Sophos-ReversingLabs (SOREL) 20 Million sample malware dataset

ai.sophos.com/2020/12/14/sophos-reversinglabs-sorel-20-million-sample-malware-dataset/

December 14, 2020

The Sophos AI team is excited to announce the release of SOREL-20M (Sophos-ReversingLabs – 20 million) – a production-scale dataset containing metadata, labels, and features for 20 million Windows Portable Executable files, including 10 million disarmed malware samples available for download for the purpose of research on feature extraction to drive industry-wide improvements in security. This dataset is the first production scale malware research dataset available to the general public, with a curated and labeled set of samples and security-relevant metadata, which we anticipate will further accelerate research for malware detection via machine learning. Code and links to the data are available here.

Why are we releasing this data?

Data is the foundation upon which machine learning models are built. Standardized datasets are the way in which new features and models are developed, tested, and compared to each other. The development and ease of access for standardized datasets such as the MNIST

digits dataset, and later, large scale, realistic datasets, such as the ImageNet dataset and the Pascal Visual Object Classification dataset, sparked an explosion in machine learning for image recognition that culminated in the super-human models available today.

Unlike image recognition or natural language processing, the area of security has seen much less activity and a relatively slower rate of improvement. A major reason for this is simply the lack of a standard, large-scale, realistic data set that can be easily obtained and tested by a wide range of users, from independent researchers to academic labs to large corporate groups. Obtaining a large number of curated, labeled samples is both expensive and challenging, and sharing data sets is often difficult due to issues around intellectual property and the risk of providing malicious software to unknown third parties. As a consequence, most published papers on malware detection work on private, internal datasets, with results that cannot be directly compared to each other.

The EMBER dataset (https://github.com/endgameinc/ember; https://arxiv.org/abs/1804.04637) was an initial step to address this problem, however the Ember dataset is relatively small (approximately 1 million files), and contains only a single label per sample (benign/malware), limiting the range of experimentation that can be performed with it. It also includes no raw samples, meaning users must rely on the features pre-extracted by the developers of the dataset. By contrast, our dataset contains complete – albeit disarmed – samples for malware, malware/benignware labels, as well as the number of positive results across ReversingLabs scanners and tag counts derived from detection names. We also provide dumps of metadata extracted via the `pefile` library (https://github.com/erocarrera/pefile) for all samples.

What's in the dataset?

The SoReL-20M dataset, developed in collaboration between Sophos AI and ReversingLabs, is intended to further accelerate research in malware detection via machine learning. SoReL 20M is a production-scale dataset covering 20 million samples, including 10 million disarmed malware samples available for download, as well as extracted features and metadata for an additional 10 million benign samples. In practice, we find that 20 million samples is sufficient to obtain a good rank-ordering of models, allowing us to obtain fair and stable comparisons between models. These samples are divided into training, validation, and testing splits on the basis of first-seen time. For each sample, we provide:

1. Features extracted as per the EMBER 2.0 dataset
2. Labels obtained by aggregating both external and Sophos internal sources into a single, high-quality label
3. Sample-per-sample detection metadata, including total number of positive results on ReversingLabs engines, and tags describing important attributes of the samples obtained as per our paper "Automatic Malware Description via Attribute Tagging and Similarity Embedding" https://arxiv.org/abs/1905.06262

4. Complete dumps of file metadata obtained from the pefile library using the dump_dict() method
5. For malware samples, we provide complete binaries, with the OptionalHeader.Subsystem flag and the FileHeader.Machine header value both set to 0 to prevent accidental execution.

To accompany this data, we are also releasing a set of pre-trained PyTorch (https://pytorch.org/) models and LightGBM (https://github.com/Microsoft/LightGBM) models trained on this data as baselines, as well the scripts used to load and iterate over the data, and load, train, and test the models. The PyTorch-based deep learning models are based on the ALOHA architecture (https://www.usenix.org/system/files/sec19-rudd.pdf); LightGBM parameters were selected using HyperOpt library. Code used to train the models, as well as links to the data, are all available at https://github.com/sophos-ai/SOREL-20M

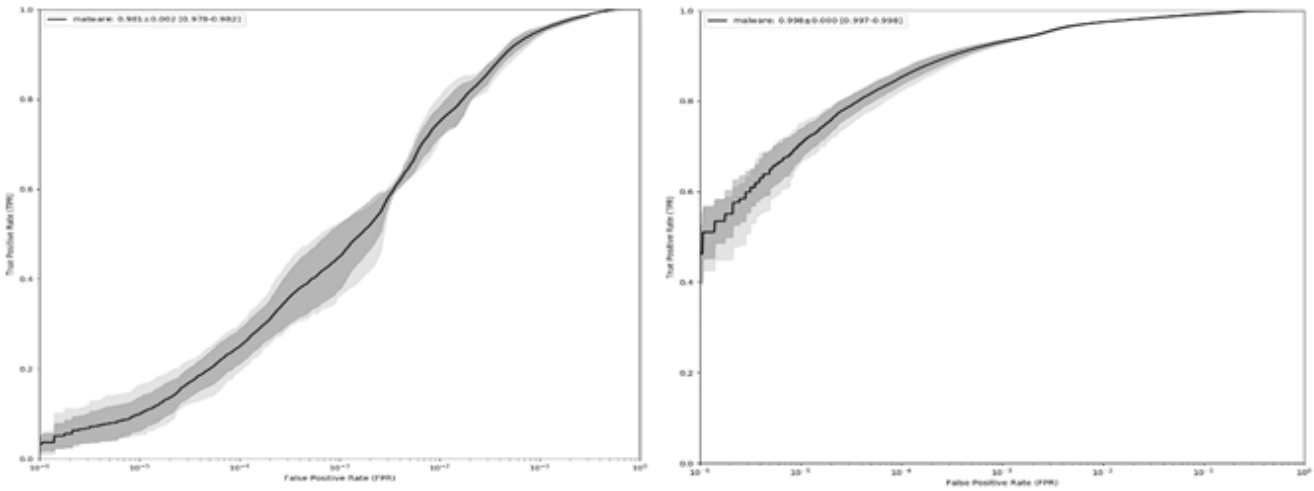

Figure 1 — Left: ROC of LightGBM model; Right: ROC of ALOHA deep learning model

Figure 1 shows the ROC curves for the LightGBM (left) and ALOHA (right) models. We also show the performance of the ALOHA tags in Figure 2. While the performance of the baseline models is quite good, they use a fairly standard set of features: those provided by the EMBER dataset (with minor modifications to make the code python3 compatible). Plots showing the distribution of tags in the malware portion of each dataset, and the co-occurrence relationship between tags, are provided in figures 3, 4, and 5.

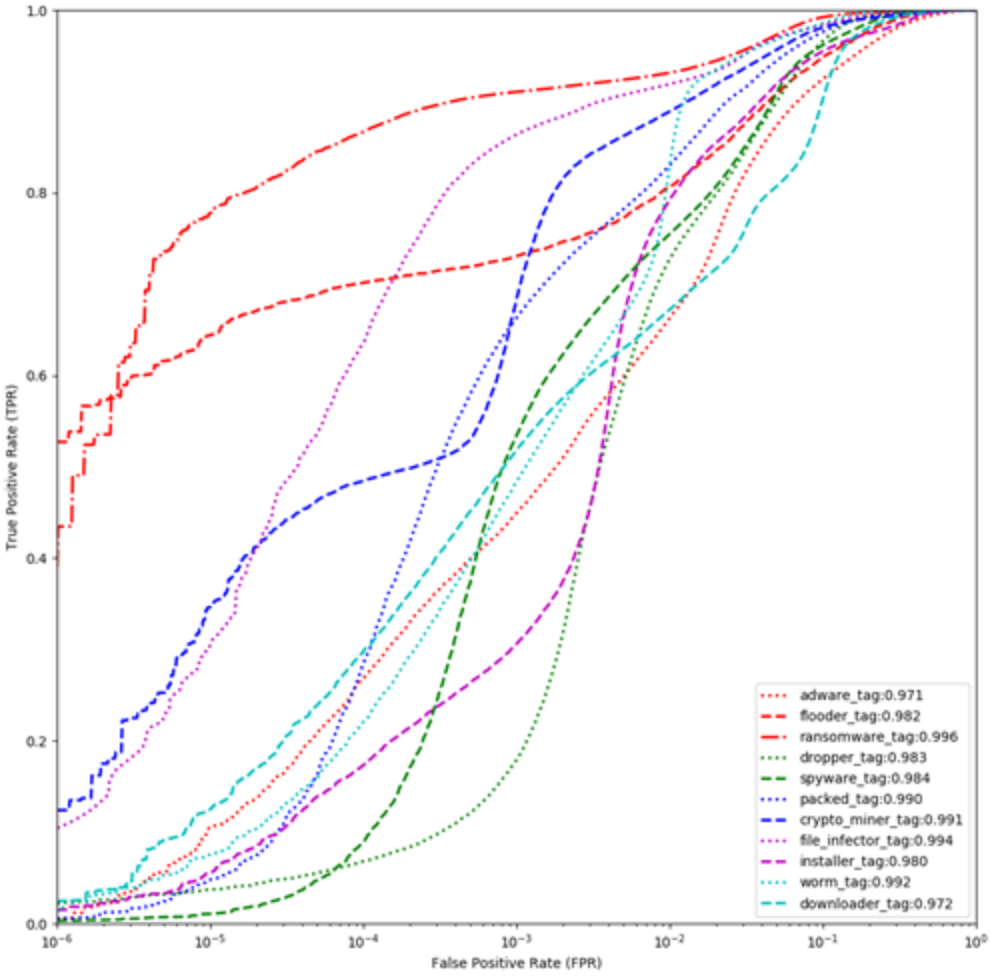

Figure 2 — per-tag ROC for ALOHA model

Are you concerned that attackers could use this dataset to cause harm?

The malware we're releasing is "disarmed" so that it will not execute. This means it would take knowledge, skill, and time to reconstitute the samples and get them to actually run. That said, we recognize that there is at least some possibility that a skilled attacker could learn techniques from these samples or use samples from the dataset to assemble attack tools to use as part of their malicious activities. However, in reality, there are already many other sources attackers could leverage to gain access to malware information and samples that are easier, faster and more cost effective to use. In other words, this disarmed sample set will have much more value to researchers looking to improve and develop their independent defenses than it will have to attackers.

There is a consensus in the cybersecurity industry that responsible offensive engagements make us all stronger, and the public release of attack tools facilitates that. Mature projects like 'afl' support vulnerability identification (https://lcamtuf.coredump.cx/afl/), tools like sqlmap (http://sqlmap.org/) and Metasploit provide adversaries means to gain entry into and privilege escalation within networks (https://www.metasploit.com/download), tools like MimiKatz

support lateral movement https://github.com/gentilkiwi/mimikatz/releases, tools like Powershell Empire support persistence, data exfiltration, and other capabilities (https://github.com/EmpireProject/Empire). Responsible availability of these tools gives defenders more information about what they are defending against, which leads to more effective defenses. Open knowledge and understanding about cyber threats also leads to more predictive cybersecurity. Defenders will be able to anticipate what attackers are doing and be better prepared for their next move.

The malware binaries we're releasing have been in the wild for some time. Even if they were operational, we expect that they would be configured to reach out to command and control infrastructure that has already been dismantled. We also expect that the vast majority of the samples will be readily recognized by anti-virus vendors. We are publishing metadata with the samples, including hash values, that will enable anti-virus vendors to recognize the samples if they don't already. We also expect that these samples will quickly become very well-recognized as researchers have the opportunity to work with them.

Finally, we anticipate that the public benefits of releasing our dataset will include significant improvements in malware recognition and defense. We're offering an opportunity for researchers to make their results commensurable. We've seen what this can do for a field in the way computer vision specifically and deep learning more generally has been revolutionized by the ImageNet benchmark image dataset. And, as with colleagues responsible for releasing related datasets, we're breaking the ice on releasing this kind of data to the community, to promote a culture of benchmark creation that we hope will go beyond what we're doing here.

Detailed corpus statistics (for the technically inclined)

All data is available via AWS S3 at s3://sorel-20m/09-DEC-2020 – baseline pretrained models and results are available in the 'baselines' subdirectory; pre-extracted features and metadata are in the 'processed-data' subdirectory, and the raw (defused) binaries – compressed using the python zlib library – are contained in the 'binaries' sudirectory.

The SQLite database schema for the `meta.db` file within the 'processed-data' subdirectory is as follows:

```
CREATE TABLE meta (sha256 text primary key,
  is_malware SMALLINT,
  rl_fs_t DOUBLE,
  rl_ls_const_positives INTEGER,
  adware INTEGER,
  flooder INTEGER,
  ransomware INTEGER,
  dropper INTEGER,
  spyware INTEGER,
  packed INTEGER,
  crypto_miner INTEGER,
  file_infector INTEGER,
  installer INTEGER,
  worm INTEGER,
  downloader INTEGER );
```

- sha256 – the sha256 of the unmodified file (note that all provided files are “disarmed”)
- is_malware – a value of 0 indicates benignware, 1 indicates malware
- rl_fs_t – the first time (in Unix epoch time) a given sample (unique per sha256) was seen in the ReversingLabs feed
- rl_ls_const_positives – the total number of ‘positive’ (i.e. malware) results from all detectors at the most recent time that the samples was seen (assuming that more recent scans will be higher quality due to signature updated etc
- adware, flooder, ransomware, dropper, spyware, packed, crypto_miner, file_infector, installer, worm, downloader – the number of tokens appearing in detection names that related to the specified tag; a value >0 indicates a positive result, larger values may indicate higher certainty in the tag

Frequency of malware and tags are given below; note that the dataset is approximately balanced with respect to malware, but otherwise rather imbalanced. There are 19724997 samples in the data, broken down as follows:

	Yes	No
Is_malware	9762177	9962820
Adware	2411262	17313735
Flooder	101595	19623402
Ransomware	1152354	18572643
Dropper	3577111	16147886
Spyware	4550007	15174990
Packed	3726059	15998938
Crypto_miner	339565	19385432

File_infector	3317569	16407428
Installer	1018880	18706117
Worm	3414132	16310865
Downloader	2565838	17159159

For per-split tag distributions, see the plots at the end of this post.

We provide metadata in LMDB databases (key-value stores) indexed by sample sha256 and containing compressed json files. Each malware sample – prior to modification – was loaded via the pefile library and the `dump_dict()` method called on the result. When the pefile module failed to parse the sample, no value was entered into the LMDB database. We also provide EMBER (v2) features, stored as zlib-compressed and msgpack-serialized numpy arrays. As with the metadata, when EMBER failed to parse a PE file, there is no associated sha256 key or value in the LMDB database.

Conclusion

SoReL-20M is the first production scale, curated, labeled dataset for use in machine learning research for malware detection, released by the Sophos AI group. We hope that it will further accelerate research on models and features for malware detection, enable fair comparisons of approaches across labs and researchers, and ultimately improve the ability and agility of defenders to offer protections against attackers.

Appendix: Additional corpus statistics

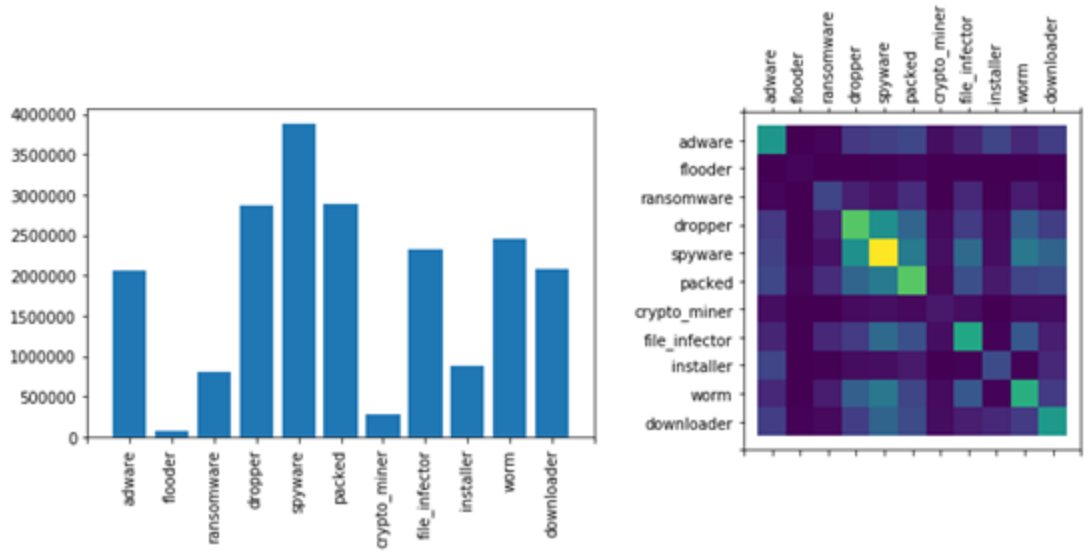

Figure 3 — Training data tag distribution and co-occurrence matrix

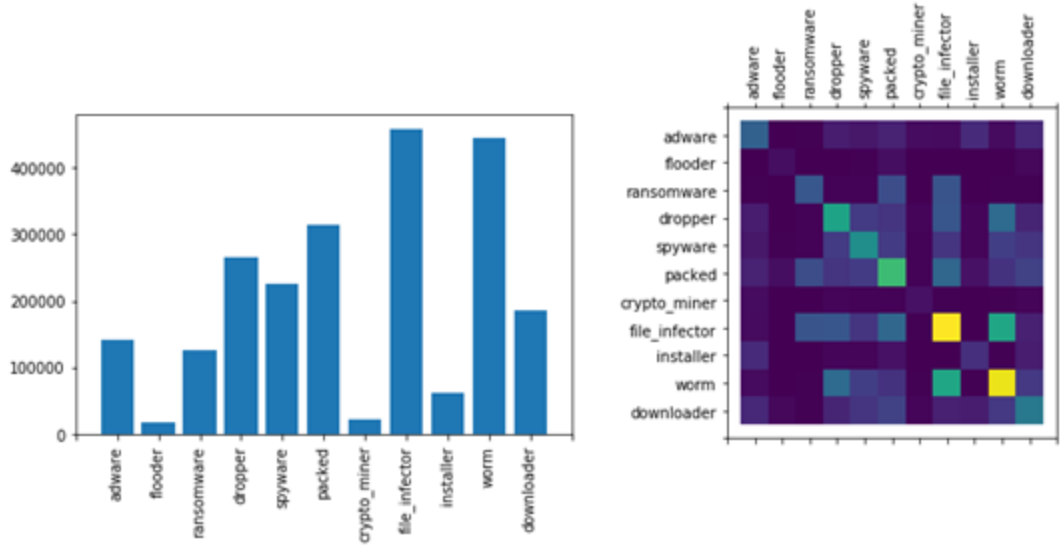

Figure 4 — Validation data tag distribution and co-occurrence

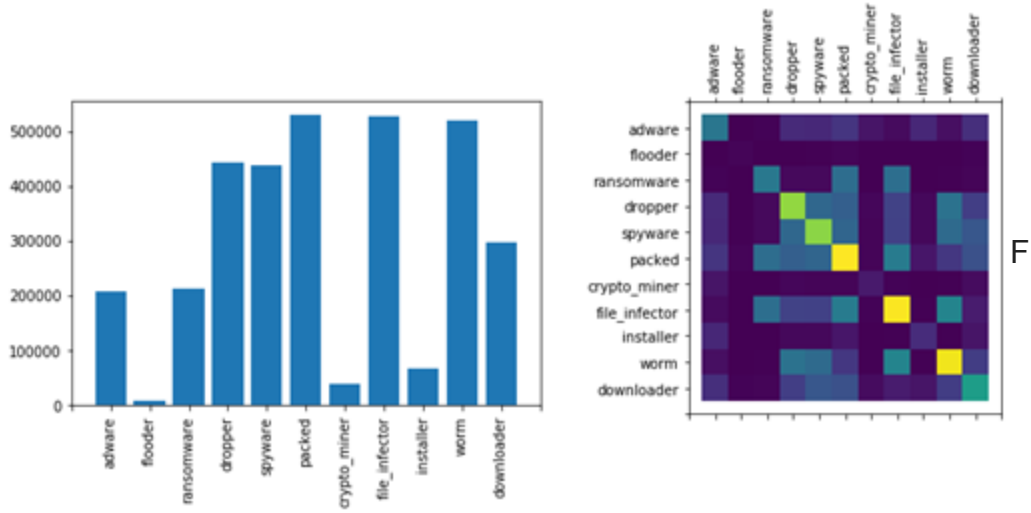

Figure 5 — Test data tag distribution and co-occurrence