Class 15

CSC 471 Modern Malware Analysis
Anti-virus Software, Dynamic Heuristic Analysis

Si Chen (schen@wcupa.edu)

What we've learned so far...

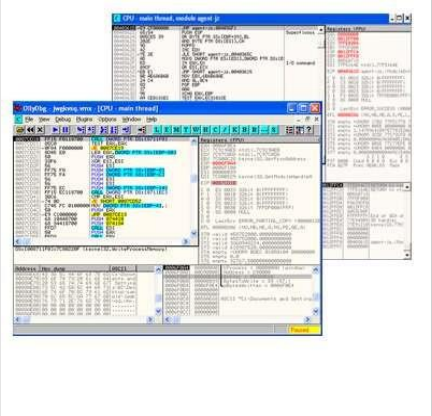
- Basic Analysis
  - Quickly glean information from the sample(s)
  - Help guide and focus Advanced Analysis

What we've learned so far...

- Advanced Analysis - Static
  - Used to see what is going on
  - Confirm suspicions aroused during basic analysis
  - Identify functionality
- Advanced Analysis - Dynamic
  - Control the program
    - Take new code paths
    - Change data

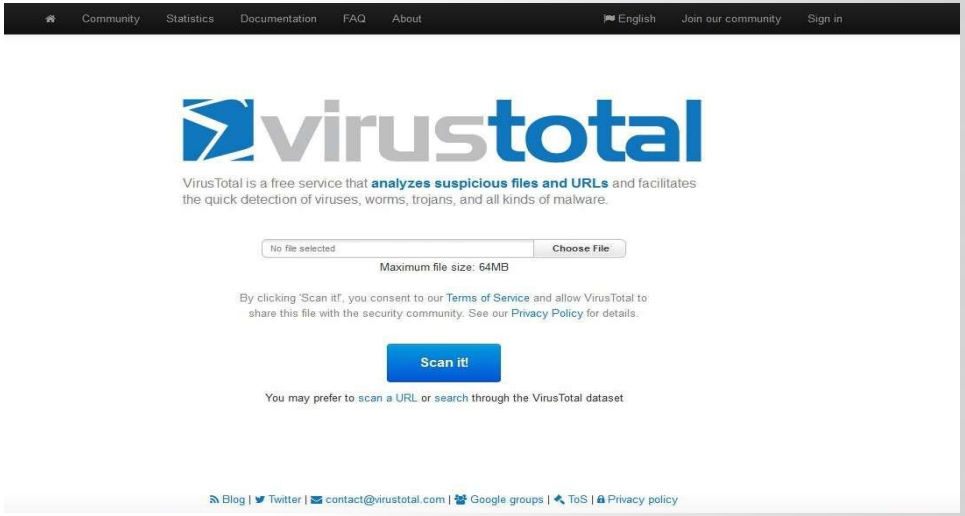
What we've learned so far...

- Windows API and systems
  - How does malware interact with Windows?
  - How does Windows make malware author's lives easier?
  - How does it make their lives harder?

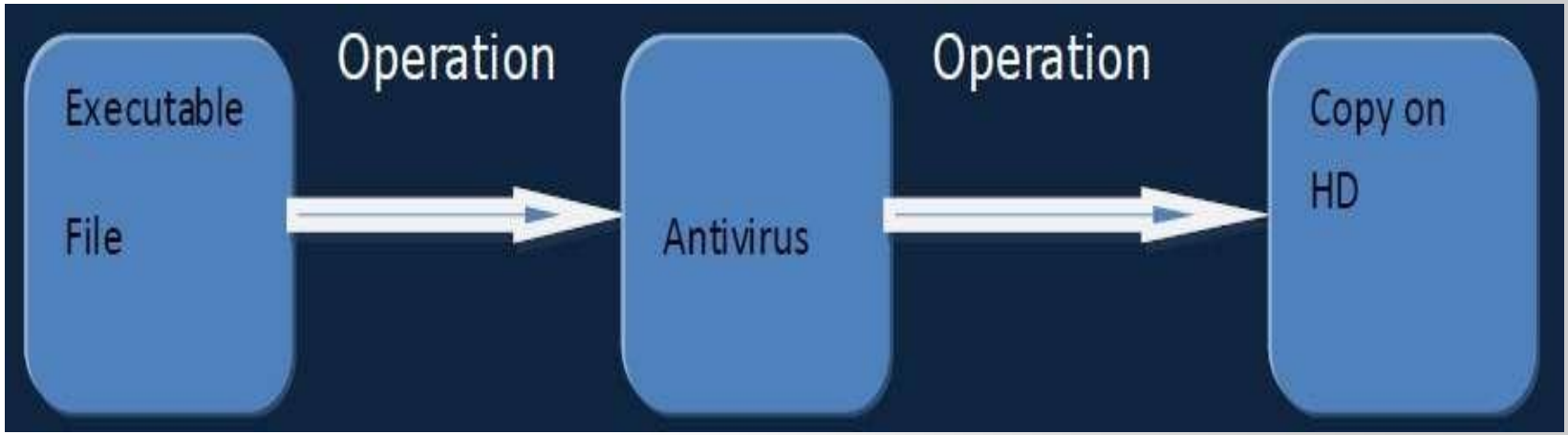


Anti-virus: How they actually work

- Nowadays AV scans our system on real-time basis.
- Information is analyzed based on the origin of the information
  - i.e. source of information.
- Operates differently depending upon source of information.



Anti-virus working from top level view.



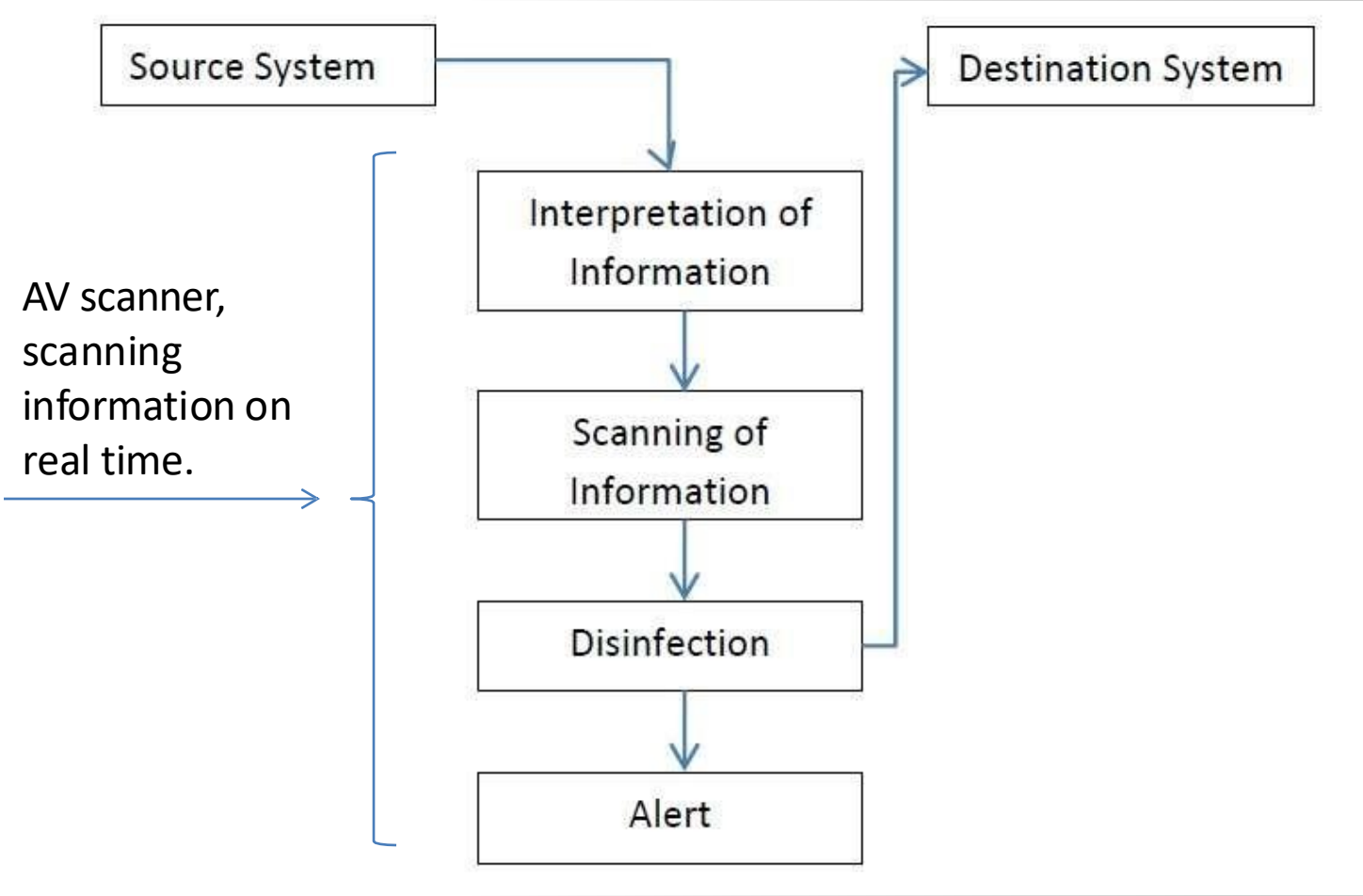
If the file is found malicious then the information will not be copied onto the destination location.
(Here destination in our case is HD)

One of the two possibilities takes place

- When the data is found to be legitimate, the scanner forwards that data to the destination location.

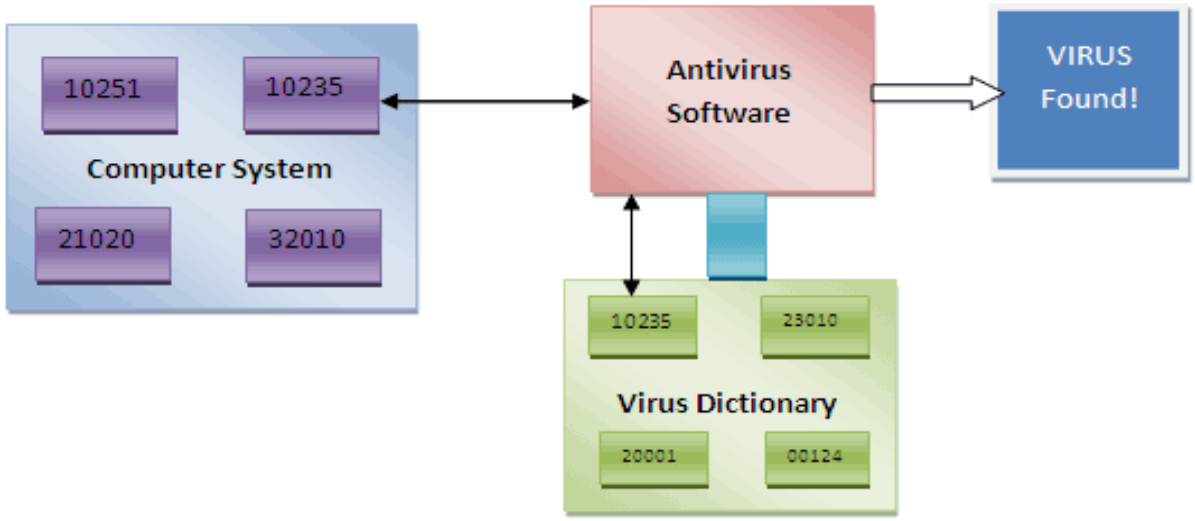
- When virus is detected then a warning is sent to UI for user`s action. Interface may vary.

Process flow of working of AV.



AV detection techniques(Scan - Engines)

- Signature Based detection (also sometimes called as “string based” detection)
- AV maintains a dictionary of the signatures of known Viruses, malwares, spywares etc.
- This dictionary is stored at client side and is usually in binary.
- Next-generation signature based detection
- Disadvantage?



Signature based Antivirus

AV detection techniques(Scan - Engines)

- Example:

  https://www.cs.wcupa.edu/schen/csc497/VirusShare_00001.md5

Heuristic based Detection

- Used to detect new, unknown viruses in your system that has not yet been identified.
- Based on the piece-by-piece examination of a virus.
- Looks for the sequence of instruction that differentiate the virus from 'normal programs'
- Disadvantage?
- Example:
  - Lab1

AV bypassing techniques

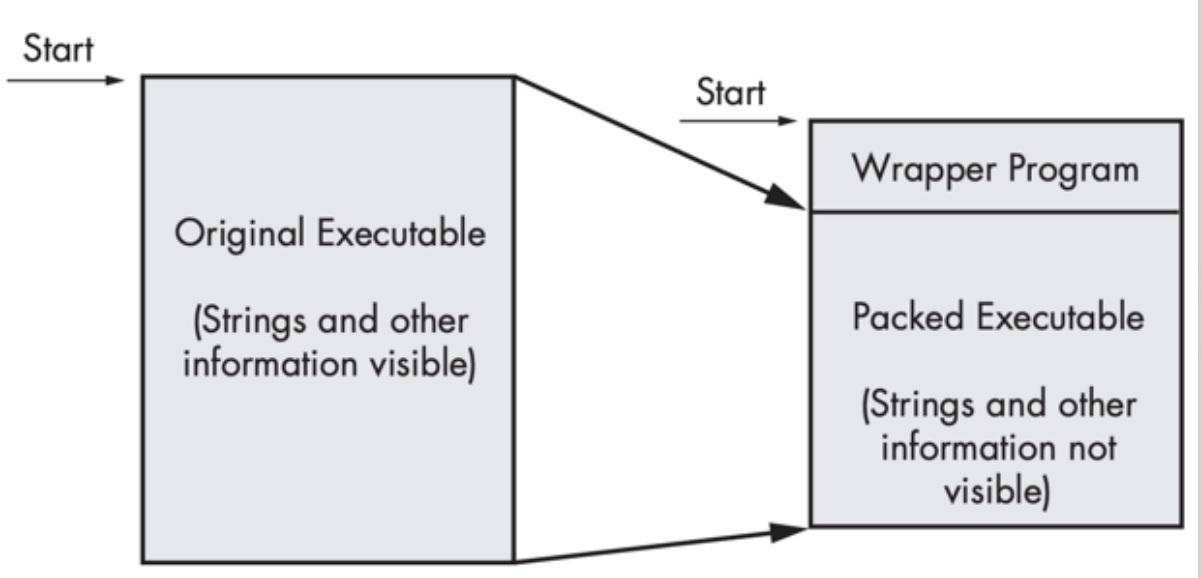
This are those techniques that the hackers and crackers already knew.

These are:

- Binders and packers
- Using splitter
- Code conversion from EXE to client side script
- Code obfuscation
- Using metasploit framework
- Code or DLL Injection

Packed and Obfuscated Malware

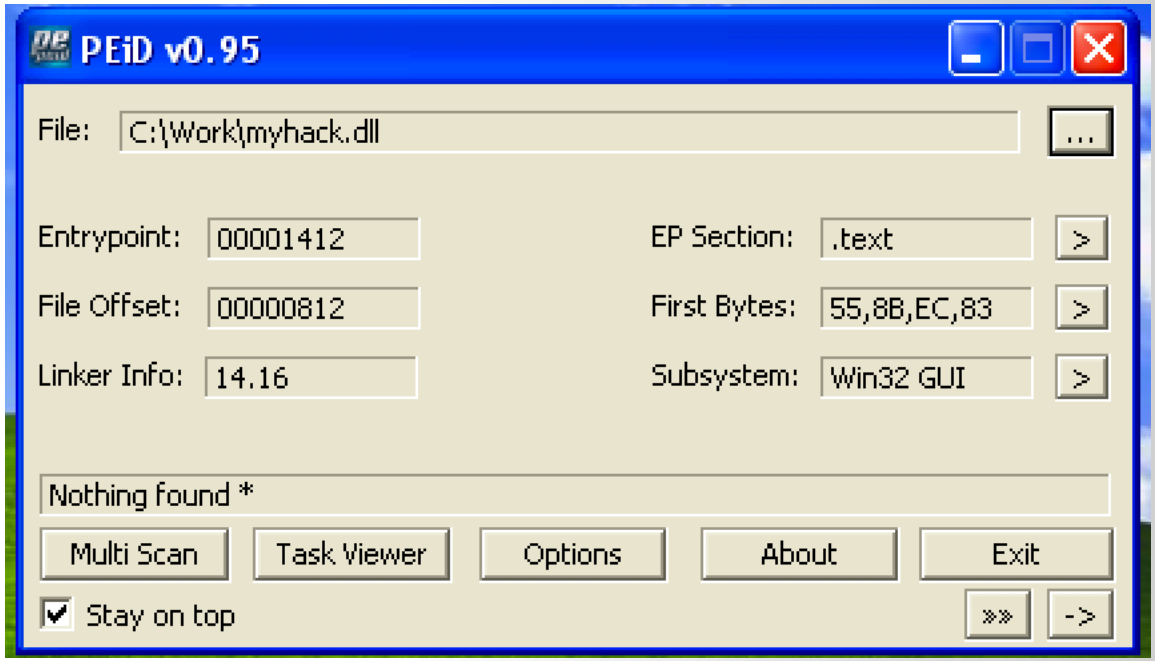
- Malware writers often use **packing or obfuscation** to make their files more difficult to detect or analyze.
- **Obfuscated** programs are ones whose execution the malware author has attempted to hide.
- **Packed** programs are a subset of obfuscated programs in which the malicious program is compressed and cannot be analyzed.
- Both techniques will severely limit your attempts to statically analyze the malware.



Packers and Cryptos

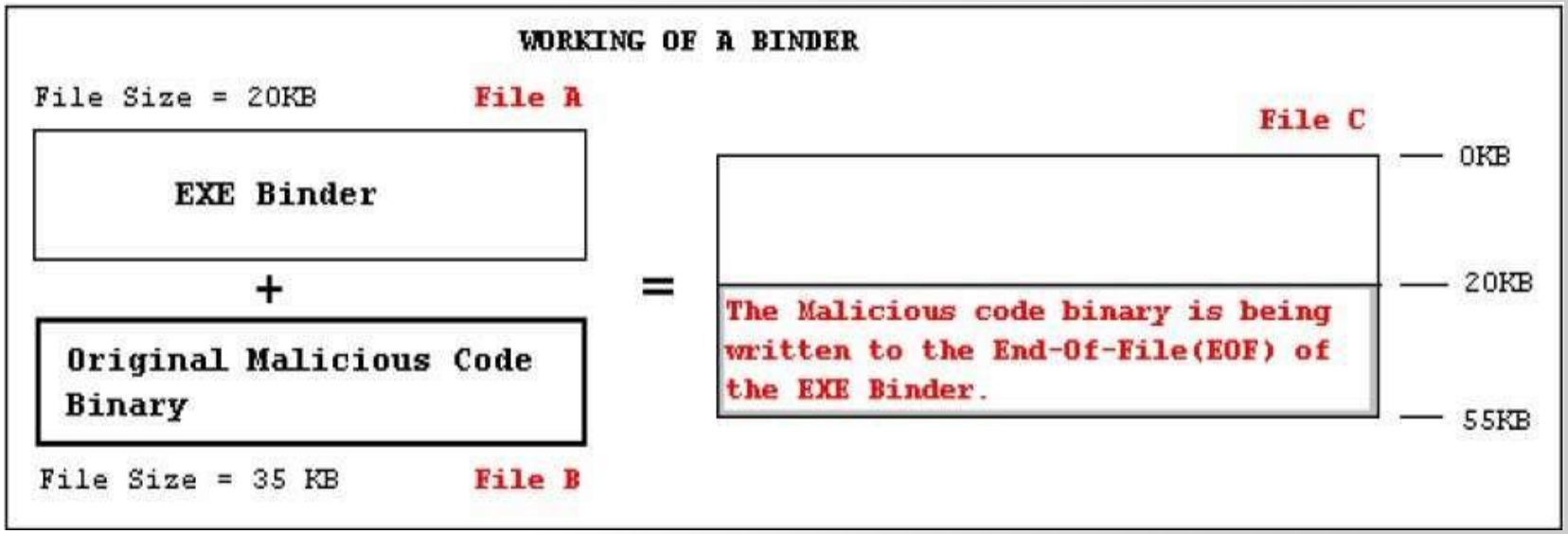
```
➜  ~ upx -o myhack_packed.dll myhack.dll
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     75264 ->     39424   52.38%    win32/pe     myhack_packed.dll

Packed 1 file.
```

Packed and Obfuscated Malware



Binders and Packers

- Binders



Splitting the File and Code Obfuscation

- These are those programs that split a single files into no. of small sized files.

- One may change some code into some small chunked file to evade AV detection and again join it and scan it to check whether AV flags it malicious or not. A trial and Error method..

Behavioral based detection

- Just observes how the program executes, rather than merely emulating its execution.
- Identify malware by looking for suspicious behavior.
- Disadvantage?

Sandboxing Based detection

- What is “sandbox” ?
- Isolate the files which are to be scanned and monitors their activity.

Heuristic Engines

- Heuristic engines are basically statistical and rule based analyze mechanisms.
- Their main purpose is detecting new generation(previously unknown) viruses by categorizing and giving threat/risk grades to code fragments according to predefined criteria.
- Heuristic engines are the most advanced part of AV products they use significant amount of rules and criteria.
- Since no anti virus company releases blueprints or documentation about their heuristic engines all known selective criteria's about their threat/risk grading policy are found with trial and error.

Dynamic Heuristic Analysis

Unknown Sample

PE file

Sandbox

contains C:\, D:\, E:\
And windows,
System32 Folder and
system file

Log File

monitoring the behavior
of the unknow sample,
logging the function
call, parameters, etc…

Malware Fingerprint

malware expert use the
log file to find the key
features and add it to
the malware database

Some of the known rules about threat grading

– Decryption loop detected

– Reads active computer name

– Reads the cryptographic machine GUID

– Contacts random domain names

– Reads the windows installation date

– Drops executable files

– Found potential IP address in binary memory

– Modifies proxy settings

– Installs hooks/patches the running process

– Injects into explorer

– Injects into remote process

– Queries process information

– Sets the process error mode to suppress error box

– Unusual entrophy

– Possibly checks for the presence of antivirus engine

– Monitors specific registry key for changes

Some of the known rules about threat grading

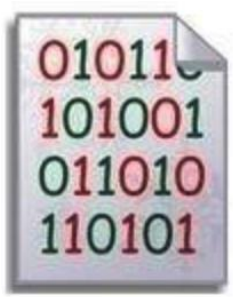
- Contains ability to elevate privileges
- Modifies software policy settings
- Reads the system/video BIOS version
- Endpoint in PE header is within an uncommon section
- Creates guarded memory regions
- Spawns a lot of processes
- Tries to sleep for a long time
- Unusual sections
- Reads windows product id
- Contains decryption loop
- Contains ability to start/interact device drivers
- Contains ability to block user input

	Pros	Cons
Static Heuristic Analysis	Fast, easy	Cannot handle shell, code obfuscation
Dynamic Heuristic Analysis	It can “reveal” the malware	May attacked by the anti-VM technology

Experiment

- In today's experiment, we want to monitoring the behavior of a ransomware, generate a log file. And analysis it.

Unknown Sample → **Sandbox** → **Log File** → **Malware Fingerprint**



Ransomware

- **Ransomware** is a type of malicious software, or malware, designed to deny access to a computer system or data until a ransom is paid.

Experiment

- Malware behavior(s):
  - Find all docx file in the current folder
  - change their filename to it's CRC32 (hash) value, and change the filename to crc32
  - Use the first 16 bytes to do XOR with 0x01 → to simulate file encryption
  - generate a txt file to ask for money (or bitcoin ☺)
  - delete itself → the best way to protect itself

Q & A