Malware támadások hatékony kezelése (esettanulmányok)

Dr. Bencsáth Boldizsár
Ukatemi, BME Crysys Lab
2022.11.04.

Problem factors during a ransomware IR

- Lack of preparation in IR: no contract, no well-defined processes (playbooks), no tools, **panic**
- The whole network attacked by ransomware after hacking DC/AD
  - Including **backup server** which was part of the domain (authentication frame)
  - **SAN** also affected as login here also based on the same domain (DC/AD)
- Partners need to be informed about events. Guess what: **partner list** and contact details are on ransomware attacked servers
- Server contains 10+ years of not so important data, but it would be great to recover as there is not a single backup available (even not a 2 years old or such)
- Basic problems: Public RDP server, public VNC server, vulnerable VPN concentrator, vulnerable Firewall, lack of Windows updates, bad passwords

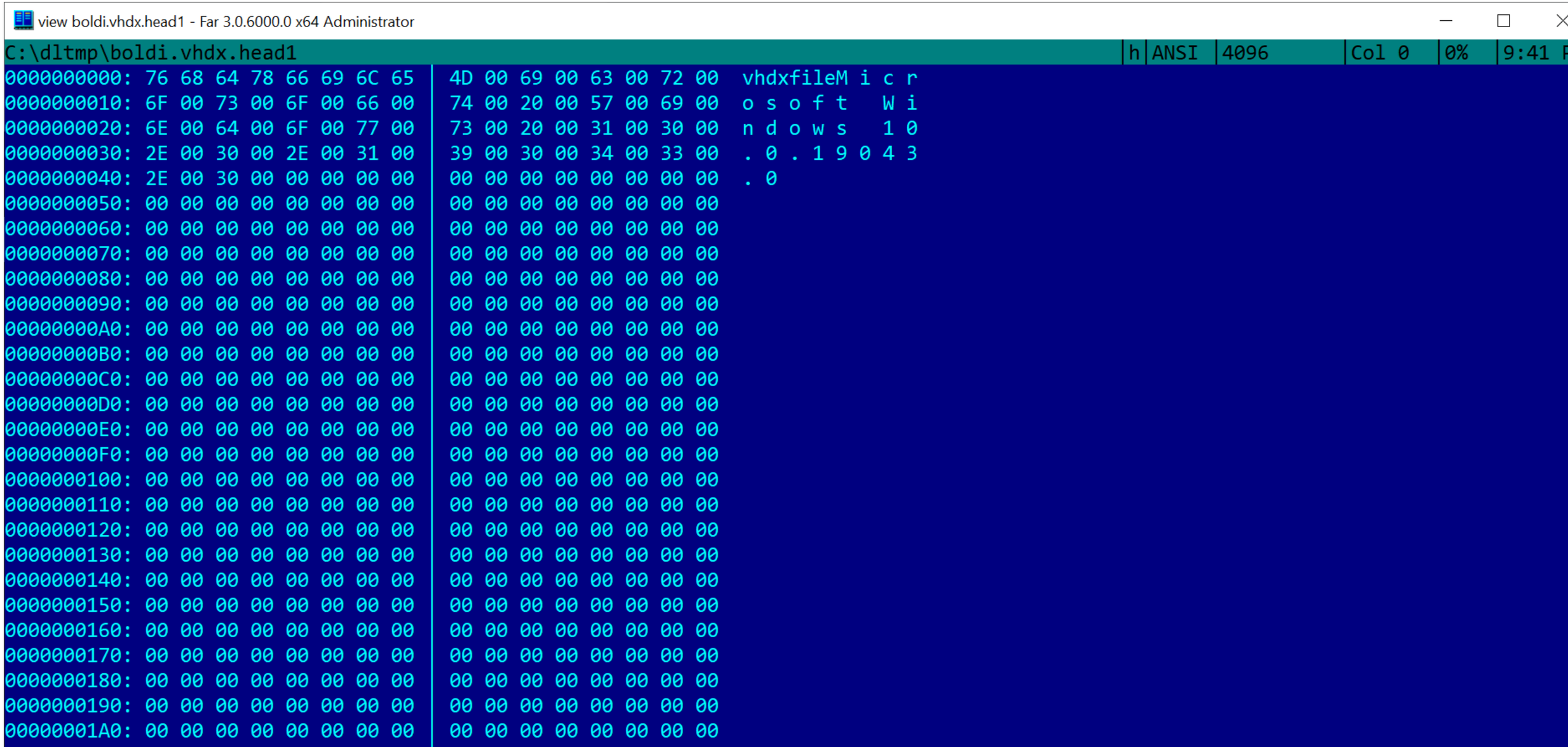
First 4k

- At one of our IR investigation, multiple, like tens of server virtual disk images were rendered unavailable by LockBit encryption
- First step: be calm, and do the IR process. Be professional, do not panic. We are there for help, not to pinpoint bad decisions.
- Okay, what is this lockbit, is it still inside the system (retrieve available related information immediately)
- Start the work: What is LockBit? Did they steal data, too? Are they still on our systems?
- Oh, they only encrypt first 4k to be fast
- Yeah, we have these important VM images encrypted first 4k
- Surely, the cannot be saved, first 4k is very very important (like boot sector, partition table, FS header and such should be there)
- Always cross-check things!

VHDX format

- The first 1 MB of the file contains 5 times 64 kb blocks.
- First block contains
- Signature (8 bytes): MUST be 0x766864786669 6C65, which is a UTF-8 string representing "vhdxfile". Creator (512 bytes): Contains a UTF-16 string that can be null terminated. This field is optional; the implementation fills it in during the creation of the VHDX file to identify, uniquely, the creator of the VHDX file. Implementation MUST NOT use this field as a mechanism to influence implementation behavior; it exists for diagnostic purposes only.

So… how does this look like?



Ransomware-as-a-Service Gang LockBit Pays First $50K Bounty

Group Thanked FBI Agent for Insider Information About Weaknesses

Prajeet Nair (@prajeetspeaks) • September 18, 2022

50k USD bug bounty?

- We found that for some VM images' first 4k bytes are not important 1,5 years ago
- Very easy to recover to valid info by dd for windows and a simple data file
- Within minutes full VM images can be fully recovered after a lockbit attack
- We did only share this information in small groups, trusted platforms to avoid ransomware developers to get know of it
- But some reckless people also found it and made money out of it by joining bug bounty platform of the ransomware creators
- We need basic morale

Things we could have sold, but…

- Duqu Oday windows bug
- Flame related stuff (windows update problem)
- FinFisher virtualization related reverse engineering efforts
- … LockBit lazy encryption on some virtualized HDD images

- Let's be clear: companies need to have clear moral grounds. Especially in this field of operation.

TLSH

- **TLSH = Trend Micro Locality Sensitive Hash**
- **TLSH is a hash function that preserves similarity**
  - **If two binaries A and B are similar, then their TLSH values are also similar**
- **TLSH difference**
  - **A metric that measures the difference between two TLSH values**
  - **Empirical observation:**

    If A and B are malware samples, then TLSHdiff(A, B) < 40 usually means that A and B are very similar samples (belong to the same malware family)

- **Performance merits**
  - **A TLSH value is only 35 bytes long**
  - **Computing TLSH values is very efficient**
  - **Computing TLSH differences is very efficient**

TLSH - example

- cp /bin/bash teszt.bin
- tlsh -f teszt.bin

T1FC455B07F6A314FEC5D6C8B0857B92B26831B4A5D1213D7B384CE6302F56F646B1EAE1 teszt.bin

- cp teszt.bin teszt2.bin
- echo "tesztelek" >>teszt2.bin
- tlsh -f teszt2.bin

T13D455B07F6A314FEC5D6C8B0857B92B26831B4A5D1213D7B384CE6302F56F646B1EAE1 teszt2.bin

TLSH in IR

- Suspicious **file** was found on a computer
- Nobody uploaded yet to **VirusTotal**
- You **won't upload** the file as it is sensitive (who knows what's inside)
- Want to know **what is this**
- Okay, let's find **similar samples** in malware repository
- Let's make a **yara rule** and run on all samples… it would take weeks… no go
- Let's calculate the TLSH value and look similar samples in **TLSH database**… it takes 20 minutes to find similar samples
- Retrieve similar samples and find them on VirusTotal … some samples are uploaded … **information** is available on them (virus scanner names, remarks, uploader information, etc.) – now we have the hint to take actions

Check out our blog

To read more about similar cases or other topics in cybersecurity, visit our website:

https://ukatemi.com/resources/

UKATEMI

DRIVEN BY CHALLENGES

Boldizsár Bencsáth, PhD, OSCP

+36309902317

bencsath@ukatemi.com

ISO 9001 apave Certification

ISO 27001 apave Certification

CrySyS

www.crysys.hu