Hiding Executable Code in Data Files

Samuel Šulovský

Faculty of Computer Science, University of Vienna

June 28, 2022

Outline

- What Are Files?
- Lazarus APT BMP RAT Attack
- Malware Recreation
- Evaluation, Discussion, Conclusions

What Are Files?

> *A file is a named collection of related information that is recorded on secondary storage. From a user's perspective, a file is the smallest allotment of logical secondary storage; that is, data cannot be written to secondary storage unless they are within a file.*

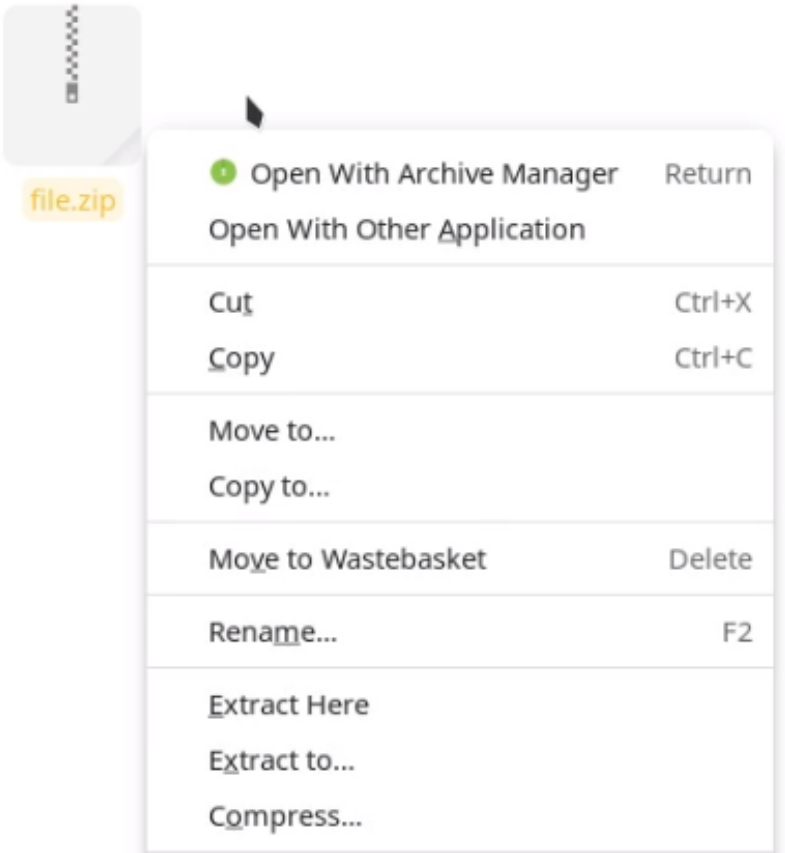
Silberschatz et al. *Operating System Concepts*

What Are Files?

- Files hold *arbitrary* data – binary, numeric, ASCII...
- Outward characteristics: file name and file *extension*
- Interpretation of a file's contents is always "guesswork"

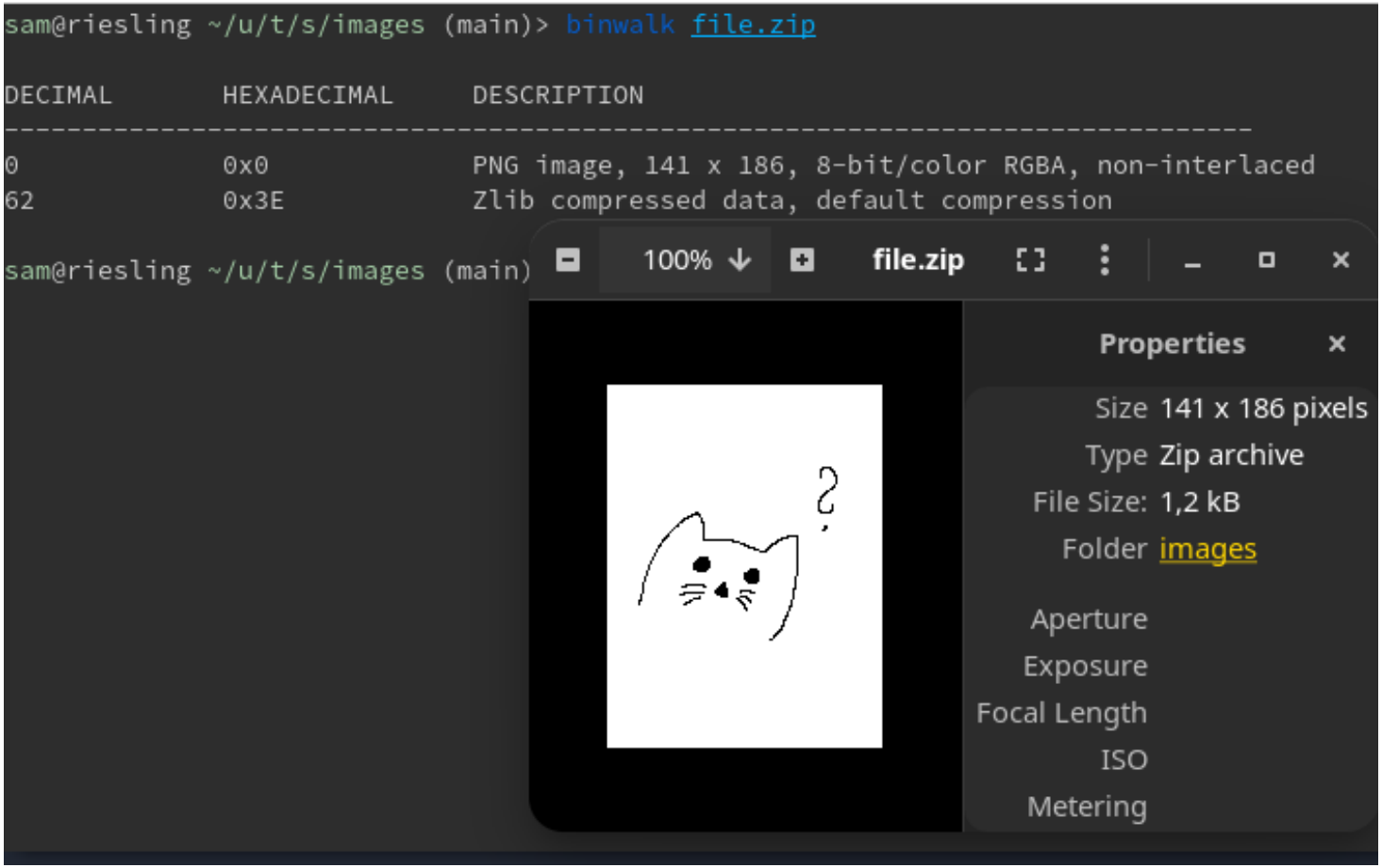
What Are Files?

- File extensions help us make a more accurate guess...



What Are Files?

- But they are ultimately arbitrary!



What Are Files?

- Trusting files to contain what they say they do can be a security risk
- Some file formats are more exploitable than others
- Studied example: the PNG file format

What Are Files?

> *The first eight bytes of a PNG file always contain the following (decimal) values: 137 80 78 71 13 10 26 10*
> *This signature indicates that the remainder of the file contains a single PNG image, consisting of a series of chunks beginning with an IHDR chunk and ending with an IEND chunk.*

W3C *PNG Specification*

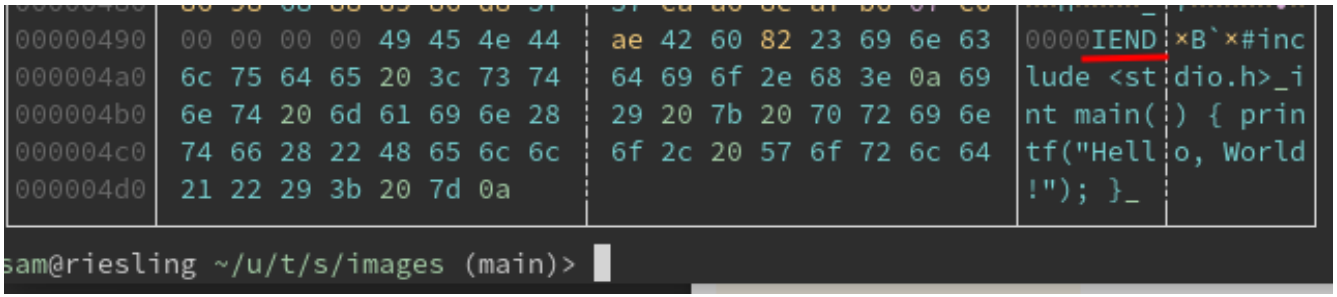
What Are Files?

- PNG data stream is terminated, all input after IEND is ignored
- Can we still append to the file? Sure!

```
sam@riesling ~/u/t/s/images (main)> binwalk image.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 141 x 186, 8-bit/color RGBA, non-interlaced
62            0x3E            Zlib compressed data, default compression

sam@riesling ~/u/t/s/images (main)> cat hello.c
#include <stdio.h>
int main() { printf("Hello, World!"); }
sam@riesling ~/u/t/s/images (main)> cat hello.c >> image.png
sam@riesling ~/u/t/s/images (main)> binwalk image.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 141 x 186, 8-bit/color RGBA, non-interlaced
62            0x3E            Zlib compressed data, default compression

sam@riesling ~/u/t/s/images (main)>
```

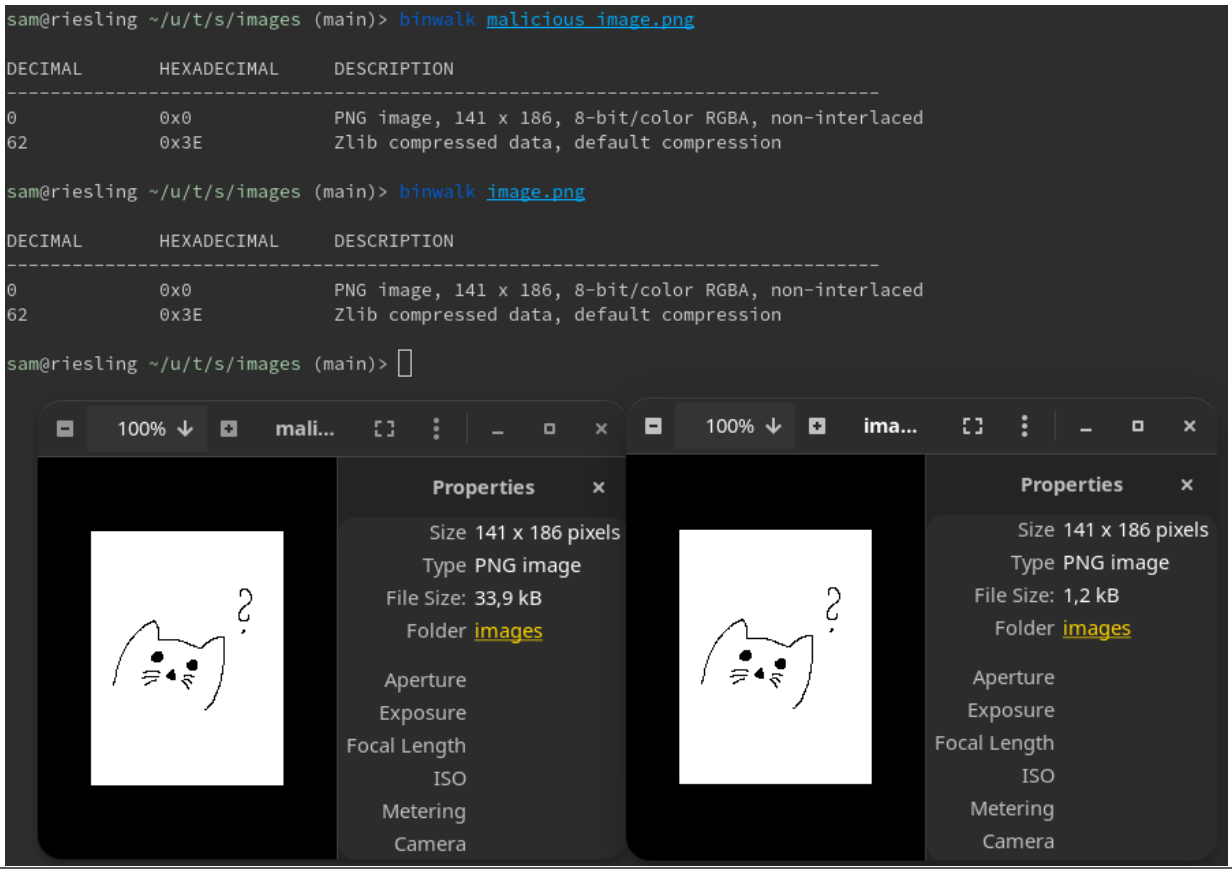
What Are Files?

- Can we make the data less visible in the PNG data stream?



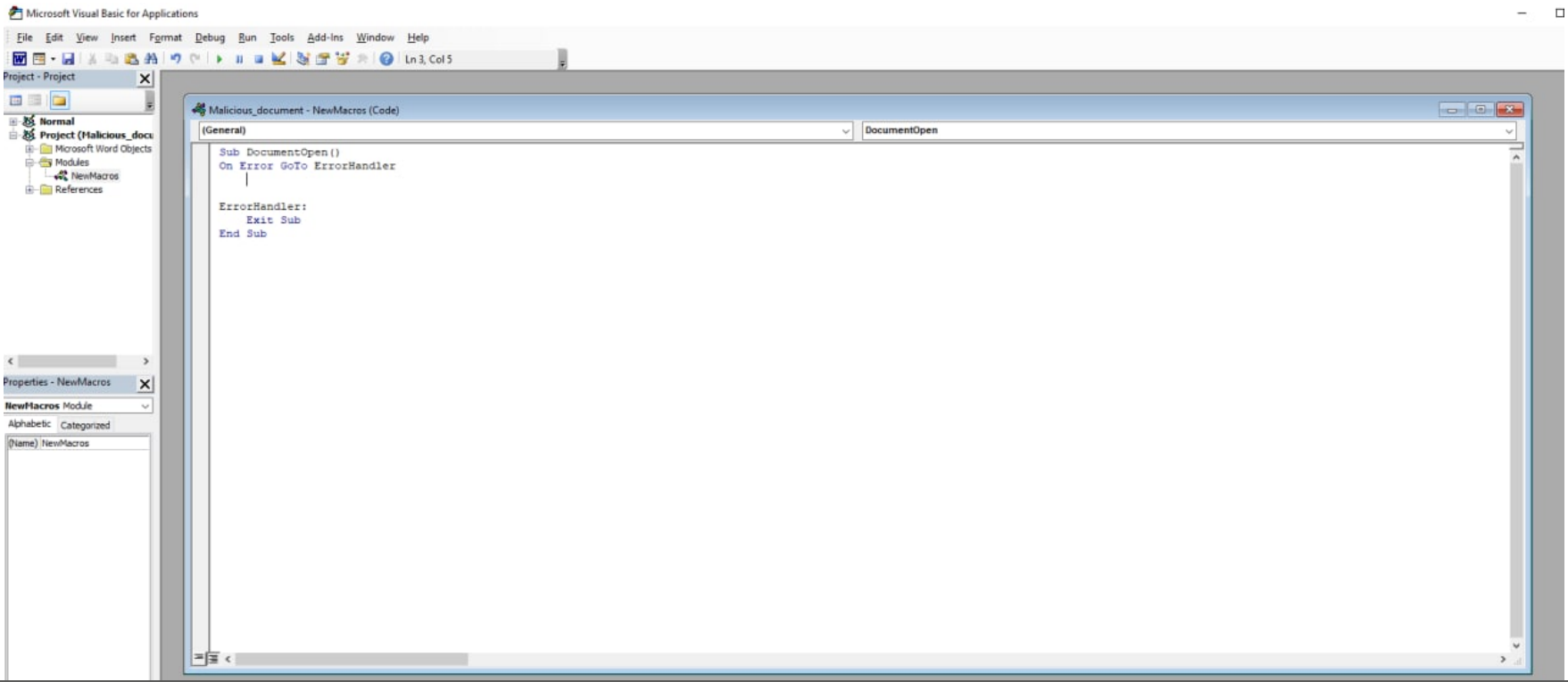
What Are Files?

- Idea: masquerade as part of the PNG data stream!
- Compress the data using the same compression as the PNG uses
- Move the IEND terminator after our new data stream



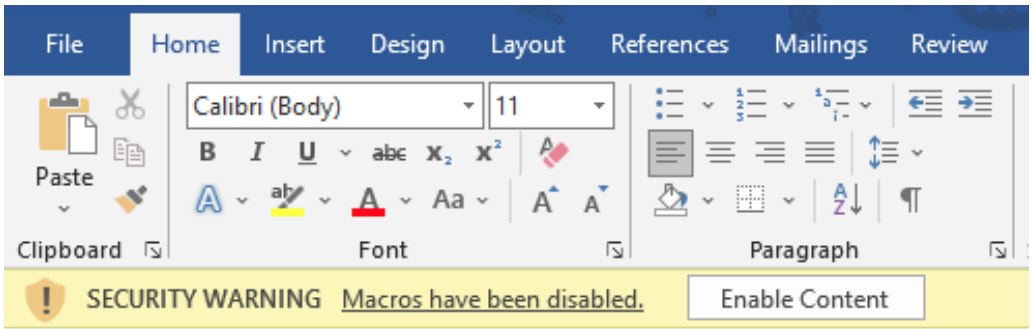
What Are Files?

- Easier way to hide executable code in data files: document formats
- Micorsoft Office Suite has *macro-enabled documents*
- Allows user to write code to be run from the document
- Very blatant security risk!



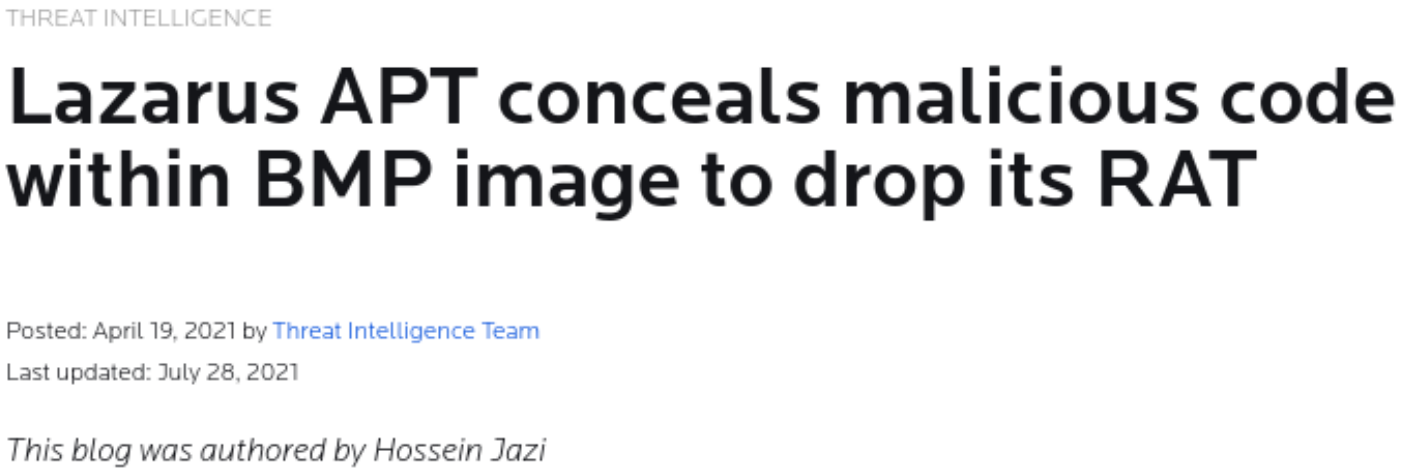
What Are Files?

- Macros are disabled by default, with a pop-up option to enable them
- "A design comedy of errors with tragic security consequences" - Gutfleisch et al. 2021
- Remain a prominent attack vector, usually used to drop a payload



Lazarus APT BMP RAT Attack

- We studied an attack thought to be perpetrated by Lazarus Group
- Thought to be a state-sponsored APT
- We based our work off of a wonderful postmortem by Hossein Jazi



THREAT INTELLIGENCE

Lazarus APT conceals malicious code within BMP image to drop its RAT

Posted: April 19, 2021 by Threat Intelligence Team

Last updated: July 28, 2021

This blog was authored by Hossein Jazi

Lazarus APT BMP RAT Attack

- Uses a Microsoft Word document as a payload dropper
- Payload contains a PNG file with data appended
- Data is mistakenly extracted by a WIA function (*WIA_ConvertImage*)
- Extracted data run using *mshta* to drop second stage payload
- Second stage payload (*AppStore.exe*) contains encrypted RAT, decrypted at runtime

Malware Recreation

- We recreated only the executable concealment and the droppers
- Demo time!

Evaluation, Discussion, Conclusions

- As is evident, the attack *no longer works*
- Why? No clear answer, most likely a patch to the faulty *WIA_ConvertImage* function
- We ran into some difficulties with this function, possibly leading to inaccurate results
- We reached out for comment and clarification to Malwarebytes, but haven't received an answer
- Image concealment mechanism not known, unable to precisely recreate binwalk signature

Evaluation, Discussion, Conclusions

- Image concealment mechanism not known, unable to precisely recreate binwalk signature
- Failure to extract data possibly due to a patch

```
                                $ binwalk image003.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 680 x 680, 8-bit/color RGB, non-interlaced
91            0x5B            Zlib compressed data, compressed

                                $ binwalk image003.zip

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PC bitmap, Windows 3.x format,, 680 x 680 x 24
54            0x36            HTML document header
1345309       0x14871D        HTML document footer
```

```
sam@riesling ~/u/t/i/src (main)> binwalk image.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 141 x 186, 8-bit/color RGBA, non-interlaced
62            0x3E            Zlib compressed data, default compression

sam@riesling ~/u/t/i/src (main)> binwalk image001.zip

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PC bitmap, Windows 3.x format,, 141 x 186 x 32

sam@riesling ~/u/t/i/src (main)>
```

Evaluation, Discussion, Conclusions

- Tested on Windows 10 (target OS for the attack)
- Microsoft Word versions: 2019, 365
- We believe modern systems are *no longer vulnerable* to this attack
- Vulnerability is likely to have been stealthily patched

Thank you for your attention!