceW = FindResourceW(hModule, L"DRV_XP_X64", L"RCDATA");
  else
    ResourceW = FindResourceW(hModule, L"DRV_XP_X86", L"RCDATA");
}
v8 = ResourceW;
```

EaseUS driver resource selection

The benign EaseUS driver is abused to do a fair share of the heavy-lifting when it comes to accessing Physical Drives directly as well as getting partition information. This adds to the difficulty of analyzing HermeticWiper, as a lot of functionality is deferred to `DeviceIoControl` calls with specific IOCTLs.

MBR and Partition Corruption

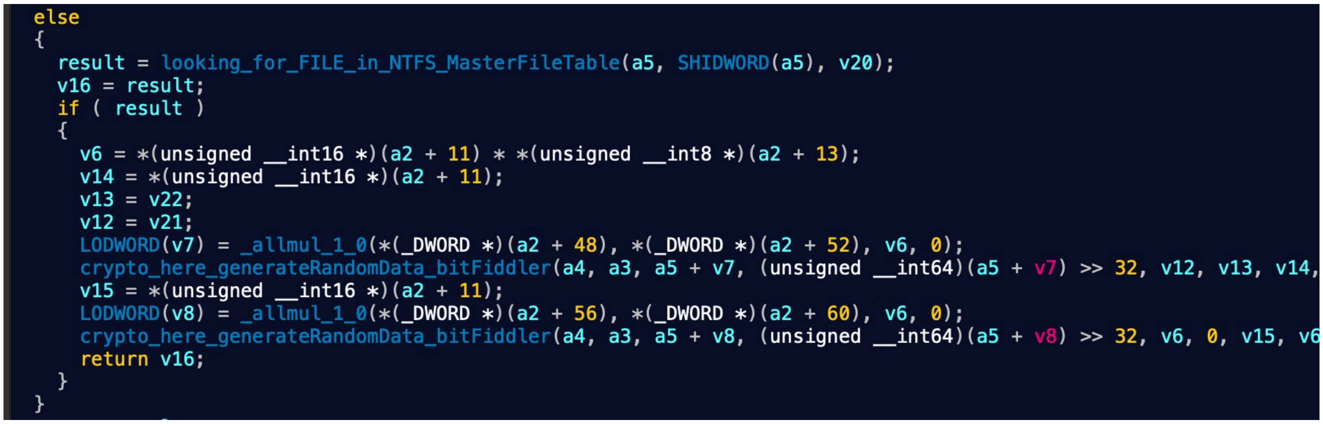
HermeticWiper enumerates a range of Physical Drives multiple times, from 0-100. For each Physical Drive, the `\\.\EPMNTDRV\` device is called for a device number.



```
*(_OWORD *)dwBytes = 0i64;
wnsprintfW(pszDest, 260, L"\\\\.\\PhysicalDrive%u", index_to_100);
DeviceNumber = createPipe_GetDeviceNumber(pszDest, (int)&v25, (int)v24);
v6 = (void *)DeviceNumber;
if ( DeviceNumber != -1 )
{
  if ( !DeviceNumber )
    return 0;
  v7 = 9408;
  ProcessHeap = GetProcessHeap();
```

The malware then focuses on corrupting the first 512 bytes, the Master Boot Record (MBR) for every Physical Drive. While that should be enough for the device not to boot again, HermeticWiper proceeds to enumerate the partitions for all possible drives.

They then differentiate between FAT and NTFS partitions. In the case of a FAT partition, the malware calls the same 'bit fiddler' to corrupt the partition. For NTFS, the HermeticWiper parses the Master File Table before calling this same bit fiddling function again.



```
else
{
  result = looking_for_FILE_in_NTFS_MasterFileTable(a5, SHIDWORD(a5), v20);
  v16 = result;
  if ( result )
  {
    v6 = *(unsigned __int16 *)(a2 + 11) * *(unsigned __int8 *)(a2 + 13);
    v14 = *(unsigned __int16 *)(a2 + 11);
    v13 = v22;
    v12 = v21;
    LODWORD(v7) = _allmul_1_0(*(_DWORD *)(a2 + 48), *(_DWORD *)(a2 + 52), v6, 0);
    crypto_here_generateRandomData_bitFiddler(a4, a3, a5 + v7, (unsigned __int64)(a5 + v7) >> 32, v12, v13, v14,
    v15 = *(unsigned __int16 *)(a2 + 11);
    LODWORD(v8) = _allmul_1_0(*(_DWORD *)(a2 + 56), *(_DWORD *)(a2 + 60), v6, 0);
    crypto_here_generateRandomData_bitFiddler(a4, a3, a5 + v8, (unsigned __int64)(a5 + v8) >> 32, v6, 0, v15, v6
    return v16;
  }
}
```

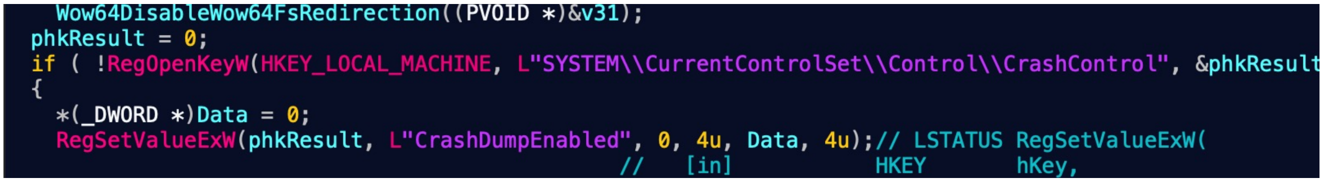
MFT parsing and bit fiddling calls

We euphemistically refer to the bit fiddling function in the interest of brevity. Looking through it, we see calls to Windows APIs to acquire a cryptographic context provider and generate random bytes. It's likely this is being used for an inlined crypto implementation and byte overwriting, but the mechanism isn't entirely clear at this time.

Further functionality refers to interesting MFT fields (`$bitmap` , `$logfile`) and NTFS streams (`$DATA` , `$I30` , `$INDEX_ALLOCATION`). The malware also enumerates common folders ('My Documents', 'Desktop', 'AppData'), makes references to the registry ('ntuser'), and Windows Event Logs (`"\\\\?\\C:\\Windows\\System32\\winevt\\Logs"`). Our analysis is ongoing to determine how this functionality is being used, but it is clear that having already corrupted the MBR and partitions for all drives, the victim system should be inoperable by this point of the execution.

Along the way, HermeticWiper's more mundane operations provide us with further IOCs to monitor for. These include the momentary creation of the abused driver as well as a system service. It also modifies several registry keys, including setting the

`SYSTEM\CurrentControlSet\Control\CrashControl CrashDumpEnabled` key to **0**, effectively disabling crash dumps before the abused driver's execution starts.



```
    Wow64DisableWow64FsRedirection((PVOID *)&v31);
  phkResult = 0;
  if ( !RegOpenKeyW(HKEY_LOCAL_MACHINE, L"SYSTEM\\CurrentControlSet\\Control\\CrashControl", &phkResult
  {
    *(_DWORD *)Data = 0;
    RegSetValueExW(phkResult, L"CrashDumpEnabled", 0, 4u, Data, 4u);// LSTATUS RegSetValueExW(
                                                 //    [in]          HKEY       hKey,
```

Disabling CrashDumps via the registry

Finally, the malware waits on sleeping threads before initiating a system shutdown, finalizing the malware's devastating effect.

Conclusion

After a week of defacements and increasing DDoS attacks, the proliferation of sabotage operations through wiper malware is an expected and regrettable escalation. At this time, we have a very small sliver of aperture into the attacks in Ukraine and subsequent spillover into neighboring countries and allies. If there's a silver lining to such a difficult situation, it's seeing the open collaboration between threat intel research teams, independent researchers, and journalists looking to get the story straight. Our thanks to the researchers at Symantec, ESET, Stairwell, and RedCanary among others who've contributed samples, time, and expertise.

SentinelOne Customers Protected

Watch Video At: https://youtu.be/keWfVA6F4IM

Indicators of Compromise

HermeticWiper	SHA1
Win32 EXE	912342f1c840a42f6b74132f8a7c4ffe7d40fb77
Win32 EXE	61b25d11392172e587d8da3045812a66c3385451

ms-compressed	SHA1
RCDATA_DRV_X64	a952e288a1ead66490b3275a807f52e5
RCDATA_DRV_X86	231b3385ac17e41c5bb1b1fcb59599c4
RCDATA_DRV_XP_X64	095a1678021b034903c85dd5acb447ad
RCDATA_DRV_XP_X86	eb845b7a16ed82bd248e395d9852f467

```
rule MAL_HERMETIC_WIPER {
    meta:
      desc = "HermeticWiper - broad hunting rule"
      author = "Friends @ SentinelLabs"
      version = "1.0"
      last_modified = "02.23.2022"
      hash = "1bc44eef75779e3ca1eefb8ff5a64807dbc942b1e4a2672d77b9f6928d292591"
    strings:
        $string1 = "DRV_XP_X64" wide ascii nocase
        $string2 = "EPMNTDRV\\%u" wide ascii nocase
        $string3 = "PhysicalDrive%u" wide ascii nocase
        $cert1 = "Hermetica Digital Ltd" wide ascii nocase
    condition:
      uint16(0) == 0x5A4D and
      all of them
}
```