Machine Learning in Cybersecurity
- Midterm Presentation -

Team :flushed:

Analyzing Apple's NeuralHash.

Q. Zheng, M. Löffler, T. Schneider

1 Refinded final idea

Some claims and ideas from our project proposal have to be corrected for them to be realistic and achievable.

We have claimed that there was no scientific papers similar to our project idea in our initial proposal. Soon this was corrected after we have been pointed to a paper which analyzes collision-resistance of NeuralHash in a very similar fashion as to what we had planned.

Under this information we want to refocus our project goal a little from producing hash-collisions, as this has clearely been done before, towards hardness of this collision finding under different inputs. As already discussed in our initial proposal we want to see if we can notice a difference in the model between pictures of humans and arbitrary ones as its supposed purpose is distinguishing CSAM material.
Can training on specific inputs make the model harder to fool in that regard? Can we enhance the resilience of the model further?

2 Progess

2.1 Model extraction

In a first step we extracted the model together with weight data from a OSX system and converted it to `.onnx` format. This model is ready to be loaded into our jupyter notebook.

2.2 Running a POC attack

We were able to reproduce a proof-of-concept attack obtaining a hash-collision for two seemingly different images by applying rather obtrusive pertubations. The reason for the high pertubation value is that the attack starts reducing pertubation once it could find a true hash collision. On our inputs however this happens rather late, this is further backed by the fact that we reduced the iteration amount by a factor of ten for now. Nonetheless we managed to produce a collision.

4d3032644e122d8c7326cfc9

1ec173f89d10be5300ac0216

1ec173f89d10be5300ac0216

2.3 Challenges

Obtaining the model and converting it to `.onnx` was straightforward, however importing it in the notebook was causing cryptic problems for some of us. Producing neural hashs worked out-of-the-box while running the attack required some tweaking. Still it is not running perfectly. First steps would be to swap out distance and loss measures to produce results under smaller pertubation.

3 Next Steps

Once we manage to reliably produce arbitrary hash-collisions with minimal pertubations we want to progress to analyze hardness of the model in regards to certain input material as well as applying common defenses like retraining the model with pertubated inputs to see if we can improve it further.