Paper #402 Summary of Changes

We thank all the reviewers for the constructive comments and suggestions. Below, we summarize the key changes that we made to address the reviewers' comments. In this document, we only list major changes to the paper. For other comments (such as typos), we directly addressed them in the paper and mark them out using a diff document.

Major Comments:

Item	Reviewer Comments	Modification
Clarification	RB: Clearly define the term "label"	We added a clarification in the paragraph with the caption "**VirusTotal Scanning APIs.**" in **Section 2**: "indicated by the 'detect' field in VirusTotal responses".
	RB: justification for label correlations between engines	We added one sentence to justify it in the second paragraph of **Section 5**: "If we could observe correlations or causalities between certain engines, then independence assumption would be questionable".
Discussion	RD: VirusTotal only including a small fraction of malicious samples: what could researchers do about it?	This is a great question. We added one paragraph with the caption "**Malware Coverage Issues.** " in **Section 7** to discuss this question. Basically, as pointed out by the two previous works [38, 48], VirusTotal's malware coverage has limitations. Right now, VirusTotal is already trying to improve its coverage by providing free malware services to the public to gather new malware samples from different sources. In the future, VirusTotal can introduce new incentive mechanisms to encourage broader sharing of malware intelligence. [38] Waves of malice: A longitudinal measurement of the malicious file delivery ecosystem on the web. AsiaCCS'19. [48] The dropper effect: Insights into malware distribution with downloader graph analytics. CCS'15.
	RD: tradeoffs of setting different	This comment covers both stabilization and accuracy.

	thresholds	For stabilization, we improved **Section 4.5** from the following aspects. First, we changed **Figure 5** from a CDF to a bar chart to better illustrate how the percentage of influenced files (i.e., files with both malicious and benign labels) changes with the threshold (t). Second, we discussed two more t values (t=39 and t=40) in two paragraphs under the captions "**Setting 2<=t<=39.**" and "**Setting t>=40.**" on Page 8 respectively. Third, we emphasized that "even though the threshold-method helps stabilize the aggregated labels, it does not necessarily mean the aggregated labels are correct" in the third paragraph on **Page 9**. Figures 14 and 15 show the trade-off between precision and recall given different thresholds. The general observation we have is "a small threshold value can balance the precision and the recall as long as the benign files are not obfuscated". We emphasized this in **Observation 10**.
	RE: differentiating their paper from [40]	We have updated the discussion in **Section 2,** under the caption "**Closely Related Works**". We emphasize that "the data collection method of [40] is different (e.g., file re-scanning frequency is not controlled), which lost the opportunity to observe fine-grained label dynamics" and "[40] did not have real ground-truth for the malware dataset, and assumed VirusTotal labels become stable after a file is submitted to VirusTotal for four weeks (for which we have different observations)". [40] Better malware ground truth: Techniques for weighting anti-virus vendor labels. AISec'15.
APK files	RA, RB, RC: Limitations of PE files? What if you test APK files?	Thank you for the suggestion. Indeed, focusing on PE files has its limitations. We discussed the limitations in the paragraph with caption "**Limitations & APK Experiments.**" in Section 7. We also added the results of a quick measurement on Android APK files in the following two paragraphs. Basically, we created a main APK dataset with fresh APK samples firstly submitted to VirusTotal on December 26, 2019. We collected their daily labels for 46 days. We have the following

		observations. First, there are both hazard flips and non-hazard flips on APK samples. Second, the top three engines with the most flips are Microsoft, Fortinet, and Tencent. The engine ranking is different from that of PE files. Third, there are still engines with label correlations on APK samples. Four, some engines that are highly-influenced under PE files become “influencers” under APK files. Overall, we think “engines face common problems such as label instability, and they have their own specialities for different malware types”.
Ground-truth Experiment	RE: The ground truth experiment: need representative and non-biased dataset.	This is indeed intriguing to have unbiased ground-truth. But it is rather challenging to construct such ground-truth datasets and re-do the experiment in the given time frame. As we mentioned in the paper, the ground-truth needs to be fresh-files (to trigger scanning rather than database lookup). Also, we need to manually inspect the ground-truth, which limits the scale of the ground-truth. We made some clarifications on such challenges and acknowledge the limitations in the last paragraph paragraph (under the caption “**Limitations.**”) of **Section 3.2** and the first paragraph under the caption “**Limitations & APK Experiments.**” in **Section 7**.
	RE: desktop experiment (Section 6.3) should be configured correctly	For the desktop experimen, we blocked the network of desktop engines in order to isolate the engines on the desktop from the engines on VirusTotal to compare them fairly. We added this clarification in the paragraph under the caption “**Experiment Setup.**” in **Section 6.3**. We also added a sanity check (the paragraph under the caption “**Sanity Check (w/ Internet)**” in **Section 6.3**). We confirmed that most (23 out of 36) desktop engines report the same results after connecting to the Internet. Our observation that “desktop engines are more conservative with a higher precision and lower recall” is still held after connecting desktop engines with the Internet.
Missing information	RE: more details about missing information from the	The missing information means “the data fields for update date or version information are ‘null’” in the VirusTotal

	data collection process	responses (that are successful), so that we cannot categorize the root causes for the corresponding flips. Since “the ‘detected’ values (i.e., label information) are still available in these responses”, “the missing information does not impact our analysis in other sections”. We added this clarification in the last paragraph but one in **Section 4.3**.
Related works	RD: Discuss the related work about automatically determining the family of a malware RE: you missed two related works	We added the discussion for the related papers [37, 55, 63] on malware family attribution in the last paragraph (“**Other Related Works.**”) of **Section 2**. We added three new papers [37, 48, 63] in our paper survey (**Section 2**). Thus, we changed **Table 1** and corresponding texts. We did not add [55], because it is to evaluate VirusTotal engines, not using “VirusTotal to label their datasets”, or leveraging “the querying/scanning API of VirusTotal as a building block of their proposed systems” (the second paragraph of **Section 2**). [37] Euphony: Harmonious unification of cacophonous anti-virus vendor labels for android malware. MSR’17 [48] The dropper effect: Insights into malware distribution with downloader graph analytics. CCS’15 [55] Av-meter: An evaluation of antivirus scans and labels. DIMVA’14 [63] Avclass: A tool for massive malware labeling. RAID’16

Minor Comments

Item	Reviewer Comments	Modification
Other Clarification	RC, RD, RE: Why should obfuscation not be treated as malicious?	We use obfuscation to create “fresh” binaries that have never been scanned by VirusTotal before. We think “obfuscation is not necessarily a determining feature of malware”, because “it is also often used by legitimate software to protect their intellectual property (copyright) or protect sensitive functions (e.g., for payments and security) from reverse-engineering [26, 61, 77].” We added this discussion in the paragraph with the caption “**Ground-truth Malware.**” in **Section 3.2**.
	RA: Why using ransomware for ground-truth? How limited is it?	We choose ransomware as the seeds to create more ground-truth malware due to two reasons. “First, it is easy to manually verify the malicious actions (i.e., encrypting files, showing the ransomware notes). Second, we manually confirmed that the majority of engines (57/65) advertise that they are capable of detecting ransomware, and thus ransomware is a relatively fair benchmark to compare different engines.” We added this discussion in the fourth paragraph of **Section 3.2**. Extending the experiments to other malware can help but it requires non-trivial manual efforts to verify every obfuscated instance’s behavior. We discussed this reason and admitted the limitation in the last paragraph (“**Limitations.**”) of **Section 3.2** and the paragraph under the caption “**Limitations & APK Experiments.**” of **Section 7**.
	RC: Why not use sliding bin	In the second paragraph of **Section 5.1.1**, we explained the reason: “We do not compute feature vectors using sliding bin to avoid counting the same flip multiple times. ”
	RC: Whether the presence of other parties can influence engines’ label	In the paragraph with the caption “**Engine Independence.**” of **Section 7**, we discussed how to interpret the identified causal relationships: “We also identified several engines whose labeling exhibits causal relationships (Section 5.2). This does not necessarily mean one engine directly copies results from

		other engines – it is also possible these engines change labels due to the impact of third parties (blacklists), but some engines react slower than others."
	RE: Why CrowdStrike had hazards for 3,739 files on a specific day	We added an explanation in the third paragraph of **Section 4.3**: "CrowdStrike has hazards on 3,739 files on day-175. After inspecting the update information on the corresponding three days (u1,u2,u3), we find that for most files, u1 is equal to u3, but u2 is much larger (both u1 and u3 are in 2018, but u2 is in 2019)."
Hazard & Flip	RA: Whether repeated API pulls can fix hazard flips	We indeed tested querying API multiple times a day. This test shows only very limited hazards can be fixed by repeated queries. We added this discussion in the first paragraph of **Section 4.3**.
	RE: in-depth reasons for hazards	We inferred the possible reasons for hazard flips in Section 4.3. We have already leveraged all the available information. Since VirusTotal engines are different from desktop engines, we do not think using desktop engines can help understand hazard flips on VirusTotal. We added this discussion in the last paragraph but two of **Section 4.3**: "We cannot use desktop engines to validate the non-determinism since VirusTotal engines are different from their desktop versions [13]."
	RE: Observation 5 is misleading? Only t=1 is too small	We emphasized that "50 out of 93 papers use $t = 1$, see Table 1)" in the second paragraph ("**Setting t=1.**") of **Section 4.5**.
Hash	RA: Do generated malware have varying hashes	Each ground-truth malware needs a new "file hash" so that VirusTotal would believe that the file is never scanned before (and thus triggers a scanning). We confirmed that VirusTotal did not have prior scan records for any of our ground-truth malware files. We further examined other malware signatures/hashes. We confirm that all malware instances have different SHA256 and ssdeep values. We added this discussion in **Section 3.2** (the

		last paragraph on **Page 4**): “these samples have different hash values (e.g., SHA256, ssdeep) from the seeds and can trigger VirusTotal’s file scanning after being submitted”.
Figures	RA: Figure 4 is not clear	We improved **Figure 4** by sorting the legend based on the line appearance in the figure.
	RD: Improve Figure 12 by changing it to ROC	Clarification: Yes, we also think it is a good idea to put TP and FP together. Such trade-off between TP and FP for each engine is already shown in Figure 13. Figure 12 is also the per-engine result but looks at a different aspect. For Figure 12, the purpose is to show TP and FP of different engines varies based on different datasets (e.g., whether the samples are obfuscated).
VM	RB: Can malware detect the VM environment in the desktop experiments?	Unlikely for our cases. Recall that we manually confirmed the malicious behavior of each malware, by running them in a VM. We indeed saw the malicious actions (encryption, showing ransom notes). The desktop engines are run in the same VM. We have emphasized this in the paragraph under the caption “**Experiment Setup.” in Section 6.3:** “We validated that our obfuscated malicious samples still have their malicious actions in the VM environment”.
Benign Set	RC: Extend the benign sample set using popular executables	This is a great suggestion. However, we could not add such results due to the limited time window for data collection, and the mis-aligned timestamp. We regarded it as a potential future direction in **Section 8**.