Using image similarity algorithms on application icons

to discover new malware families on multiple platforms

VB 2014, Seattle

Martin Šmarda
Pavel Šrámek

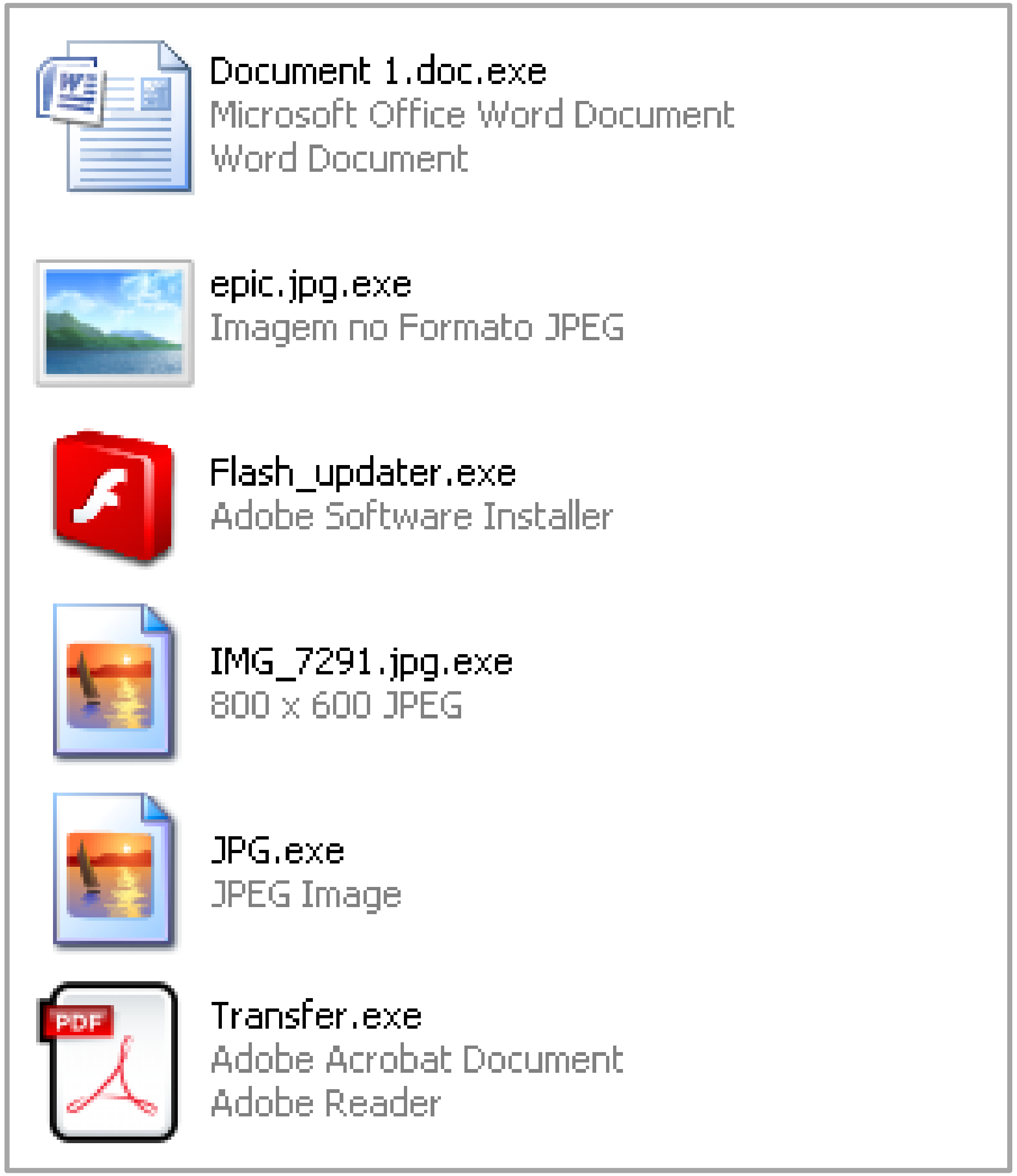
Motivation

Humans perceive the world in images.

1973: Xerox Alto
First viable GUI-based computer

Also the first system where one could **click an icon**.

Motivation



Document 1.doc.exe
Microsoft Office Word Document
Word Document

epic.jpg.exe
Imagem no Formato JPEG

Flash_updater.exe
Adobe Software Installer

IMG_7291.jpg.exe
800 x 600 JPEG

JPG.exe
JPEG Image

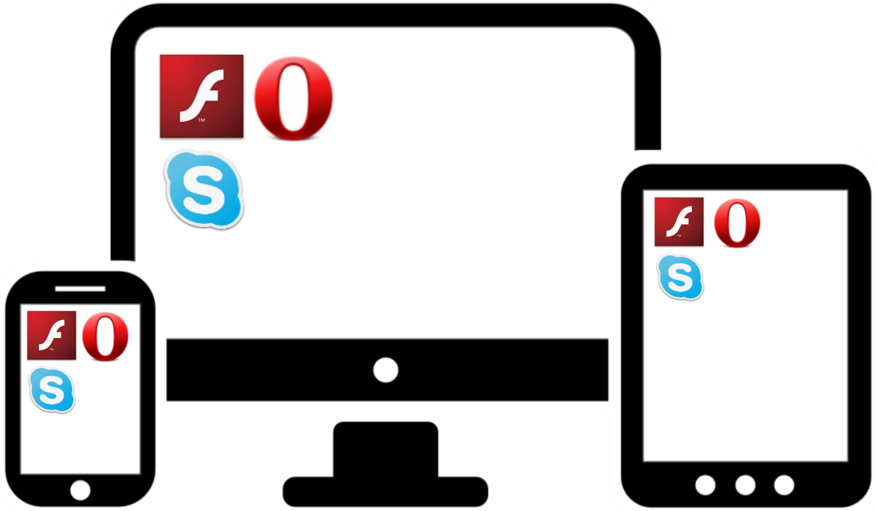
Transfer.exe
Adobe Acrobat Document
Adobe Reader

The bad guys know the importance of images.

2000–2014:
Windows malware

social engineering via
icon-based masquerading

Motivation



New platforms have emerged.

They all use icons
to represent apps.

The concept is **ubiquitous**,
and so is malware abusing it.

In the wild

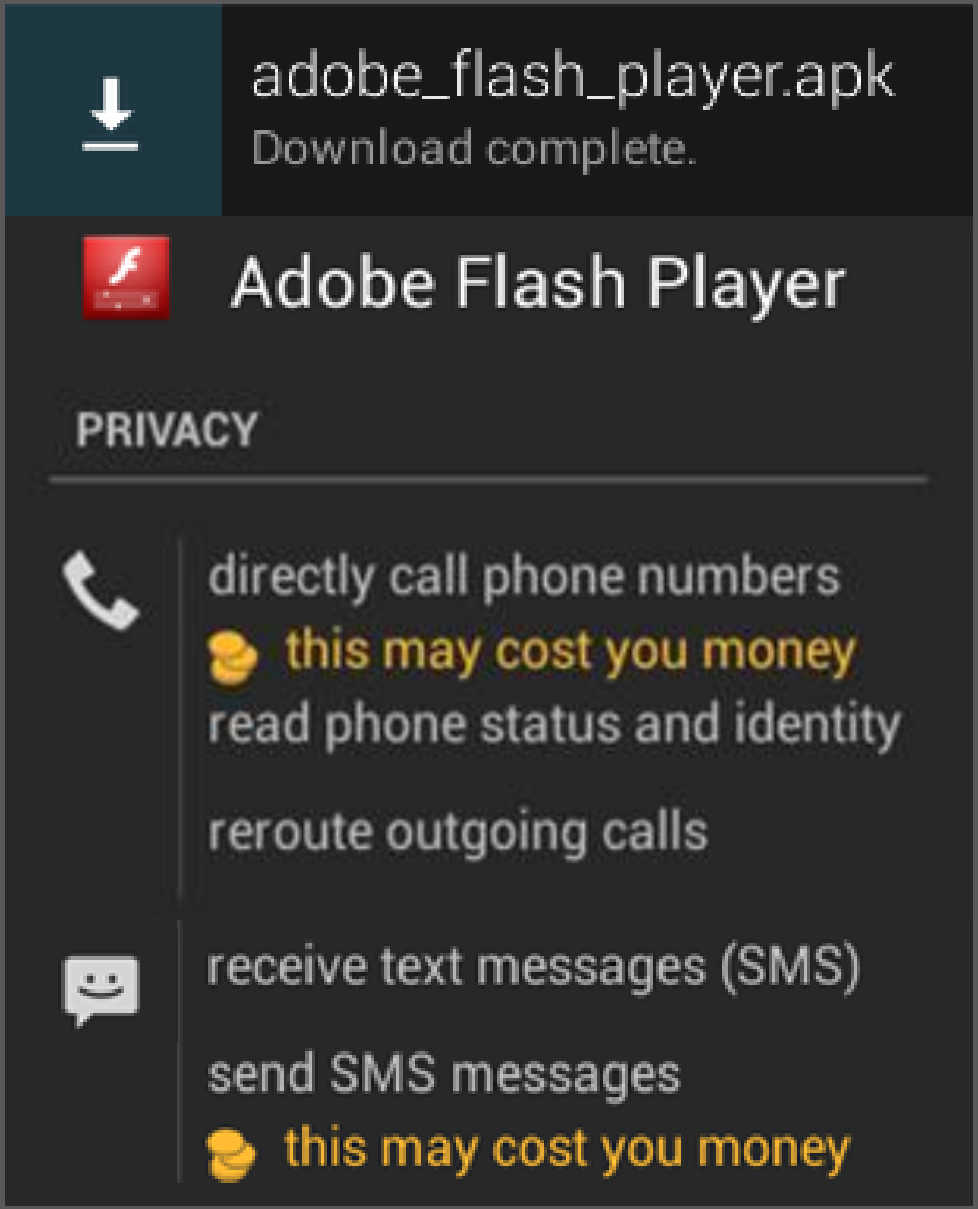
Android:FakeKRB

Win32:Vobfus

Android:OpFake

Win32:Hesperbot

In the wild



adobe_flash_player.apk
Download complete.

Adobe Flash Player

PRIVACY

directly call phone numbers
this may cost you money
read phone status and identity

reroute outgoing calls

receive text messages (SMS)

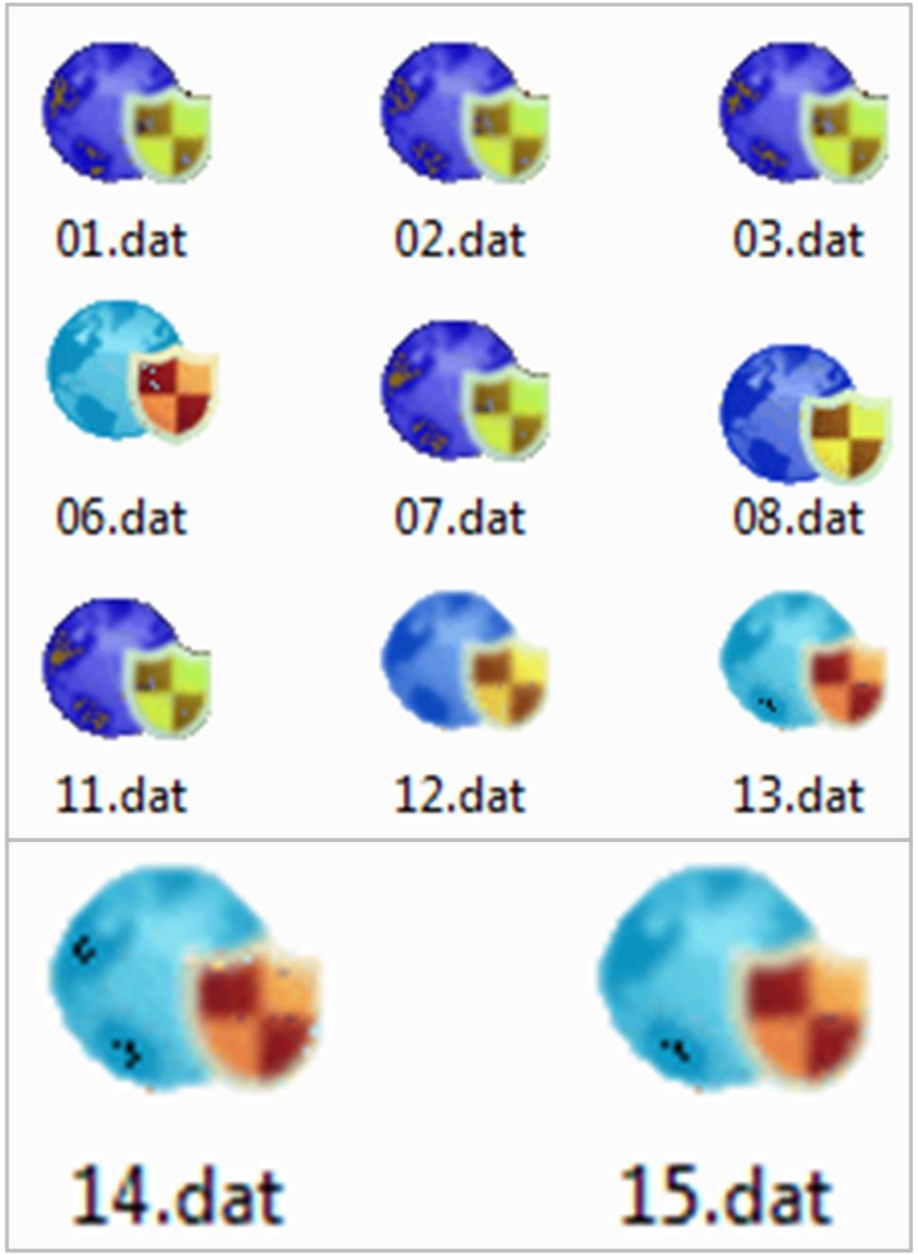
send SMS messages
this may cost you money

Social engineering elements

- Well-known target (Adobe Flash)
- Forged name and metadata (`com.adobeflashplayer`)
- Fake (or stolen) icon

→ Users can be fooled

Randomization



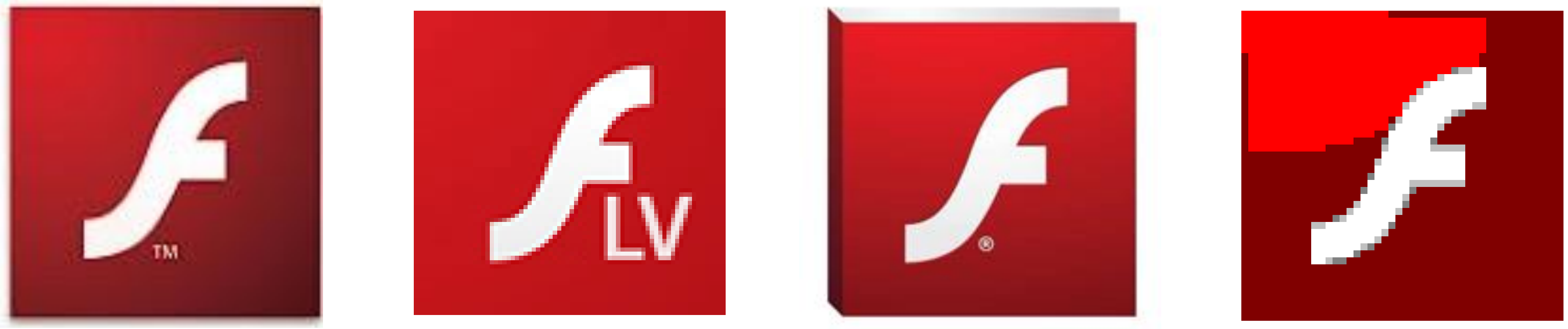
Malware authors already know the icon may be a weakness.

They are using **randomized icons**.

Approach

How to catch the bad guys by the icon?

Teach the computer to recognize **visual similarity** among icons.



Approach

> *Seems like a terrible way to do detection (alone). Umm… am I missing something?*
>
> -anonymous

Yes.

- This is not a *detection* engine
- This will not work alone

Hurdles

How to make it work?

for different platforms
even for damaged files

for different image formats

fast enough for deployment,
lightweight enough for storage

for unusual images
transparent, solid colored

Hurdles

How to make it work?

for different platforms
even for damaged files

multiple extracting...

for different image formats

...and decoding algorithms

fast enough for deployment,
lightweight enough for storage

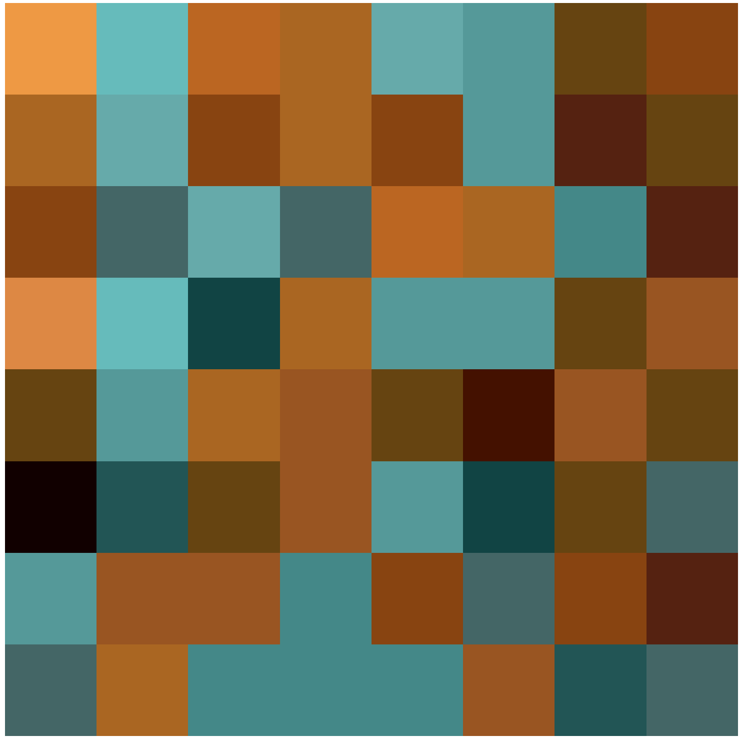
statistical approach
based on freq. transform.

for unusual images
transparent, solid colored

clever image preprocessing

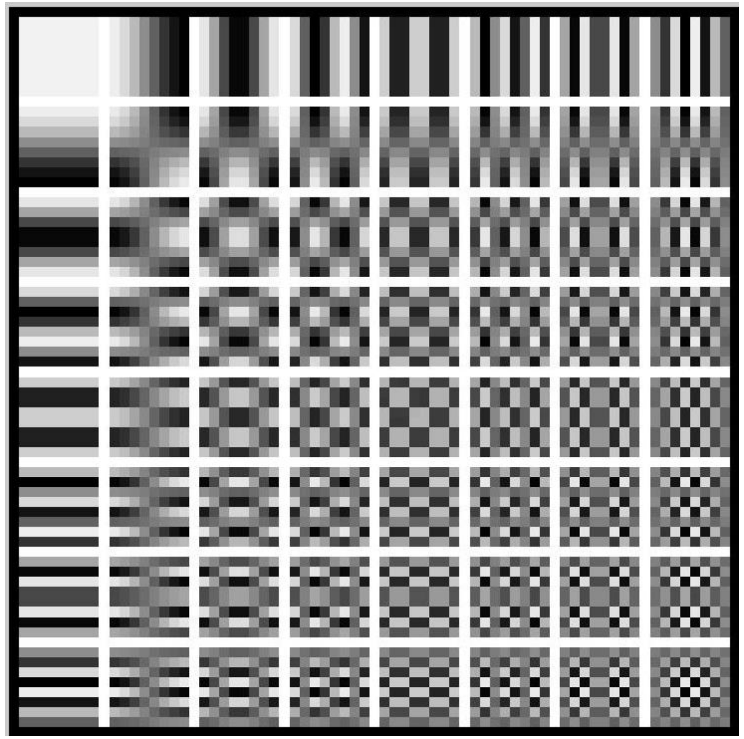
Frequencies

DCT ( )

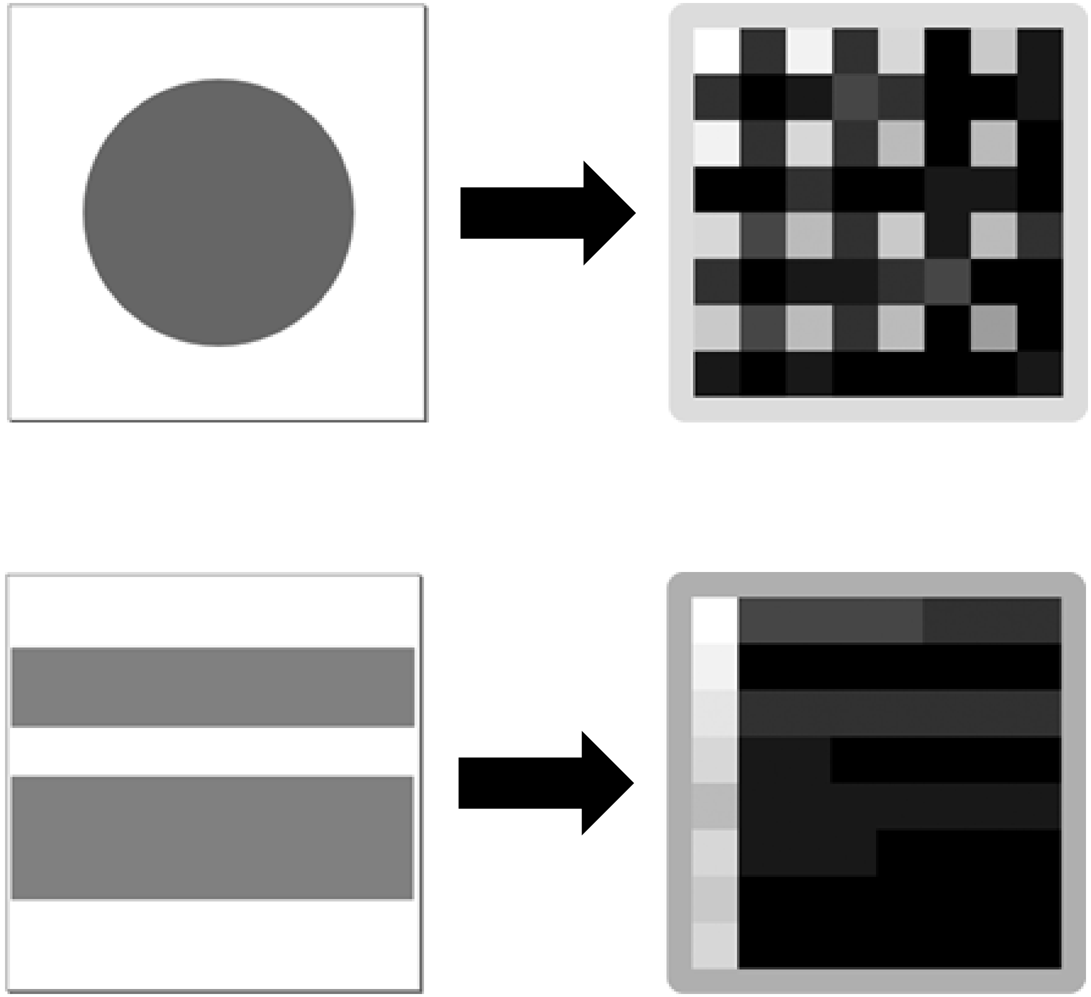


Discrete Cosine Transformation produces a matrix.

Coefficients mean frequencies, stay similar for similar images.



Frequencies



What does the computer see?

Specific shapes generate specific frequency imprints.

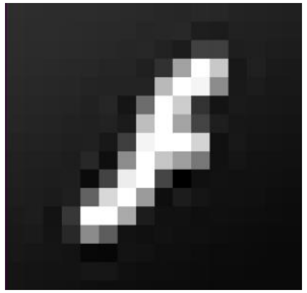
Edge cases must be accounted for.

Similarity algorithm

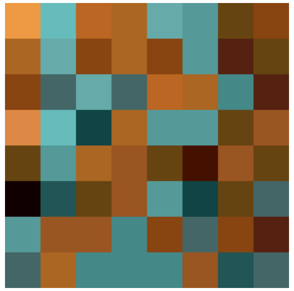
Extraction, decoding

Trim, grayscale

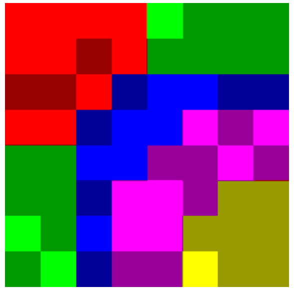
Blur, contrast



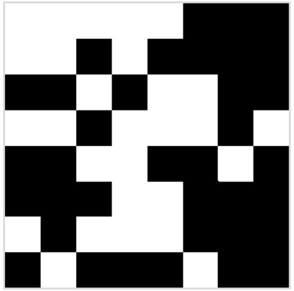
Resample



Freq. transform.



Data harvesting

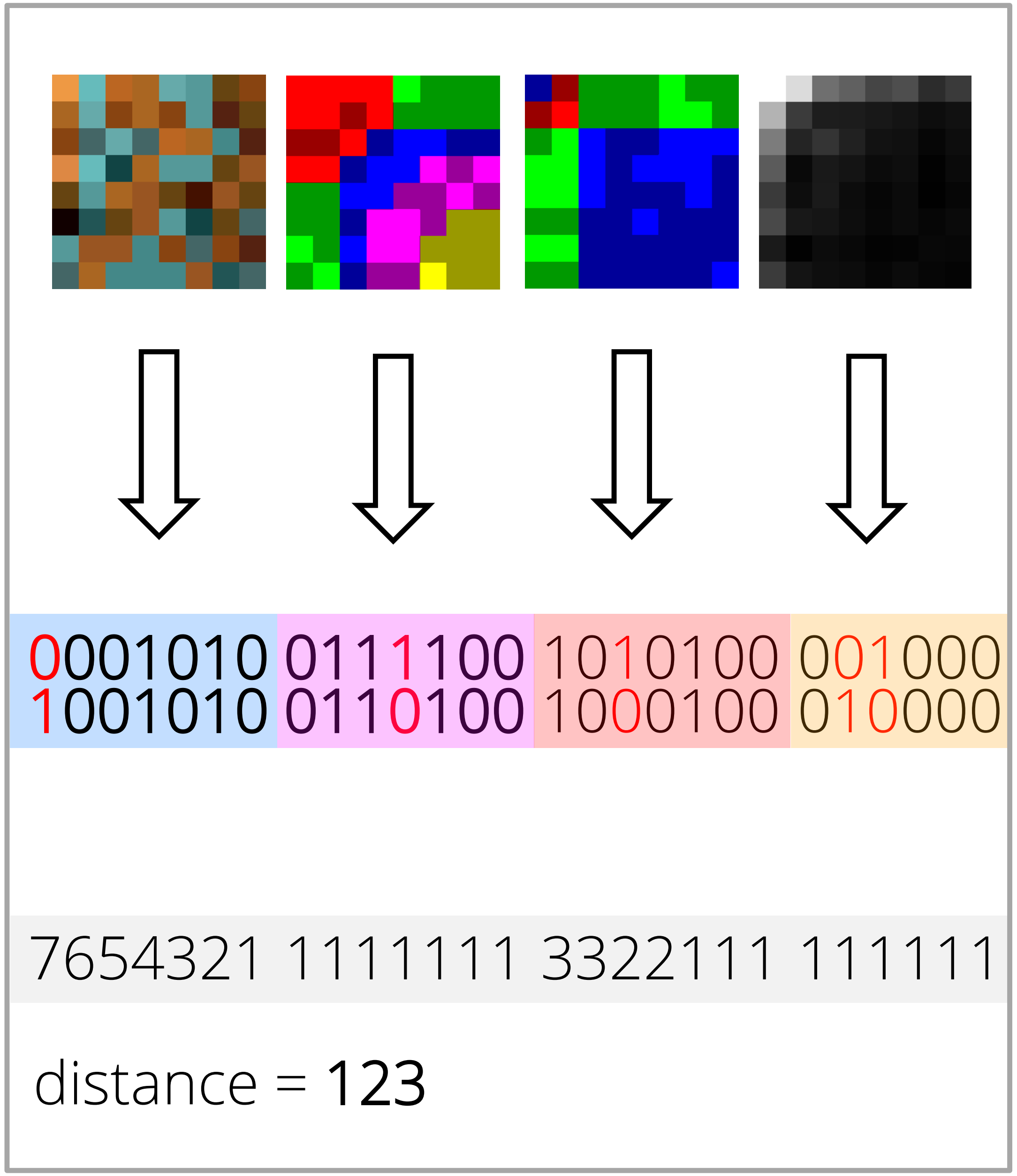


Hash encoding

0011
1010

Distance comp.

Tunability



0001010 0111100 1010100 001000
1001010 0110100 1000100 010000

7654321 1111111 3322111 111111

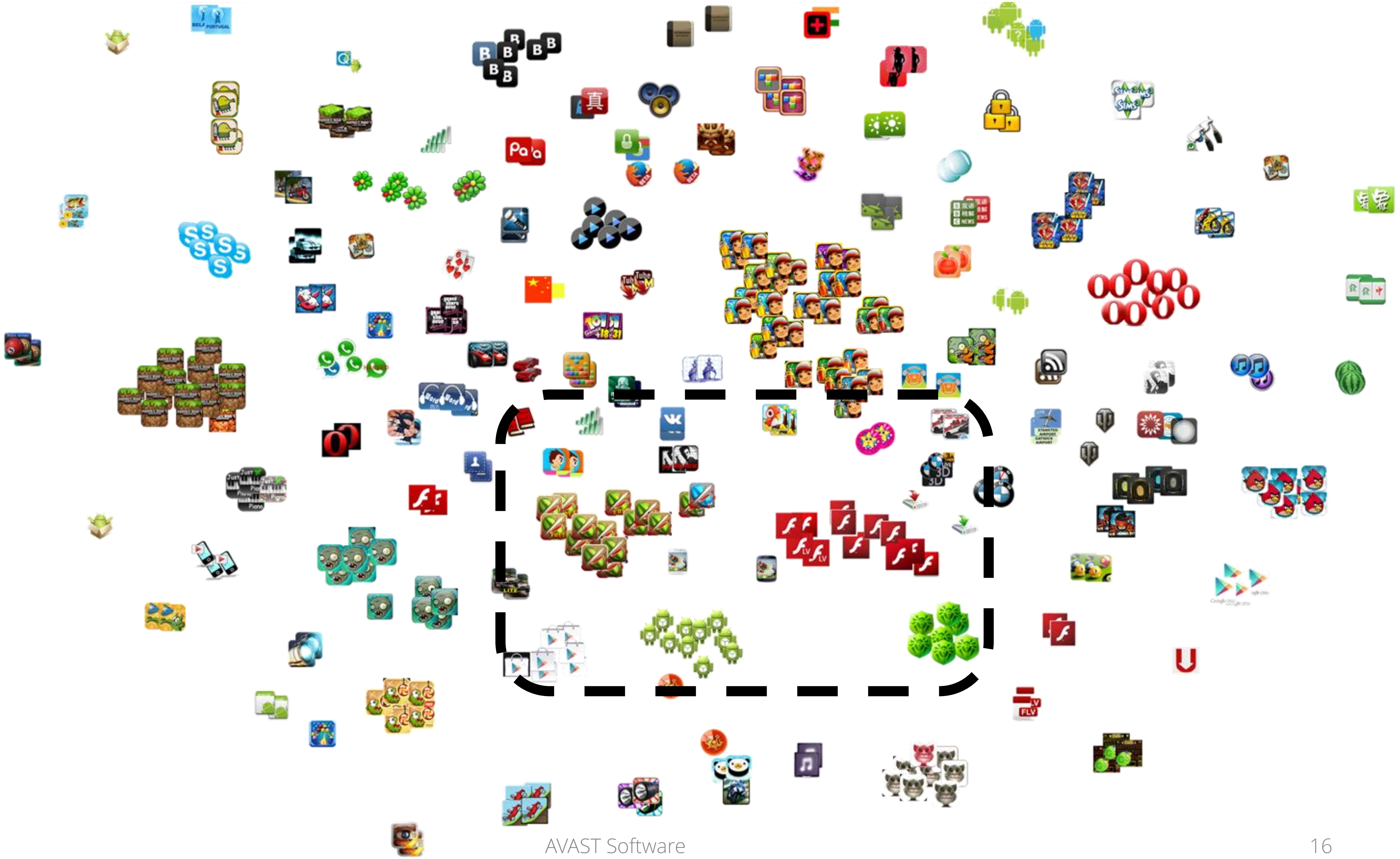
distance = 123

We can tune our algorithm to icon-specific features.

Multiple components extracted from DCT.
Processed per freq. zone.

Weighted Hamming distance
→ **comparison-time weights**

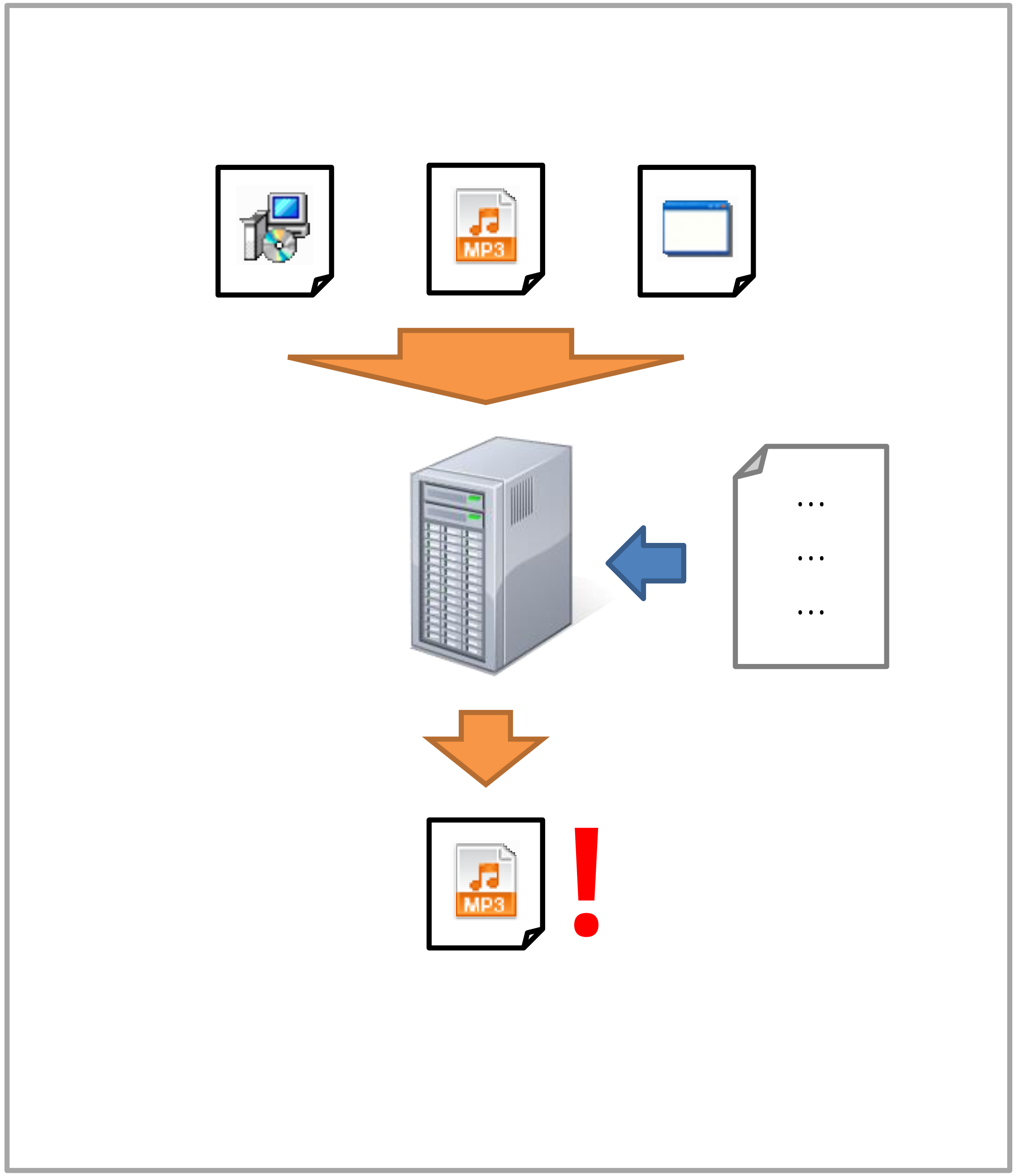
Grouping capabilities



Grouping capabilities

FREE
MAX PAYNE
LITE
LV
3D
LIVE
3A
BMW
MOST WANTED
EA

Deployment

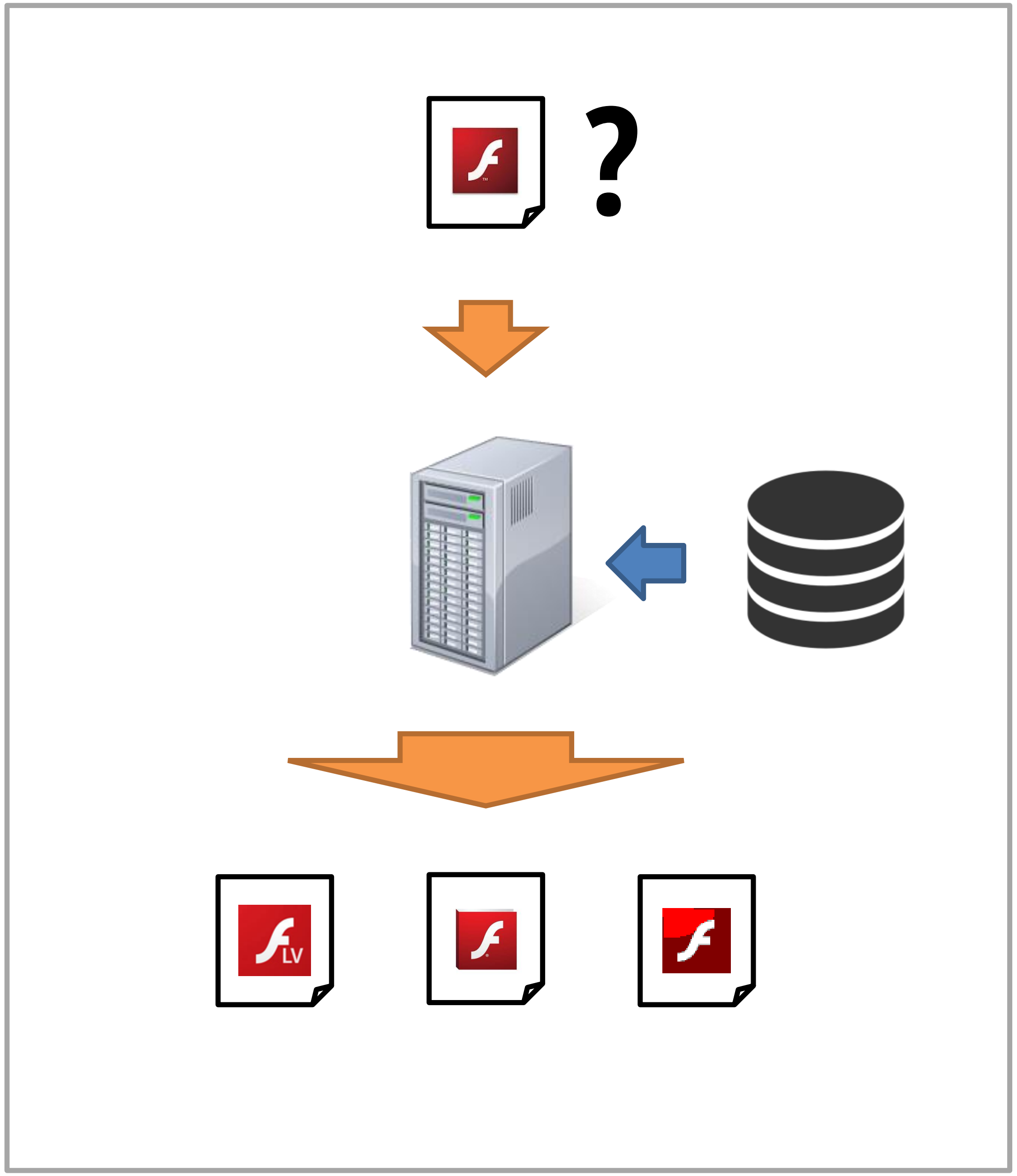


Deployed on the back-end. Incoming samples processed on-the-fly.

Hashes close to predefined list classified as suspicious.

+ other metadata → **auto or manual detection.**

Deployment

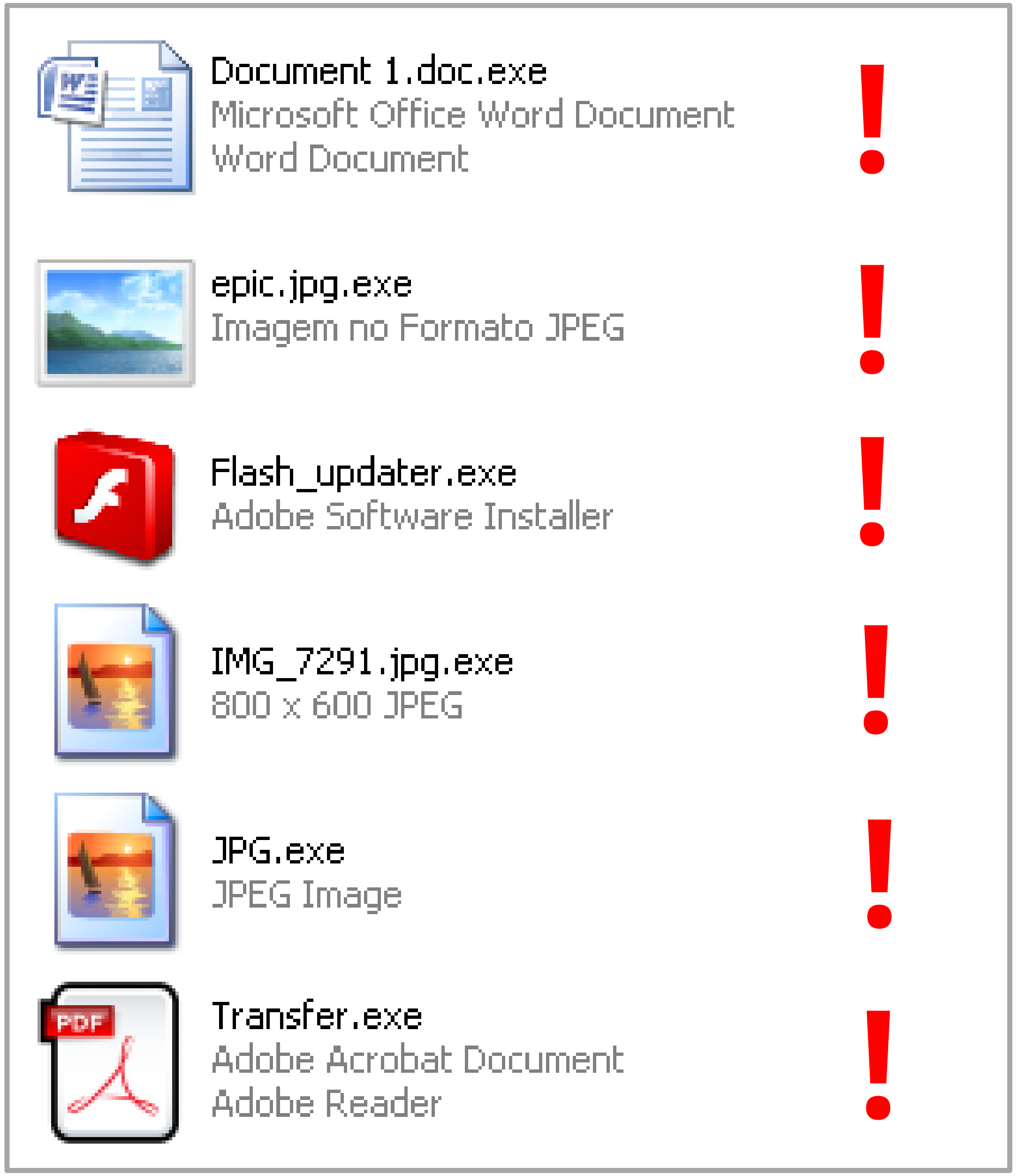


All samples with icons are indexed.
Regardless of platform.

Computed hashes are stored in a DB

Comparison is fast → **global icon-based search**.

Final thoughts



Document 1.doc.exe
Microsoft Office Word Document
Word Document
!

epic.jpg.exe
Imagem no Formato JPEG
!

Flash_updater.exe
Adobe Software Installer
!

IMG_7291.jpg.exe
800 x 600 JPEG
!

JPG.exe
JPEG Image
!

Transfer.exe
Adobe Acrobat Document
Adobe Reader
!

Icon familiarity is a vital part of socially engineered attacks.

Icons considered similar by humans look similar to our algorithm.

Attacks like these will be **automatically suspicious**.

That’s all, folks.

Questions?

Martin Šmarda smarda@avast.com
Pavel Šrámek sramek@avast.com