## Static Analysis Process

Static Analysis is reviewing the executable without running the code. We use tools to inspect areas of the executable in the hope to identify possible actions it will take on the environment.

The sample file is a fake Excel Document:

Graphical user interface, text, application, email

Description automatically generated

### **Stage 1:** File information.

This section attempts to determine what the file is and how it will be executed. Confirm the file information using TrID.

Graphical user interface, text, application, email

Description automatically generated

Results show a compressed file and the time to try to extract it. Switching to the tool: Exeinfo PE. This tool can review the file and tell us how to extract it.

Graphical user interface, text, application

Description automatically generated

Extracting the file with UPX.EXE

A screenshot of a computer

Description automatically generated with medium confidence

Now back in TrID, we can see the accurate information:

Graphical user interface, text, application, email

Description automatically generated

### **Stage 2:** View the information from the executable.

Here we attempt to find out how the file executes, libraries, integration points and other information—by opening the tool in PEStudio.

Graphical user interface, text, application, email

Description automatically generated

Here we can see the file has something in it. It's been created in Russia, has two functions, and is flagged in VirusTotal.

Graphical user interface, text, application

Description automatically generated

VirusTotal shows a massive number of triggers:

A screenshot of a computer

Description automatically generated with medium confidence

Functions are showing some network connections:

Graphical user interface, text, application

Description automatically generated

### **Stage 3:** Dumping the string.

In this stage, we dump all the strings out of the files. We are looking for known locations or commands in the file. Strings can be found in two ways; the first is via PEStudio:

PEStudio is reporting over 2000 strings. Looking at this, they appear to be obfuscated.

Graphical user interface, application, table

Description automatically generated with medium confidence

The other method uses BinText. Open the BinText Tool. This tool exports the string from the file as well.

Graphical user interface, text, application, email

Description automatically generated

Interesting points in the string dump show download location.

Graphical user interface, text, application, email

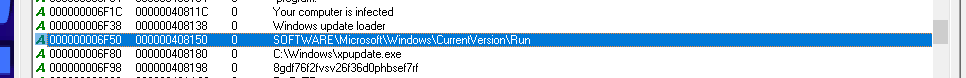
Description automatically generated

We can also see some HTML.

Graphical user interface, text, application, email

Description automatically generated

Here we can see it will load some values in the run registry key to keep control of the system.



The file is Malware. It's downloading from a site. The next step is to work out what it's downloading.

The last check is for encrypted strings with HTTP: This will use the tool xorsearch

Text

Description automatically generated

Found only the known HTTP address, no encrypted details.

### **Stage 4:** Identify the Hash.

All files can have a hash created; here, we dump the hash and check it with VirusTotal and other sites. This should provide some information on known Malware. If the hash is known, then we can follow the existing Analysis.

To work out the hash open the tool HashCalc.

Compress File Hash:

Graphical user interface, table

Description automatically generated with medium confidence

Uncompressed Hash:

Graphical user interface, application

Description automatically generated with medium confidence

Using the Hash on VirusTotal, search find the existing Malware.

Graphical user interface, application

Description automatically generated