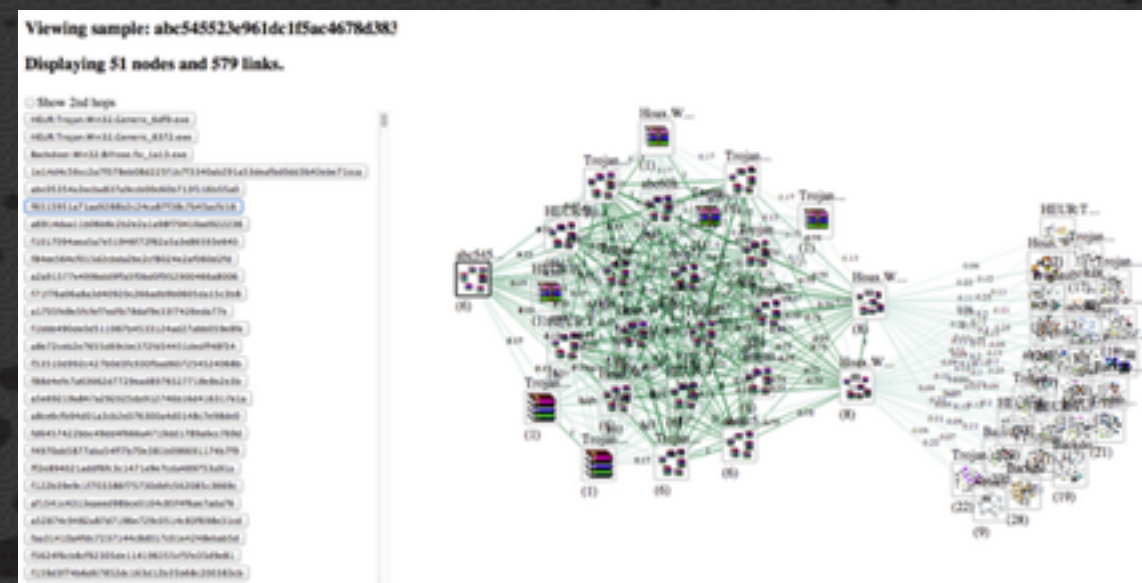


Graphic Content Ahead: Towards Automated Scalable Analysis of Graphical Images Embedded In Malware

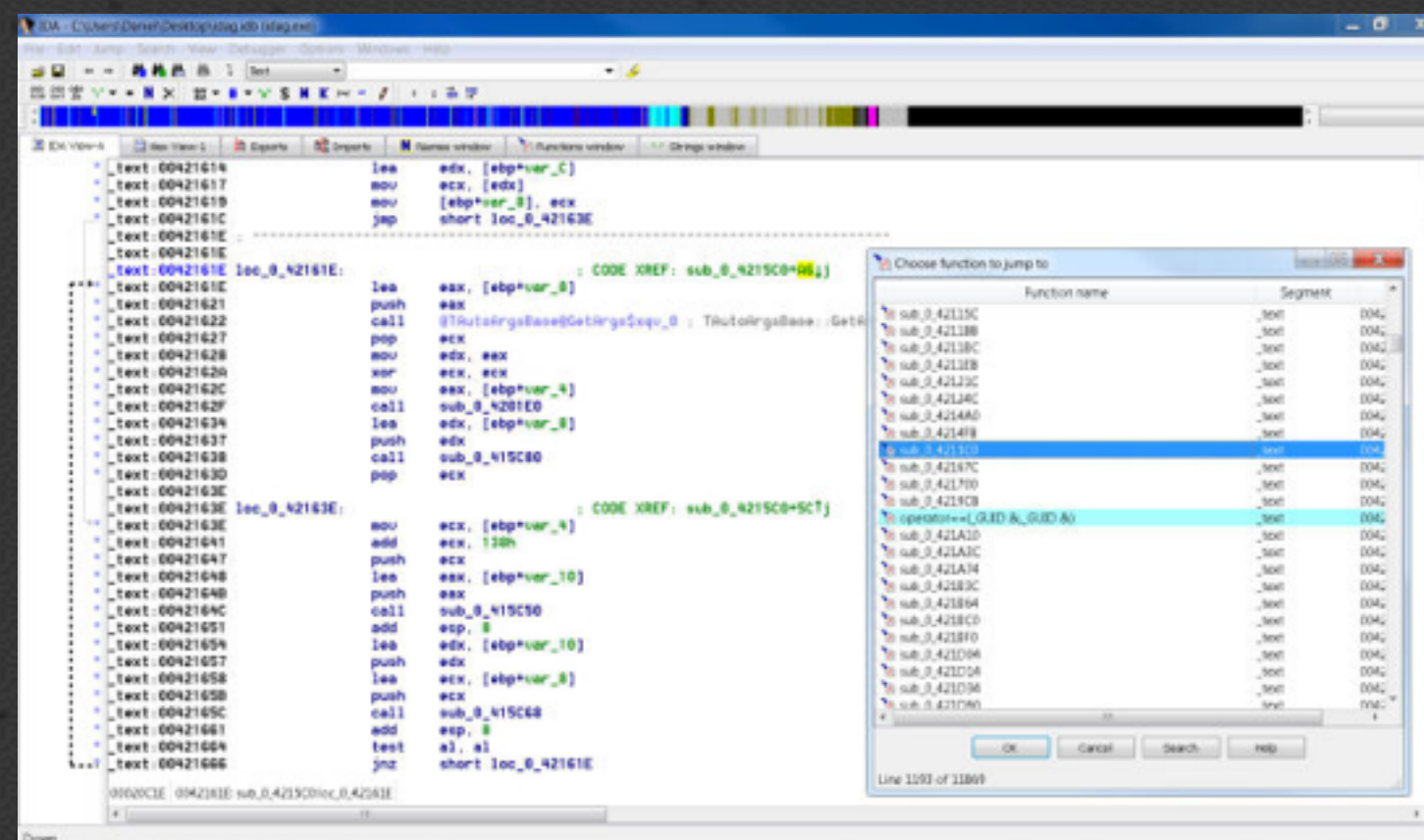


Alex Long, Joshua Saxe, Robert Gove
Invincea Labs

1. The Status Quo of Malware Analysis
2. Hard Problems The Industry is Dealing With
3. Our Approach
4. Two Research Experiments
 1. Detecting and Visualizing Image-Sharing Relationships (Live Demo)
 2. Automatically Classifying Images by Their Semantics



Malware analysis treats malware as just a set of instructions



Analysis typically consists of analyzing the disassembled code and/or observing the malware's runtime behavior



Malware could be packed or use VM detection tactics

Manual analysis of each sample is intractable given huge numbers of polymorphic variants



Malware is not just code,
it's also

- natural language
- documents
- audio
- video
- images



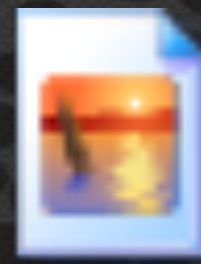
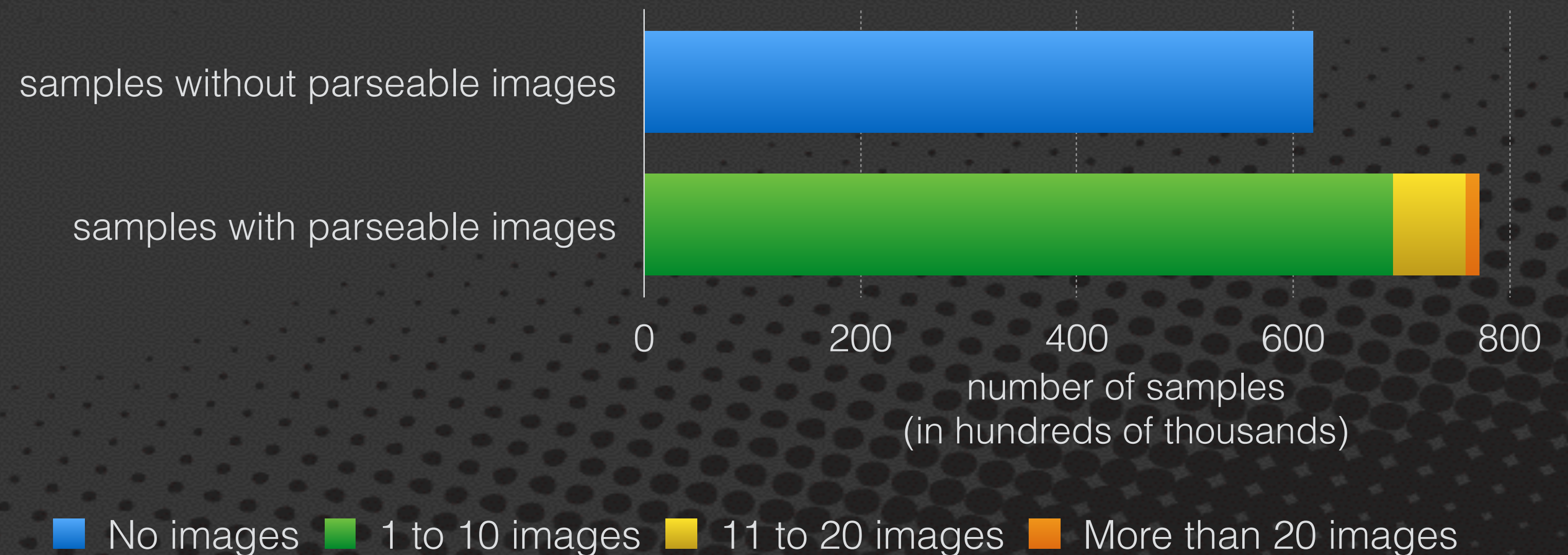
```
text:00401010      push    ebp
text:00401011      mov     ebp, esp
text:00401013      sub     esp, 40h
text:00401016      push    ebx
text:00401017      push    esi
text:00401018      push    edi
text:00401019      lea     edi, [ebp+var
text:0040101C      mov     ecx, 10h
text:00401021      mov     eax, 0CCCCCCC
text:00401026      rep stosd
text:00401028      push    offset ??_C@
text:0040102D      call    printf
text:00401032      add     esp, 4
text:0040103C      mov     eax, [ebp+var
```

“CLICK HERE FOR
FREE PICS”



- **Problem:** Graphical assets are an untapped resource in the malware analysis space; image analysis done manually.

Of a collection of 2 million malware samples provided to us by DARPA, over half had at least one image embedded.



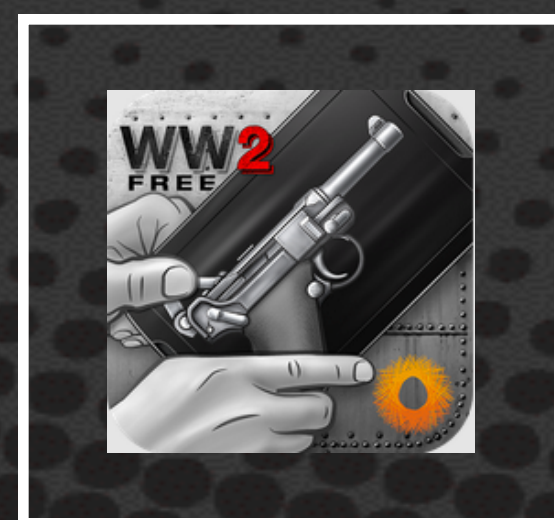
How Image Analysis is Useful

A packed Trojan still needs an attractive icon to lure a user into executing it

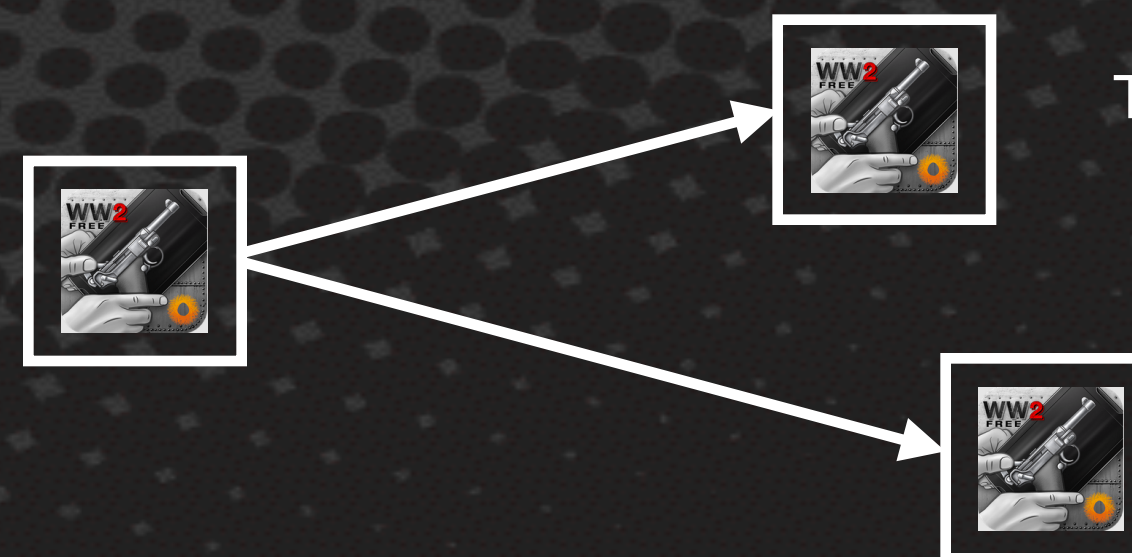


Images can hint at the ways in which attackers are tricking the user and the purpose of a binary artifact.

By exploring the malware's "social network" through shared rare images, you can learn about an otherwise hard-to-reverse sample.



Game-related



Trojan.Win32.VBKrypt...

Trojan.Win32.Swizzor...

Survey of the threat landscape

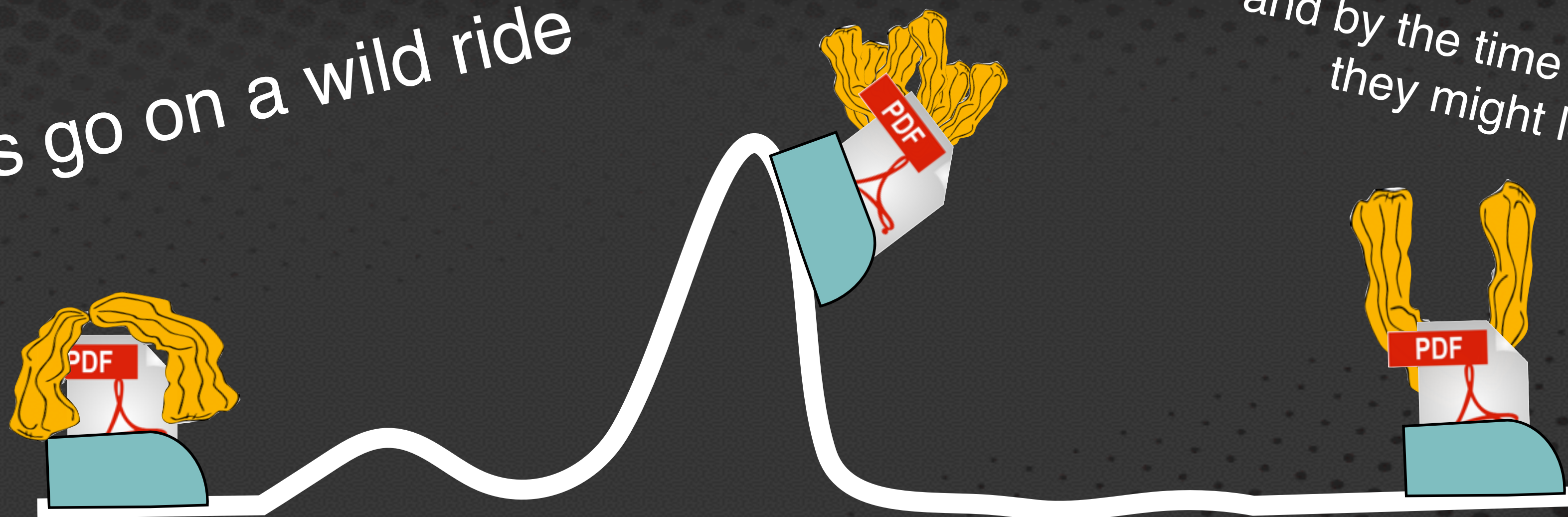
We're seeing an up-tick in malware masquerading as PDFs, let's alert our employees to be on the lookout

Quick analysis of new samples

Our system has found a previously analyzed sample that shares an image with this email attachment

Why Not Just Compare Hashes?

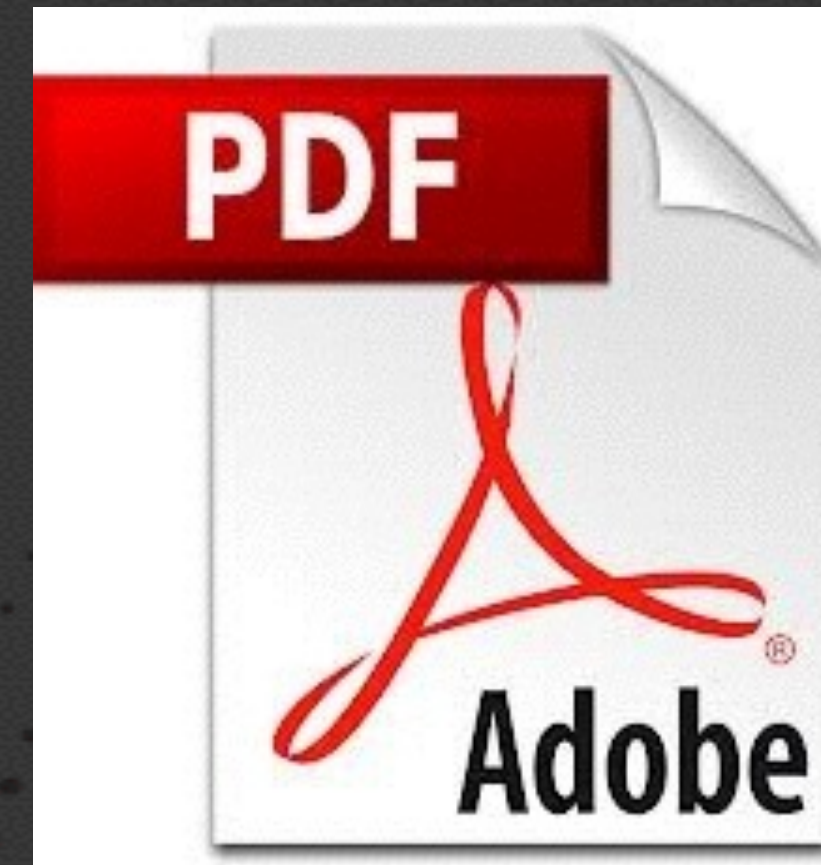
Images go on a wild ride



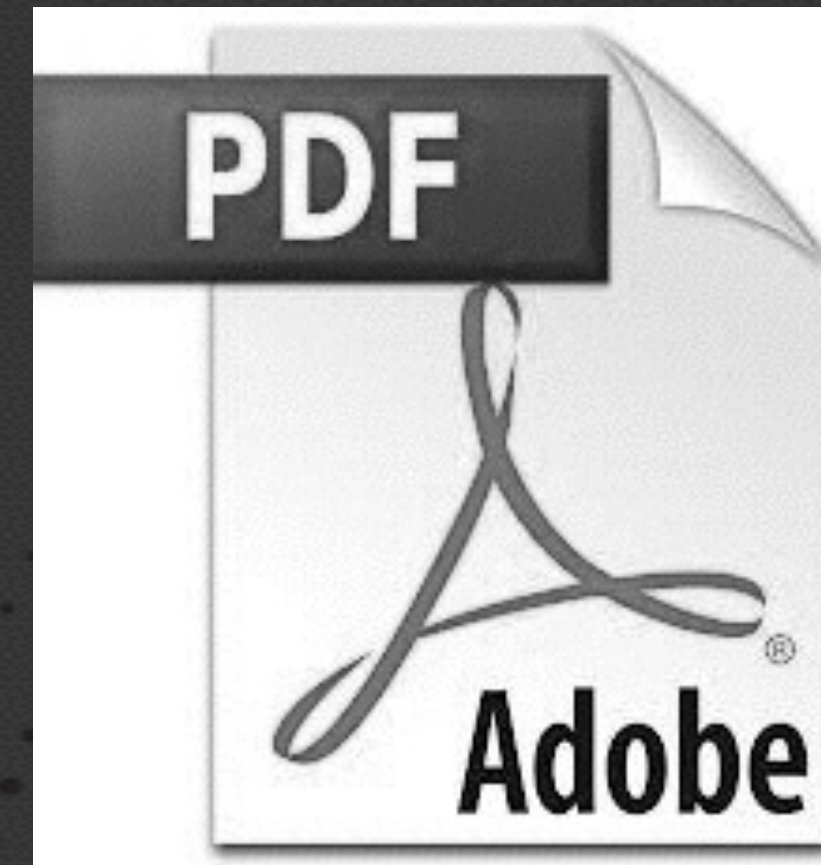
and by the time they end up in malware,
they might look a little different

Hash comparison will fail if the image was
compressed
copied and pasted
modified deliberately

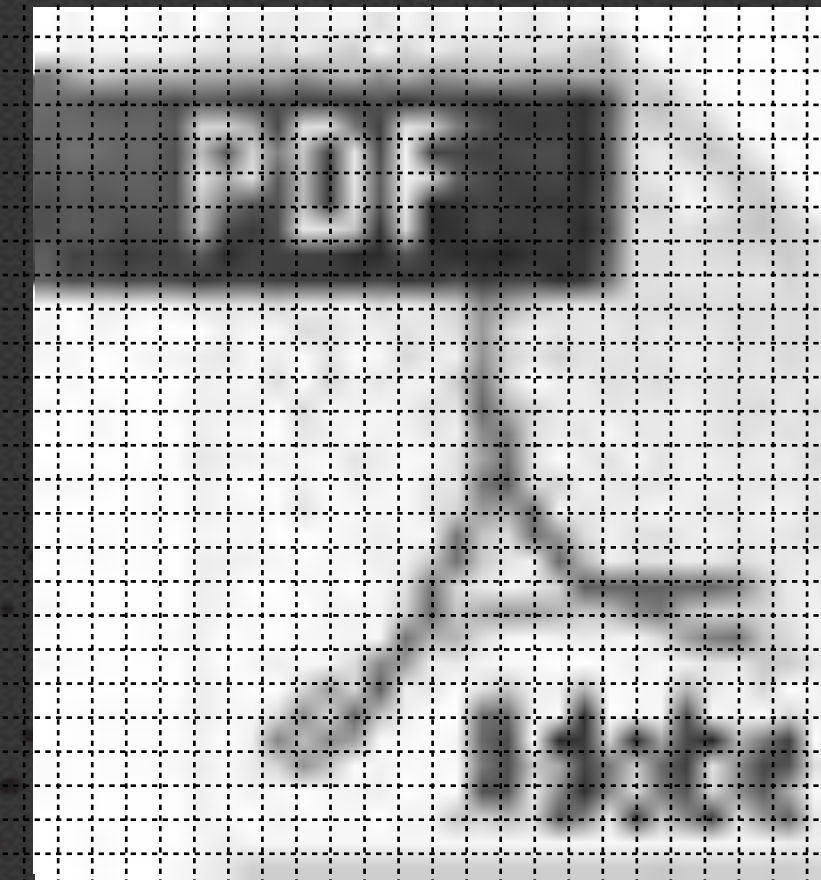
1. Take an image



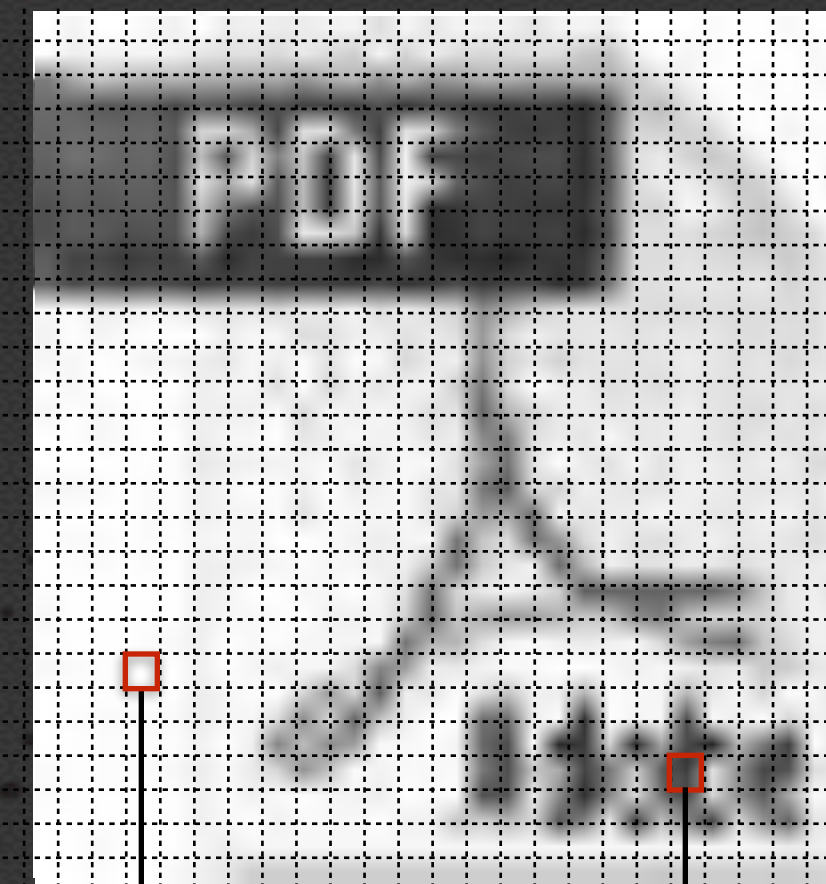
1. Take an image
2. Reduce to grayscale



1. Take an image
2. Reduce to grayscale
3. Stretch/shrink to 32x32

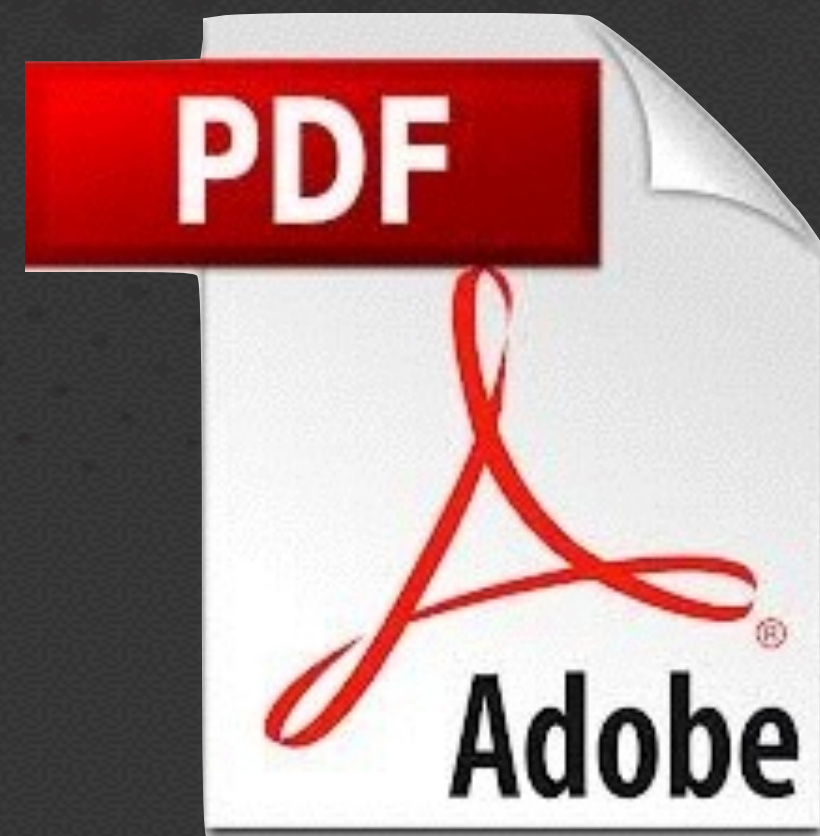


1. Take an image
2. Reduce to grayscale
3. Stretch/shrink to 32x32
4. Convert to high contrast
 - a. Get average value of pixels
 - b. For each pixel,
 - if above average, set to 255
 - if below average, set to 0



1. Take an image
2. Reduce to grayscale
3. Stretch/shrink to 32x32
4. Convert to binary vector
 - a. Get average value of pixels
 - b. For each pixel,
 - if above average, set to 255
 - if below average, set to 0





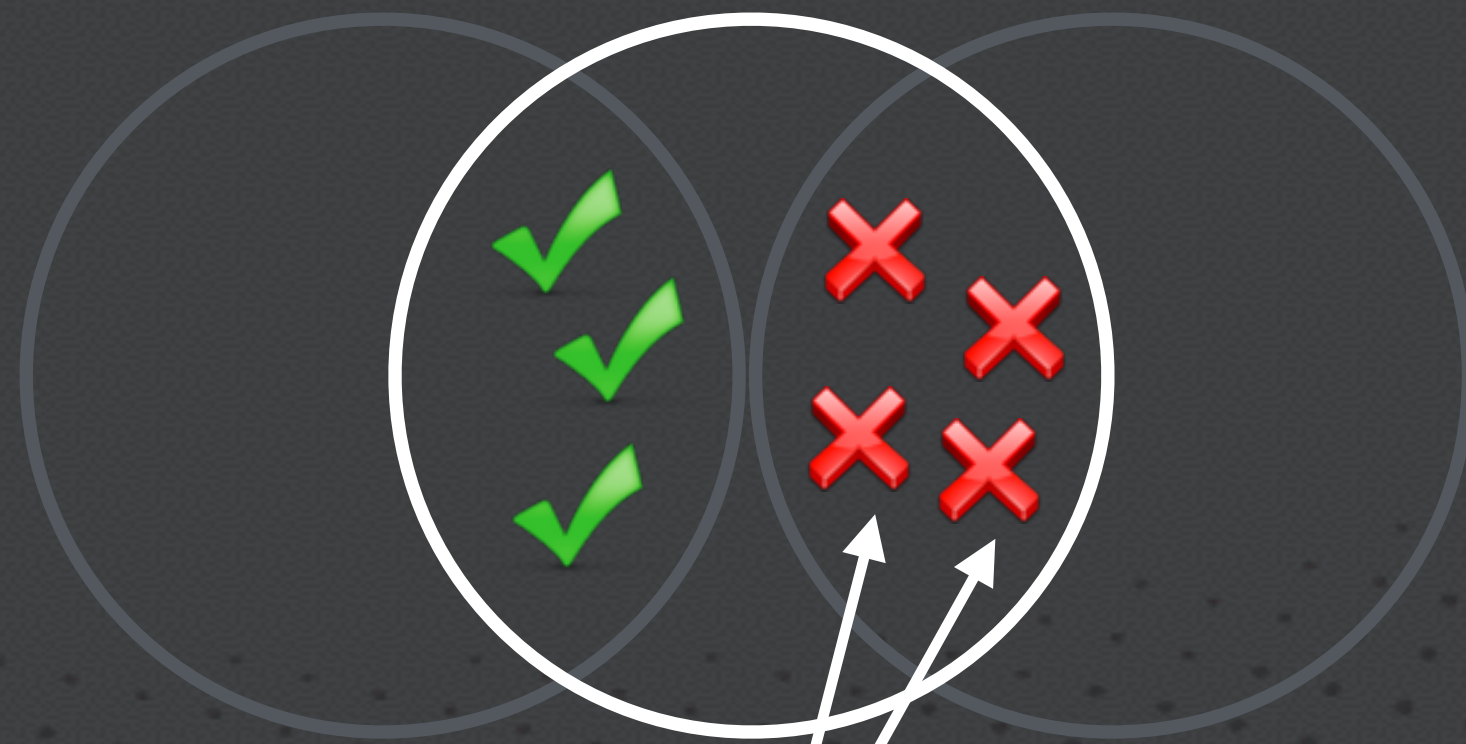
Precision:

What percent of the pairs matched by the system actually had similar images?

Precision

Image Matcher Results

Matching Pairs



Non-Matching Pairs

False positives

Our system says a pair of images are similar that actually are different

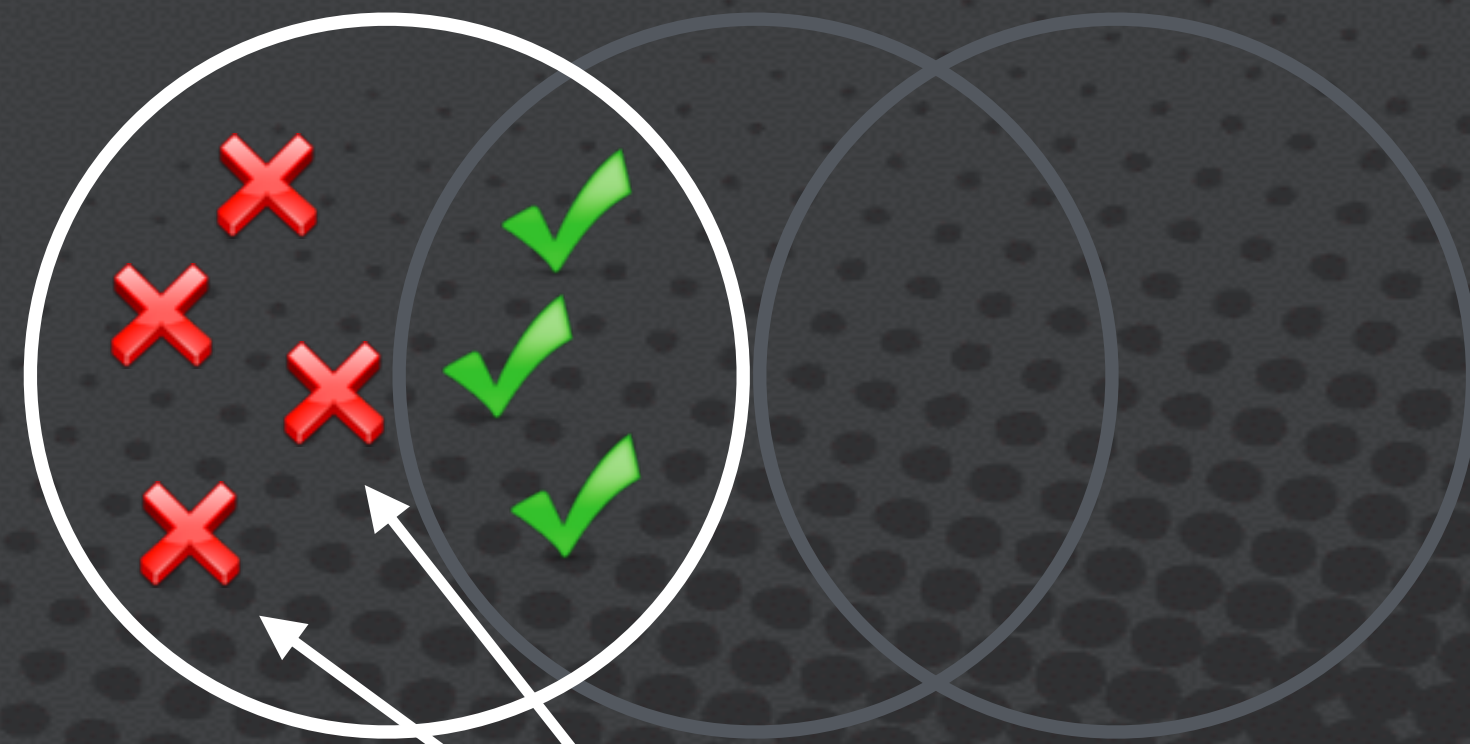
Recall:

What percent of all matching pairs did the system find?

Recall

Image Matcher Results

Matching Pairs

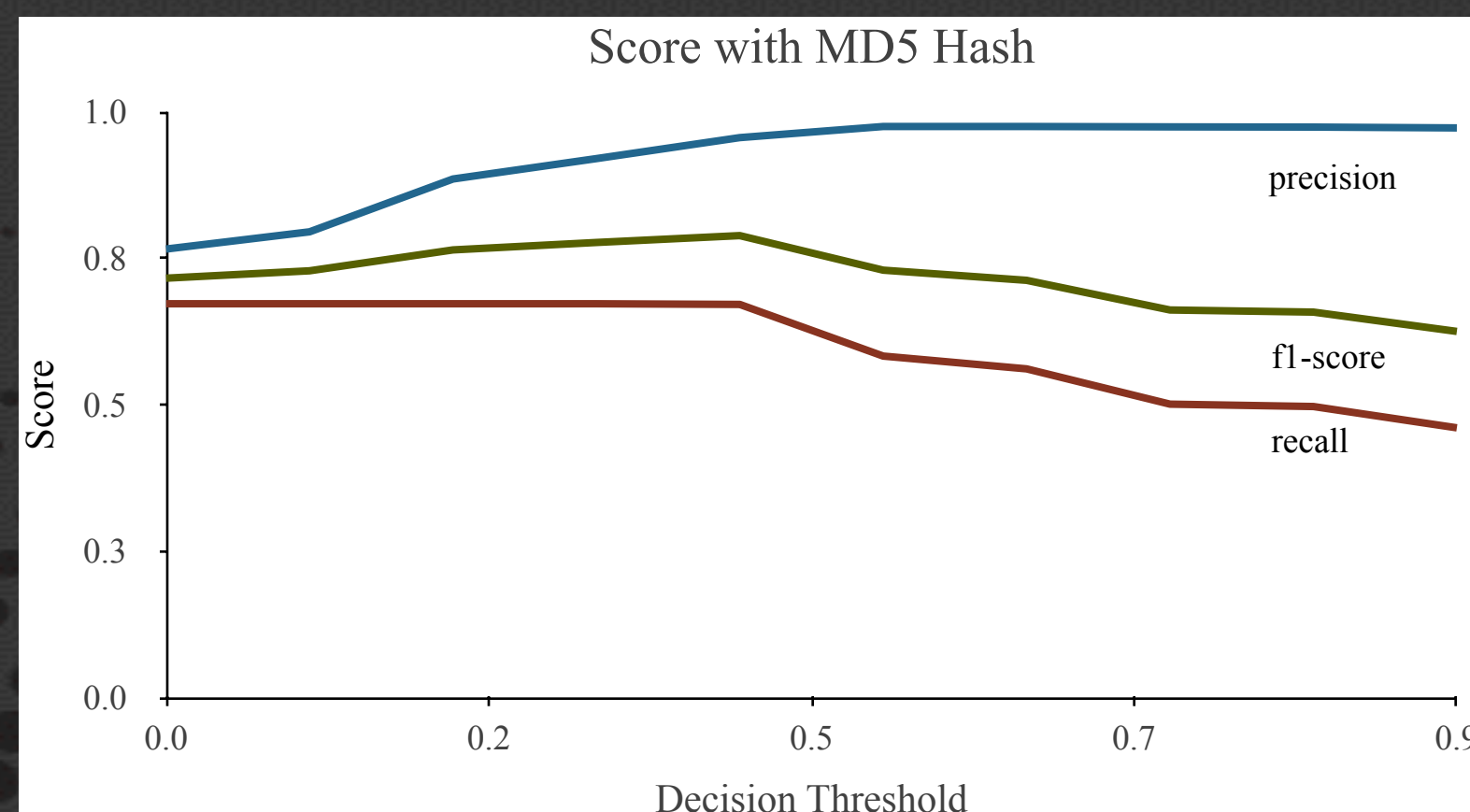
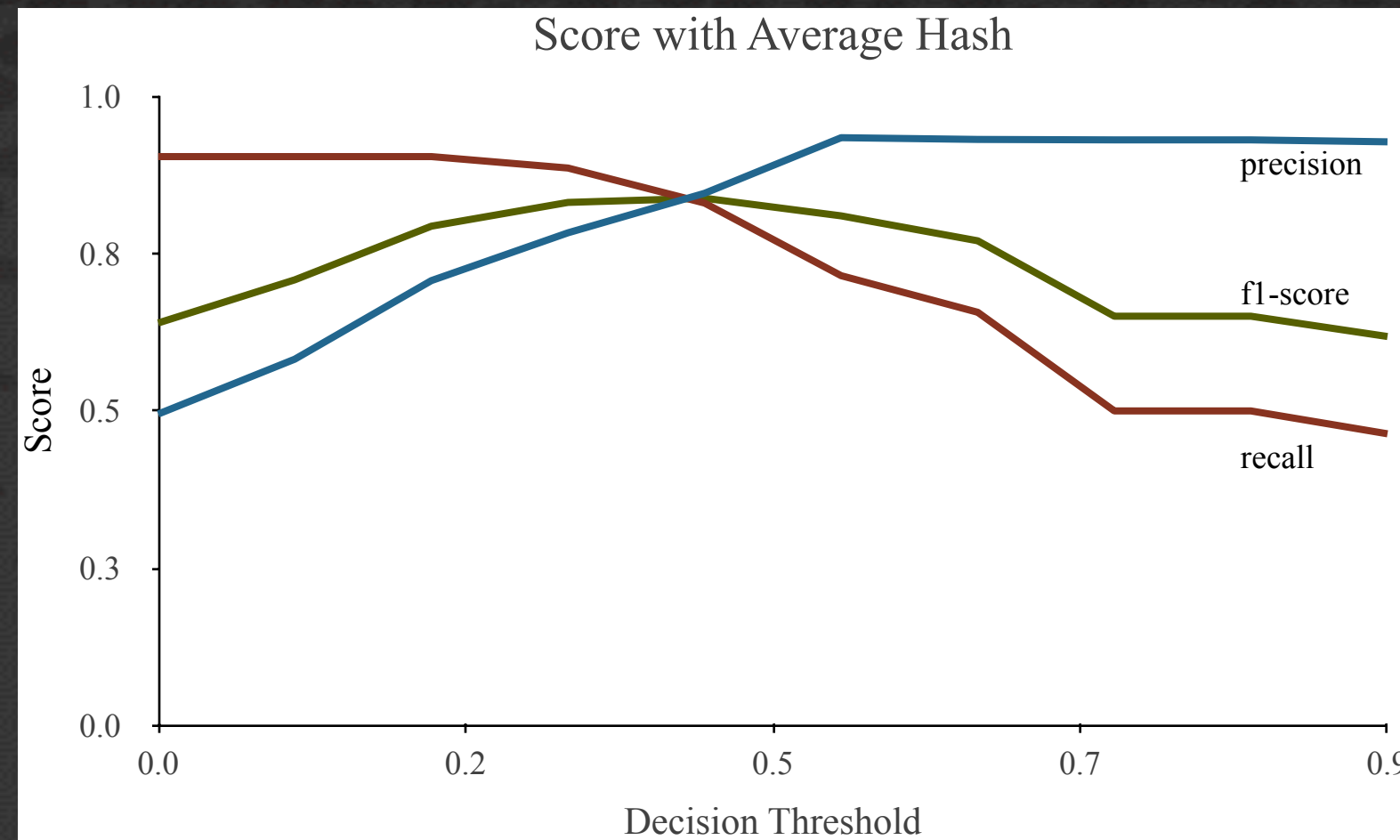


Non-Matching Pairs

False negatives

Our system fails to say a pair of images are similar when they actually are

Average Hash vs MD5



- 200 hand clustered samples
- MD5 wins out in precision
- Average hash wins in recall
- Humans can easily detect and ignore images that don't match (false positives)
- False negatives invisible to user and lost forever
- We'd prefer to have more false positives (lower precision) in order to have fewer false negatives (higher recall)

Comparing Sets of Images

=



?



Comparing Sets of Images

Sample A



=

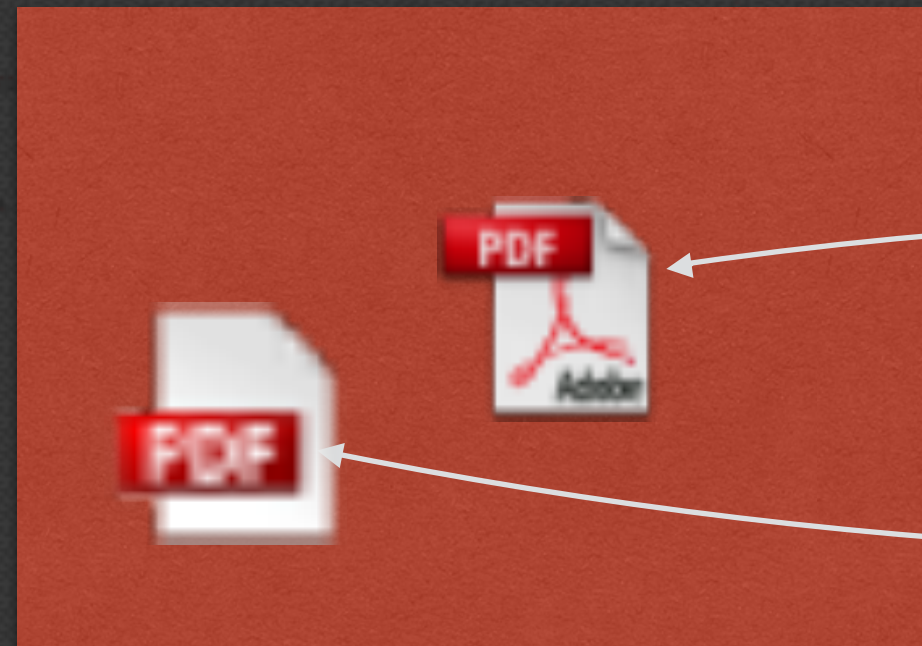
Sample B



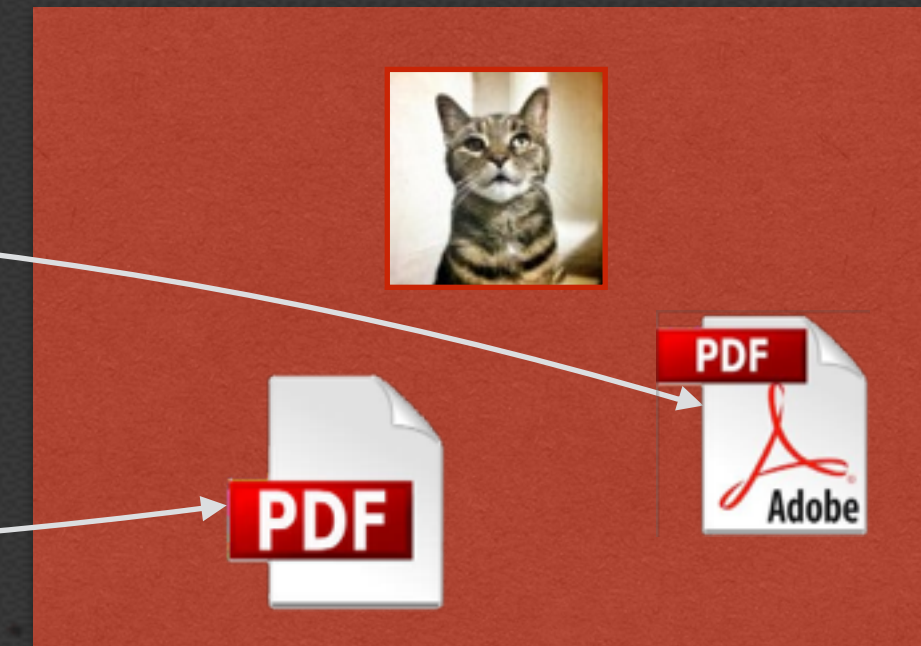
?

Comparing Sets of Images

Sample A



Sample B



Number of matching pairs (2)

$$\frac{\text{Number of matching pairs (2)}}{\text{Number of possible matching pairs (3)}} = 0.66 \text{ similarity}$$

Number of possible matching pairs (3)



Live Demo



Browser



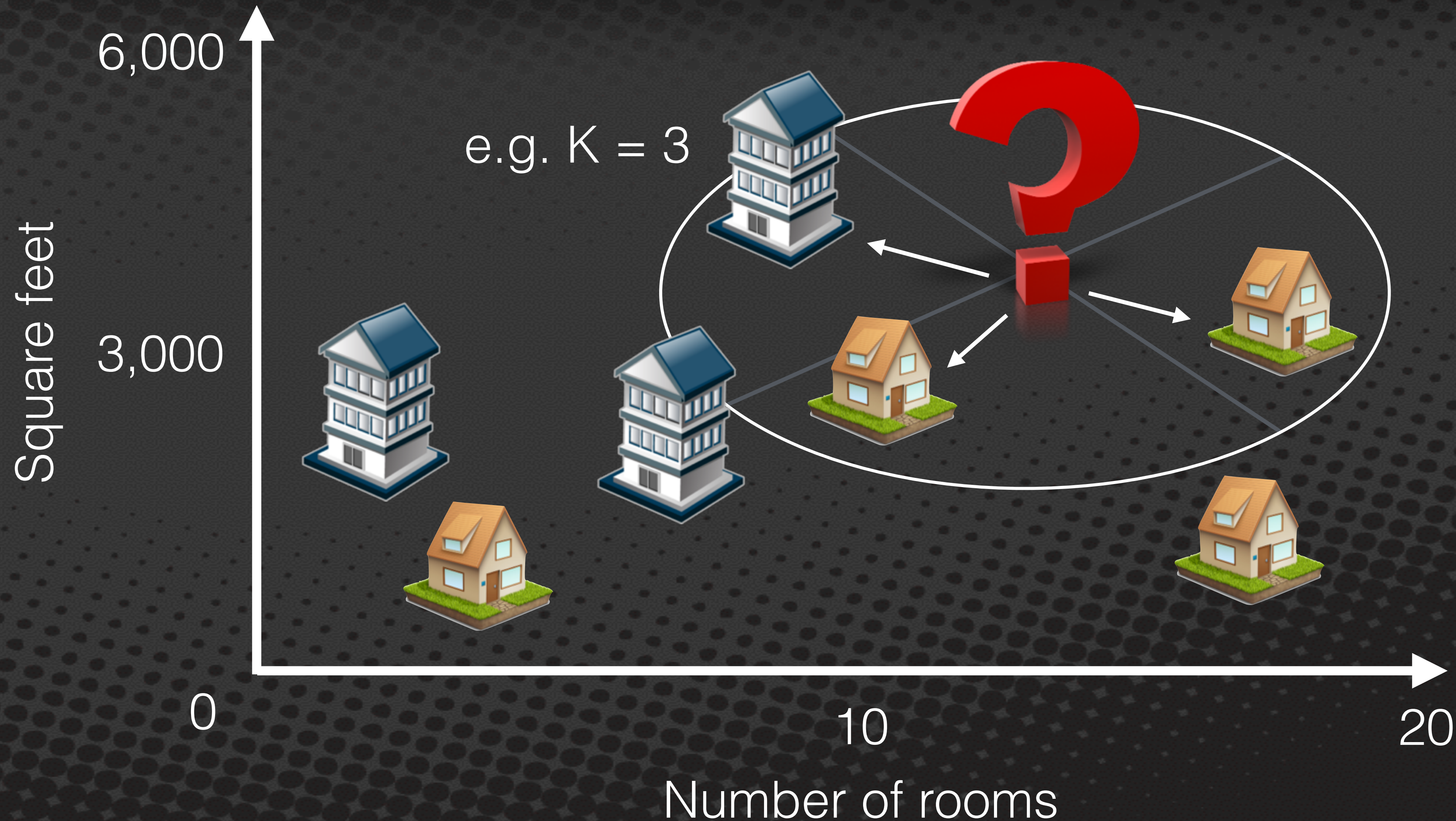
Anti-virus



Document

- Reveal purpose of malware
- Survey threat landscape
- Assign risk factor

Machine Learning 101: K Nearest Neighbors



K Nearest Neighbors

1. Plot points on graph
2. Assign classes to points (e.g. house or apartment)
3. Plot point on graph with unknown class
4. Pick a number K as appropriate for your data
5. Get the top- K nearest neighbors (AKA closest points) to the point with the unknown class
6. Classify by majority vote (e.g. if the 3 nearest points are 2 houses and 1 apartment, unknown class is house)

Classification by Color Histogram

Image with
unknown class

Predicted class

K Nearest Neighbors

	malware_icons/adobe flash.png	adobe flash	
	malware_icons/antivirus.png	windows media player	
	malware_icons/antivirus2.png	windows media player	
	malware_icons/jpg file icon.png	jpg file icon	
	malware_icons/real player icon1.png	windows folder	
	malware_icons/unknown.png	vmware	
	malware_icons/windows installer.png	windows drive icon	
	malware_icons/windows installer1.png	windows installer	
	malware_icons/windows installer2.png	windows media player	
	malware_icons/windows installer3.png	windows internet icon	
	malware_icons/windows network icon.png	antivirus	

Graphical content of malware is a significantly under-utilized signal in malware analysis

Automation and visualization make this “human” signal accessible at a large scale