What do malware analysts want from academia? A survey on the state-of-the-practice to guide research developments

Marcus Botacin
botacin@tamu.edu
Texas A&M University
USA

ABSTRACT

Malware analysis tasks are as fundamental for modern cybersecurity as they are challenging to perform. More than depending on any tool capability, malware analysis tasks depend on human analysts' abilities, experiences, and practices when using the tools. Academic research has traditionally been focused on producing solutions to overcome malware analysis technical challenges, but are these solutions adopted in practice by malware analysts? Are these solutions useful? If not, how can the academic community improve its practices to foster adoption and cause a greater impact? To answer these questions, we surveyed 21 professional malware analysts working in different companies, from CSIRTs to AV companies, to hear their opinions about existing tools, practices, and the challenges they face in their daily tasks. In 31 questions, we cover a broad range of aspects, from the number of observed malware variants to the use of public sandboxes and the tools the analysts would like to exist to make their lives easier. We aim to bridge the gap between academic developments and malware practices. To do so, on the one hand, we suggest to the analysts the solutions proposed in the literature that could be integrated into their practices. On the other hand, we also point out to the academic community possible future directions to bridge existing development gaps that significantly affect malware analysis practices.

CCS CONCEPTS

• **Software and its engineering** → Software creation and management; • **Security and privacy** → **Malware and its mitigation**; *Human and societal aspects of security and privacy*.

KEYWORDS

Malware, Malware Analysis, Reverse Engineering, Analysis Tools

ACM Reference Format:
Marcus Botacin. 2024. What do malware analysts want from academia? A survey on the state-of-the-practice to guide research developments. In *The 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2024), September 30–October 02, 2024, Padua, Italy.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3678890.3678892

RAID 2024, September 30–October 02, 2024, Padua, Italy
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0959-3/24/09
https://doi.org/10.1145/3678890.3678892

1 INTRODUCTION

Malware is a major security concern nowadays and the academic literature is full of works presenting strategies to better perform malware-related tasks, from threat hunting [39] to triaging [33], and from machine learning training [21] to detection rule generation [57]. Despite the significant contributions academia made to the field, not all proposals made by academia are adopted in practice. In fact, many complain about academically-proposed ideas not being practical [12]. As a research community, we would like to cause the greatest positive impact possible, which in the malware analysis case means that people will be more protected if we can facilitate malware analysts' lives. Thus, we want to understand what malware analysts need in terms of scientific developments.

In this work, we present an analysis of the malware analysis practices and the developments proposed in the literature. Our goal is to help academic researchers guide their efforts toward more practical solutions and to help professionals find the best proposals that fit their real-world needs. To do so, we first rely on the available literature to systematize the practice of malware analysis, pointing out the challenges analysts face in their daily tasks. We identified key points related to the challenges that are not covered in the current literature works, such as prevalence rates for analysts facing certain conditions (e.g., malware variants), and how often they adopted tools proposed in the literature (e.g., graph-based binary comparison). Further, we developed a set of 31 questions to collect data about these previously not-characterized aspects. We systematized these questions in a survey (Appendix C) that was applied to 21 professional malware analysts actuating in different security fields–from CSIRTs to AV companies. Based on the analysts' answers, we prepared a second round (follow-up) of questions (Appendix D) to clarify any remaining imprecision. Upon it, we present a critical discussion on how to move the field forward.

Among our discoveries about the challenges for multiple malware analysis practices, we highlight that: (1) a significant part of malware analysis tasks are performed manually, such that developing automation mechanisms is a promising avenue to contribute to the field; (2) existing automation tools such as automatic tracing via public sandboxes are not enough because they are not configurable and do not provide fine-grained information about analysis outcomes, such that developing methods to explain malicious paths is key to streamlining automation procedures; (3) decompilers are very popular tools among analysts, but significant developments are still possible in this domain; and (4) Many State-Of-The-Art (SOTA) solutions described in the academic literature are still not widespread in practice, such that they need to be transitioned to practice to help analysts. We expect that pointing out and quantifying these limitations might foster future malware research.

In summary, this paper's contributions are as follows:

- We characterize the typical malware analysis workflow regarding multiple challenges.
- We present a survey on how 21 professional malware analysts tackle these challenges in practical scenarios.
- We point out existing development gaps and future development opportunities to bridge them.

This paper is organized as follows: In Section 2, we present the challenges in a typical malware analysis workflow; In Section 3, we present our methodology to survey how professional analysts tacklçe these challenges; In Section 4, we present a profile of the surveyed professionaçs; In Section 5, we discuss the analyses practices reported by the surveyed professionals; In Section 6, we discuss the surveyed professionals' opinions about existing analysis tools; In Section 7, we point out future directions on the field; In Section 8, we present related work to better position our contributions; In Section 9, we draw our conclusions.

2 WHAT DOES A MALWARE ANALYST DO?

We refined an existing Systematic Literature Review (SLR) on malware analysis [12] (see details in Appendix A) to systematize the tasks performed by analysts (Appendix B) and their challenges. We here present the existing Knowledge Gaps (KGs) to be bridged whenever an aspect is not properly addressed in the literature.

The Analyst Role. The Malware Analyst is the professional responsible for collecting, triaging, understanding, and reporting new threats. To provide insights about the threats, analysts employ multiple security analysis strategies, mainly reverse engineering [65]. Whereas reverse engineering is a critical step of the malware analyst task, professional malware analysts often perform other company tasks, such as training and recruiting new professionals. The time analysts have to analyze a sample is a key challenge to understanding threats, such that giving analysts tools that reduce their analysis load is key if analysts are in time-struggling conditions. **KG1:** It is currently unclear which fraction of the malware analyst job is dedicated to reverse engineering tasks.

The Analysis Group. Malware analysts might work in teams or individually [56]. When working in teams, analysis data must be shared between the analysts, such that developing collaborative solutions might increase analysts' productivity. **KG2:** It is unclear how often analysts work in teams and individually.

The Analysis Request. Whereas some analysts might start their own research endeavors, in most cases analysts are some company's employees who start analysis procedures based on the company's requests. Thus, the samples they have to analyze usually come from different sources, depending on the type of company. The context of infection might shape the attacker's strategies [11] and having information about that might facilitate analysis and proper threat identification. **KG3:** It is unclear how much knowledge analysts have about the context of the samples they analyze.

The knowledge to run analyses. Malware analysis requires specialized knowledge. For instance, analysts must have a solid understanding of binary internals. Ideally, even specialized knowledge like this should be widely available to enable the formation of new analysts [44]. The more analysts enter the field, the more analysis procedures tend to scale [40] and incident response tends to be faster. A main academic goal is to develop strategies to provide analysts access to this knowledge. **KG4:** It is unclear if the knowledge required to perform malware analysis is accessible to students.

The feeling about maliciousness. To understand an attack you have to think like an attacker. This famous saying summarizes well most of the analysis challenges faced by analysts. In addition to technical knowledge, malware analysts must develop intuitions about how malware behaves [10, 46]. Intuition can be developed over time, but seniority is a scarce resource for companies, such that developing tools to transition knowledge from seniors to juniors is a key research contribution. **KG5:** It is unclear how experience and expertise are distributed in malware analysis teams.

Keeping up with malware evolution. Malware evolves fast since attackers often develop new ingenious ways to bypass detection mechanisms. This fast evolution requires malware analysts to keep studying to stay updated with new attackers' practices [2]. Part of the skill updates of malware analysts comes from the practice and information sharing with other analysts and teams [49]. Part of the new knowledge might come from academia, which also produces vast material about malware analysis. Ideally, academics would like to apply the developed scientific knowledge to the practice of malware analysis. **KG6:** It is unclear how malware analysts keep updated and if academic knowledge reaches them.

Selecting representative samples to analyze. Whereas some malware samples easily reveal their tactics, other malware samples only reveal patterns when combined. Aspects such as the reuse of code and the evolution of techniques can only be characterized by analyzing a set of samples from a given family. Thus, in some analysis cases, the malware analyst might collect additional samples to analyze in addition to the one initially requested to expand the conclusions enabled by the analysis procedures. Understanding the sample's reuse is important to develop better solutions for the sample's correlation [17]. **KG7:** It is unclear how often and when analysts enrich their analyses with additional malware samples.

Saving samples for the future. As some analyses require additional samples, it is expected analysts store some samples they deemed interesting from previously to be eventually used in future ones. This procedure should happen in a structured manner, with the AV company storing the samples according to well-defined criteria. In practice, however, it might also happen that the AV analyst might store the samples him/herself, according to some personal criteria. Academic research could contribute to establishing guidelines and storage practices for the samples. **KG8:** It is unclear how often analyzed samples are stored by analysts and/or AV companies.

Selecting where to analyze. Different malware samples require different strategies to be analyzed, thus different tools. Even inside the same category of tools, there might be multiple possible solutions. For instance, whereas sandboxes are popular solutions, the characteristics of public [74] and private (e.g., AV-owned) sandboxes might make the difference for some types of samples. Also, different employers have different policies. Some allow the collected samples to be uploaded to public sandboxes, where they will be shared with a large community of researchers [25]. In turn, some companies cannot reveal information about infected customers, so they require the use of a private sandbox, with no data sharing. Understanding analysts' choices, requirements, and constraints is key for academia to develop better analysis platforms. **KG9:** It is unclear how often analysts opt for public and/or private sandboxes.

The variants that keep going. Malware samples do not always appear alone, they might also appear as variants [61], that impose extra analysis work. Ideally, variants should be filtered by good triage systems, but it does not always happen in practice, such that analysts might sometimes recognize constructions previously seen in other analyzed samples. If the re-analysis of variants is a significant problem, academic research should focus more on developing better triage strategies. **KG10:** It is unclear how often analysts inspect malware variants that could have been triaged.

The need for manual tasks. Although many tools are available for malware analysis, it is common that analysts need to make manual adjustments to tools and binaries to make malware samples run. Ideally, manual adjustments should be minimal, because manual work means scalability limits. In practice, however, analysts might experience different scenarios, with a large manual effort if tools are not appropriate [29]. It is a relevant academic research goal to overcome the limits of existing approaches to allow analysis to scale if it is identified as a bottleneck. **KG11:** It is currently unclear to which extent malware analysis requires manual work.

The malware analysis that becomes multiple. Modern malware is not unitary but is usually composed of multiple layers [13]. Each layer might be implemented using a different technology, which requires a different strategy [67] and thus tool to be inspected. Ideally, all malware stages should be analyzed in a smoothly integrated environment. However, integrating multiple technologies is hard, such that malware analysts might need to treat each malware stage as a totally new analysis step. Developing integration strategies to present a uniform view of code for multiple technologies is a significant academic research topic that must be boosted to help malware analysts if they struggle to analyze multi-stage malware. **KG12:** It is unclear how malware analysts handle multi-stage malware.

The hard tasks. Not all malware analysis tasks are equal. Malware creators use constructions to purposely complicate reverse engineering, thus tasks such as unpacking, deobfuscation, and finding execution triggers require knowledge and time from the analysts [59]. Once again, it limits scalability, as junior researchers might not have the expertise to do so and seniors are in limited numbers and time availability. If this becomes a bottleneck for the analysis process, academic research should focus more on developing solutions to automate unpacking [28], deobfuscation [62], and to make these tasks feasible for junior analysts. **KG13:** It is currently unclear how much malware analysts struggle with a lack of skills or lack of time to perform complex analysis tasks.

When to stop analyzing. Knowing when to stop analyzing a sample is as critical as knowing how to start analyzing it. There is an ideal amount of analysis. Analysts should not under-analyze the sample under the risk of missing hidden behaviors. They should also not over-inspect the sample, as it limits scalability with a task that does not produce new results [31]. To avoid under-analysis, analysts might re-run analysis procedures multiple times. Academic research might produce solutions to help identify the amount of information present in different traces to contribute to the identification of the ideal amount of analysis. **KG14:** It is unclear how many analysis runs are performed and what is their stop criteria.

Extending analyses to other environments. A natural derivation of the amount of analysis discussion is the amount of traces to consider. Sandboxes have their results widely tied to the environment, such that the execution in different sandboxes leads to a different amount of data [30]. Academic research might help identify how much information one can extract from a binary via new, proper metrics. **KG15:** It is unclear how often analysts use different sandboxes to analyze the sample.

Identifying when malware does not run. A key reason for analysts changing sandboxes is when a malware sample does not run (or evade) a given sandbox [8]. To overcome evasion routines, analysts have to change the sandboxes' default configurations. Academic research should provide solutions to automatically identify the root cause of evasions. **KG16:** It is unclear what analysts change in the sandboxes and how they choose which sandbox to use.

When having multiple traces, the analysts face the challenge of identifying which of the traces are correct and which information to consider from each trace [6]. Academic research should provide solutions to automatically compare them. **KG17:** It is unclear which strategy analysts use to compare traces from multiple sandboxes.

Waiting for analysis results. Some analysis procedures take significant time [31]. For instance, sandbox execution requires analysts to run samples for a few minutes, a time that is hard to reduce as the samples need to actually run. Tracing might become even slower when emulation layers are considered [22]. Academic research can focus on developing solutions for fast emulation and instrumentation to limit waiting time to the minimum value possible: the actual running time. **KG18:** It is currently unclear how much analysts are bothered by slow execution environments.

What to do with analysis results. A malware analysis does not stop when an analyst reaches a given program state in the debugger. Some might even say that is where the analysis begins. Analyses are valued by the outcomes they produce. The most common analysis outcomes are reports [69] and signatures [57]. Academic research could contribute to these steps by developing automatic summarization and signature writing tools, depending on analysts' needs. **KG19:** It is unclear how often analysts write reports or signatures.

The effort to write defenses. When writing signatures, analysts need to find a balance between matching capabilities and performance. In the first case, one wants to detect the biggest amount of malware possible. In the latter, one does not want to delay scans for a long. Writing performance-efficient signatures is hard (e.g., regex might end up in loops), such that academic research could provide solutions for effective signature writing [16]. **KG20:** It is currently unclear how much effort analysts put into controlling the performance of the generated signatures.

Selecting the best tool for each task. Different malware samples require different tools to be analyzed. Each tool implements a different analysis strategy/technique, thus they present different pros and cons [43]. Analysts end up developing a feeling on how and when to use each tool, but there is no guarantee that these are the best use cases possible. Academic research should provide formal guidance on tool evaluation and selection to better help analysts. **KG21:** It is currently unclear how much and when malware analysts use the different tools and techniques.

Setting up the analysis tools. More than selecting a tool to use, analysts often have to select complements to them. For instance, in addition to mastering a good debugger solution, analysts also have multiple plugins in their toolchains to complement the debugging experience. These plugins and extensions often add heuristics and analysis capabilities that are not native to the solutions. Some of the extensions present significant scientific challenges that would be worth investigating by academic research, such as creating different representations for the same data (e.g., assembly debugging vs. decompilation [14]). **KG22:** It is unclear how analysts rate the solutions they use and which ones they would like to have.

Preparing for the future. The goal of malware analysis is to prepare us for a more secure future. Thus, malware analysts are also preparing themselves for the future, as they learn with samples and also with training. Academic research should be a key partner in preparing analysts for the future, providing solutions for problems yet to come, and developing next-gen solutions. This should be ideally coupled with the malware analysts' needs and expectations. **KG23:** It is unclear what analysts expect for the future.

3 METHODOLOGY

We surveyed professional malware analysts to bridge the KGs presented in Section 2. We here present our survey methodology.

Recruitment. We reached out to malware analyst teams' leaders and asked them to share our survey invitation with their team members. We clearly expressed that filling out the survey was voluntary, optional, and not a job-related activity. We also reached out to individual malware analysts who cooperated with our research team in the past. Most invitations were performed by email. Some analysts reached out to us via social networks, through which we sent them the invitation. The recruitment consisted of a brief description of the project followed by a link to access the survey platform online. The survey has been open for responses from Jan-Mar/23.

Selection criteria. We included in our survey only professional malware analysts, thus maximizing representativity and reducing noise. The analysts are identified as professionals by their managers or our research team. Since we did not post any public link on any Internet webpage (as previous work did [70]), we ensured that our survey was distributed only inside the malware analysis community. Despite not publicly posting our survey, we received the same number of valid responses (21) as the largest previous work [73]. We show a replication study with additional participants in App. G.

The Survey. Our survey consisted of 31 questions (Appendix C) divided among multiple research areas. Whereas some questions were worth additional clarification, we tried to minimize the number of questions to maximize the likelihood of receiving complete survey responses. The survey is composed of alternative questions and open-response questions. We always placed open questions after alternative questions to try to minimize the introduction of biases. The survey was completely anonymous, but the participants had the opportunity to voluntarily deanonymize themselves. We identified cases of (i) participants who remained anonymous; (ii) participants who directly disclosed their identity; and (iii) participants who were indirectly and partially anonymized by their managers (sending us a confirmation message that their whole team has filled out the survey). The survey was approved by our university's IRB (2022-1327).

Follow-Up. Based on the compiled survey answers, we prepared a second survey (Appendix D) with questions aiming to obtain a better understanding of blurry points and to ask the respondents to better elaborate their open answers. This follow-up survey was sent only to the participants who voluntarily deanonymized themselves and thus agreed to receive additional questions.

Survey Limitations: This research presents the same limitations as any research with human subjects; Although we tried to minimize any suggestions effect, analysts might be biased towards responses depending on the writing of the question and different analysts might have different anchors for scaling their responses.

4 PARTICIPANTS

We here characterize the professional malware analysts who responded to our survey to position our work as relevant for professionals in the field and present the first evaluation of the field based on the characteristics of representative stakeholders for the field.

4.1 Professional Background

Where do the respondents work? Characterizing the responder's occupation is important to evaluate if our survey reached the target audience and thus ensure representativity. Although the filling of our survey was anonymous, we were able to identify the occupation of all survey responders (Table 1), as (i) some of them disclosed their identities; (ii) some of them indirectly disclosed their occupation when answering the open questions; or (iii) their managers confirmed that their teams answered the survey. We highlight that for these cases identifying one's occupation or organization does not imply that we identified the individual professional inside the organization that answered the survey.

Table 1: Analysts' occupation.

#	Role	Company	Obs.
1	CISO	Non-Security	
1	Threat Hunter	Intelligence Agency	
1	Leader	Government CSIRT	
1	Member	Bank CSIRT	
4	Consultant	Independent	Ex AV analysts
5	Analyst	Sec. Consultancy	2 companies
8	Analyst	AV company	4 AV companies

Table 1 shows that the survey achieved representants of multiple sectors (public and private). Naturally, the survey has a bias towards respondents working in AV companies, which is likely the main occupation role for malware analysts. The survey reached both professionals who currently work for AV industries as well as former AV analysts (now consultants). The survey also reached leaders of organizations, which also provided the survey the opportunity to hear professionals who work with multiple malware sources due to their positions. We conclude that the survey population is not representative of any particular scenario, but diverse enough to provide multiple views of the malware analysis scenario and thus characterize the general needs of malware analysts.

How much malware analysis do analysts perform? Characterizing the analysis tasks performed by the analysts is key both to evaluating representativeness as well as to understanding the role of the malware analysis job. Table 2 classifies analysts' responses to the job characterization question as: Full-time malware analysts; Most tasks are malware analysis; a Reasonable number of tasks are malware analyses; Eventual performance of malware analysis; or they Never perform malware analysis. We notice that no respondent never performs malware analysis, which shows that our survey actually reached professionals in the malware analysis field. However, we notice that the majority of analysts (86%) do not perform only malware analysis, although it might be their job title.

Table 2: Analysts' Malware Analysis Tasks Frequency.

Category	Full	Most	Reasonable	Eventual	Never
Answers	3 (14%)	5 (24%)	6 (29%)	7 (33%)	0 (0%)

We discovered that occupation is a moderate determinant factor for the amount of malware analysis tasks an analyst performs. The overall Pearson correlation between occupation and amount of tasks is 0.54. This shows that whereas occupation is important, the scenario also plays a key role. Whereas one could hypothesize that AV analysts would be busier with malware analysis, the correlation between only doing malware analysis and working for an AV company is only 0.24 (weak), as many consultants stated that they work full-time with malware analysis as well. If we consider both full-time and most-time classes, the Pearson correlation is 0.69 (high), because all AV analysts fit this category.

How are the malware analysis teams organized? Understanding the scenario in which malware analysts actuate is key to designing proper solutions. Table 3 categorizes analysts' actuation as: the analyst is part of a Team and they analyze the samples Together (TT); the analyst is part of a Team, but works Individually on samples (TI); and the analyst is an Independent Individual (II).

Table 3: Analysts' Type of Tasks vs. Analysis Teams.

Category	TT	TI	II
Answers	1 (5%)	16 (76%)	4 (19%)

We notice that most professionals (81%) are part of a team, which provides insights that collaboration tools are required to allow the professionals to share information. However, most of them (76%) perform individual tasks, such that real-time collaboration is not necessarily the goal of the tools to be developed. We observe no significant different distribution of responses among job occupations.

How much context about sample capture do analysts have? The amount of information an analyst has about the sample to be analyzed changes the way the analyst starts approaching the problem. Also, the analyst might be more familiar with the techniques used by malware known to come contexts than with malware for other contexts. Table 4 categorizes responses by collection type: Regional threats; Local threats; or the source is Unknown.

Table 4: Knowledge on Samples' Collection Context.

Category	Regional	Local	Unknown
Answers	11 (52%)	3 (14%)	7 (33%)

We notice the majority (52%) of participants report analyzing samples belonging to a specific region, which is somehow against our initial hypothesis that most analysts would analyze general malware. This finding reinforces the importance of developing regionalized studies about malware trends [11].

The second most popular (33%) analysis type involves the samples collected from diverse sources [68], thus with no infection context. This is the usual case for those working as an outsourced security team for other companies. The less popular scenario (14%) is the analysis of local samples (e.g., collected in the internal network). The small number of professionals in this category is explained by the fact that not all companies can afford a local security team, thus they end up outsourcing their analysis events, boosting the previously mentioned category.

The correlation between professionals working in CSIRTs and the analysis of local threats is 1.0, as one could hypothesize since it is the nature of the job. The correlation for AV company professional s is 0.61 with regional threats and 0.52 with collected samples. The correlation for consultancy professionals is 0.81 and 0.37 respectively.

Despite the significant statistical number pointing out the importance of regional datasets, one analyst reported that context does not matter for the reverse engineering process. In the follow-up, the analyst clarified that: *"The importance of the context depends on the goal. Context matters for incident response and threat intelligence, not for rule generation. I just detect it."*

4.2 Professional Skills

How did the analysts learn malware analysis? Malware analysts learn to analyze malware via different methods and understanding these methods is important to foster good training and education. Whereas previous works [70, 73] looked at participants' degrees, they did not assess specifically how and at each education level they learned the malware analysis skills. Whereas many professionals might have formal training in computer science, the aspects specific to malware analysis procedures might have been learned in different ways. Table 5 summarizes how this survey's participants learned malware analysis techniques. We made explicit in the survey question that the point was about learning malware analysis and not what were their highest degrees. The responses are categorized as: Post-Grad in the field (PG); a Major in the field; Industry Certification; Work Experience: or Self-Taught.

Table 5: Analysts' Strategies for Learning Malware Analysis.

Category	PG	Major	Cert.	Work	Self
Answers	2 (10%)	0 (0%)	0 (0%)	10 (48%)	9 (42%)

We notice that the majority of the analysts learned malware analysts by themselves (The ones that reported self-taught and working experience account for 90% of all cases). Only 10% of the analysts had formal, academic training in malware analysis. This result is in line with the scarce number of malware analysis courses available in most university's course catalogs [63] and it shows that the academia is not exploring its full education potential in the field. The lack of better educational strategies was voluntarily pointed out by the analysts as an aspect to be addressed to move the field forward (Section 7). The path to move forward is tied to increasing the security background of CS professionals. The difference between having CS background and having a malware background was well summarized by one of the analysts: *“You need to think about maliciousness. Nobody teaches you it“*.

How much experience do the analysts have? We previously seem that many malware analysts learned their professional while working, which reinforces the role of experience in the practice of malware analysis. Figure 1 shows the distribution of the years of experience the malware analysts reported in the survey.

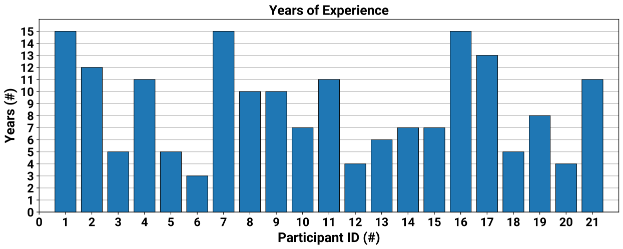

Figure 1: Analysts’ Years of Experience.

The respondent population is diverse in experience (ranging from 3 to 15 years), thus presenting multiple views of the malware analysis landscape. There is a trend toward more experienced researchers (The average analyst experience is 8.7 years). This expertise is distributed in all fields, with AV analysts being as experienced as professionals working in other companies, which shows that the role occupied by a professional is a more determinant factor than the nature of the industry. The correlation between years of experience and working in AV companies is low (0.28) because an industry in need of qualified professionals recruits the available ones regardless of their current experience.

How do malware analysts stay updated? Malware analysts use multiple information sources to stay updated. Table 6 shows how the surveyed participants stay updated with malware evolution. The table presents the rate of participants that mentioned each information source and how much the participants rate that each information source contributes to their update.

Table 6: Analysts’ Knowledge Updating Strategies.

Category	Academic Papers	White Papers	Videos	Events	Training
Answers	15 (71%)	21 (100%)	13 (61%)	18 (85%)	12 (57%)
Rate	14%	46%	11%	21%	12%

We notice that papers are the most important source of information to most analysts (60% of their updates are performed by reading academic or whitepapers). All analysts reported reading whitepapers to some extent. A significant but smaller fraction of the analysts read academic papers, thus showing that the distribution of academic knowledge still has space to grow among analysts, as they are already used to reading papers, but not academic ones.

5 ANALYSES

5.1 Analyses Practices

Do the analysts collect more samples to analyze? Enriching analyses with more samples might be key to the analysis outcomes. Whereas previous work [73] pointed out that analysts collect multiple samples, it did not measure the prevalence and motivation for this practice. We asked the analysts if and when they enrich analyses with more samples. Table 7 shows their response broken down by the motivating reasons: Writing `Signatures`; Writing `Reports`; `Understanding` techniques; or `No` collection.

Table 7: Additional Samples Collection by Analysts.

Category	Sig.	Report	Understand	No
Answers	8 (38%)	2 (10%)	7 (33%)	4 (19%)

We observe that most analysts (81%) collect additional samples for their analysis procedures. This shows that developing tools to capture similar samples and correlate them is an important step ahead for this industry. Analysts mainly collect samples to write signatures and to understand malware samples‘ techniques. Whereas the reported proportions are similar, the nature of the problems is very different. Whereas signature generation is a generalization effort, the understanding of technique is about the development of new knowledge. This shows that developments in the field should be in complementary directions [12].

How are analyzed malware samples stored? We asked the analysts if and when they store the analyzed samples. Table 8 shows responses broken down by the reasons for storing the samples: They are `Often` stored by the `Company` (OC); They are `Often` stored by the `Analyst` (OA); They are `Sometimes` stored by the analyst, but only as a `Curiosity` (SC); and They are `Never` stored (N).

Table 8: Samples storage

Category	OC	OA	SC	N
Answers	9 (42%)	5 (24%)	5 (24%)	2 (10%)

We notice that in the majority of the cases (66%), the samples are stored by either companies or analysts for reasons that are not simple curiosity. The analysts report the most common use case is to use the samples to enrich further analyses, which complements the previous response. It shows that developing tools to correlate and search samples [24, 58] is key for malware analysis.

Who hosts analyses procedures? Whereas some analysis strategies might be well-known, the environment they are performed might change the challenges they present to the analysts. To better understand that, we characterize the hosting of the analysis solutions. Table 9 categorizes the hosting entities as: Analyst’s `Own` machine; `Public` infrastructure; or `Company`’s internal hosting.

We notice that the majority (85%) of the analysts run most of their analysis procedures on their own machines. This finding should shape the way research is performed because tools should not be developed to run in big cloud servers, but they must be able to run on more constrained devices. In this sense, a good design rationale is presented by solutions such as the laptop-based Yara signature generation procedure [57], that states that *“the tool must be lightweight enough that it can run on low resource machines (e.g., a laptop with 4 GB of RAM or less) to support the maximal number of analysts, who do not always have significant compute resources available.”*

Table 9: Analysis hosting.

Category	Own	Public	Company
Answers	18 (85%)	1 (5%)	2 (10%)

An important reason mentioned by analysts to not rely on public infrastructure (e.g., public sandboxes) is that some AV companies do not allow the sharing of the samples with the community, the default option in these services [71]. AV companies tend to offer their own sandbox solution to analysts. All (100%) of the analysts who reported using a company-provided sandbox work for AV companies. On the other hand, public CSIRTs tend to not have restrictions for sharing samples [26]. The only individual reporting to perform most tasks on public service works for a CSIRT.

What are the malware analysts' relation to public sandboxes? The previous finding about the use of self-host analysis procedures led us to investigate the malware analysts' relation to public infrastructure. We asked specifically about their relation to public sandboxes, as it is likely the most popular type of public malware analysis tool. Table 10 shows responses broken down by analysts' opinions about these services: if they like them or not; and if their use is allowed or not by their employees. We notice a divide between those who like (52%) and do not like/use (48%) these services. Whereas the case of analysts disallowed to use these services has already been discussed in the previous question, there was still a remaining question: Why are most analyses self-hosted even for the analysts that report to like public sandboxes?

Table 10: Use of public sandboxes.

Category	Like	Dislike	Disallow
Answers	11 (52%)	6 (28%)	4 (20%)

The answer to this question is a frequent complaint of analysts about these services: the lack of configuration [42]. Most of these public services do not allow configuring the analysis environment at a fine-grained level, a task that is left for their self-hosted tools. We also notice that the type of analysis task performed by the analyst also shapes their opinion about the sandboxes. In the opinion of an analyst: *"They work for detection but not for response"*, Detection tasks might require less environmental configuration. The simple fact of observing an IOC might be enough for detection, whereas it might not suffice for understanding the sample. As another analyst said: *"Automated sandbox is just for the high-level, not to get the details. It saves time to show what I should look for"*. In this sense, a frequent complaint is that sandboxes do not allow the analysts to understand how the malware did a given action. Explaining malicious paths [45, 47] is a key open academic research problem to be addressed in the malware analysis domain.

How often do malware analysts recognize malware variants? We asked the analysts how often they notice that the samples they receive to analyze are variants of some malware samples they previously analyzed. Table 11 shows analysts' responses broken down by frequency. We notice that most analysts (52%) often receive variants to analyze and only a single analyst reports rarely receive malware variants to analyze. This result shows that the triage mechanisms can still be improved. The identification of similar constructions is still a significant research challenge [32].

Table 11: Malware Variants Re-Analysis Rate.

Categories	Very Often	Sometimes	Rare
Answers	11 (52%)	9 (43%)	1 (5%)

How much of the analyst's work is manual? We asked analysts to rate how much of their work involves manual analysis. Table 12 shows the survey's results broken down by the amount of work: `Fully`-automated analysis; `Half`-automated analysis; and Mostly-`Manual` analysis. We notice that none of the malware analysts reported the use of fully automated solutions to malware analysis, which suggests that the investigation of strategies to fully automate malware analysis tasks is a significant goal for academic research. The limited use of automation reported by analysts corroborates the impressions of the industry on the sector [51].

Table 12: Analysis Automation Rates.

Category	Fully	Half	Manual
Answers	0 (0%)	11 (52%)	10 (48%)

We observe a practical divide (52%-48%) between researchers who use some automation and those who mostly do manual work. In both cases, however, manual analysis is required to some extent, which allows us to conclude that all samples (100%) require manual intervention. Ideally, analysis solutions should be more automated to alleviate the analyst's load and scale the procedures. Also, automation tends to reduce errors, as it makes procedures more deterministic. A main explanation for the lack of full automation is the lack of configuration support in many sandboxes, which forces analysts to configure their own environments.

Whereas the industry is in favor of automation and many analysts recognize that their job involves huge manual efforts, not all analysts are in favor of automation. One analyst reported that *"I like automation, but I don't trust it. I recheck everything"*. This claim is not a suspicion about the companies providing sandbox, but a sandbox engineering perspective clarified in the follow-up survey: *"Automated everything is problematic. Sandboxes do not reach 100% coverage because of the arms race. The maintenance work is high to keep up with new TTPs. It is not sustainable many times."*. The challenge to keep up with automation is also reflected in the limits pointed to many tools: *"Most automation tools are useless because they do not cover variants–ex: plugins for single families"*. The development of strategies to easily expand automation tools is a significant research challenge to be addressed by academia.

How do malware analysts handle multi-stage malware? The lack of automation identified in the previous response might pose significant extra work especially in cases where the malware samples have multiple attack layers. We asked the analysts how they handled these cases. Table 13 shows the responses broken down according to the number of used tools and their integration: `Multiple` tools `Automatically` integrated (MA); `Multiple` tools, treating stages `Individually` (MI); `Multiple` tools, `Manually` pasting data from one to another (MM); with a `Fully-Automated` (FA) tool; or if they only analyze `Single-Stage` (SS) malware.

Table 13: MultiStage Handling.

Categories	MA	MI	MM	FA	SS
Answers	1 (5%)	14 (66%)	4 (19%)	2 (10%)	0 (0%)

The lack of automation is reflected in the handling of multi-stage malware. Most tools (90%) do not handle the whole analysis of multi-stage malware, even though all analysts report analyzing multi-stage threats. only 2 are fully automated. In most cases (90%), sample analysis requires manual intervention to some extent. This indicates that the development of stage-aware mechanisms by academic research is a valuable contribution to the field. Among the different strategies, the majority of the analysts (66%) handle analysis individually, treating each stage as a new analysis every time, whereas another portion (19%) analyzes stages individually but within the scope of the same analysis. Whereas treating stages individually is more laborious, it has the advantage of allowing future data correlation, such as spotting the same analyzed stage as the payload of future malware samples. It reinforces the need for developing tools [9] to correlate data from multiple analyses [20].

Are analysts' skills enough to handle complex malware? The lack of automation tools adds to the analyst the burden of conducting complicated tasks, such as unpacking, deobfuscating [59], and finding evasion triggers, all of which require expert knowledge. Whereas previous work [73] already pointed out the need for analysts to overcome malware evasion strategies, it did not measure how much analysts struggle to do it. We asked analysts to rank how much they struggle to perform these tasks when considering their current malware analysis skills. Figure 2 shows the fraction of the samples the analysts struggle to perform each task against.

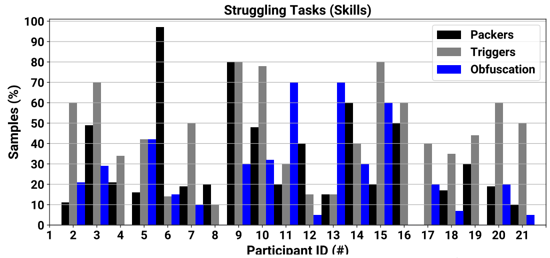

Figure 2: Analysts' Most-Struggling Tasks (Skill-Wise).

We notice that there is wide variation among the responses (ranging from 0 to 97%). On average, the analysts struggle in 32% of the unpacking cases, 45% of the triggering cases, and 23% of the deobfuscation cases. Whereas the numbers for the individual tasks are moderate, we need to remember that the same sample might require analysts to perform all these tasks. In this sense, 13/21 (61%) of the analysts struggle with at least one of these tasks for more than 50% of samples. This finding shows that a significant direction for academic research is to investigate how to facilitate (e.g., automate [37]) the performance of these complex tasks.

We found no significant correlation between struggling at any task and occupation, which indicates that analysts in AV companies struggle as much as analysts in consultancy companies. The correlation between unpacking and expertise years is only moderate (0.37), thus showing an intrinsic difficulty in the problem [1, 67] that is not only related to having previous contact with the packer or not. We did not find relevant correlations for other factors.

Is analysts' available time enough to handle complex malware? Limited automation affects analysts in multiple aspects. They not only struggle to have the skills required to analyze malware but also to have enough time to perform the required tasks, even when know what to do. Many malware constructions require analysts to perform repetitive tasks to be overcome, which reduces the analysis throughput. We asked analysts how much they struggle to perform the same 3 tasks but now focusing on their available time. Figure 3 shows the response distribution as the fraction of the samples the analysts struggle to perform each one of these tasks. Once again, wide variation is observed, with responses ranging from 0 to 80%. 90%, and 70%, respectively.

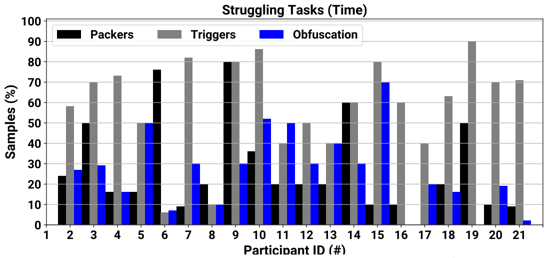

Figure 3: Analysts' Most-Struggling Tasks (Time-Wise)

Timing is revealed to be an important bottleneck for malware analysts. The number of analysts that struggle to have time to perform one of the tasks for more than 50% of all samples is 17 (80%) in comparison to the 13 (61%) that struggle due to the required skills. The average number of samples analysts struggle to have time to analyze is 26%, 56%, and 26%, respectively, in comparison to 32%, 45%, and 23% from skills. Packers requiring more skills than time to analyze is explained by the fact that although understanding the packer requires specialized skills, the unpacking process itself can be automatized via scripts written by the analysts. In turn, automating triggers required both deep inspection of the code as well as the significant dedication of time, as every sample tends to use different triggers [18]. Thus, we conclude that automating malware analysis tasks is key not only because it lowers the entry barrier to the field, as in the case of packers, but also because it gives scale to the process. The development of efficient solutions for automatically identifying malware triggers is still an open problem in the academic literature. The recent development in symbolic execution although promising [7, 53] is still impractical [15] to present time gains to the analysts. We did not find any correlation between time-struggling and years of expertise, thus showing that time is a constraint for all analysts.

5.2 Analysis Accuracy

How many times do malware analysts typically run samples? The limited automation also raises concerns about the correctness of the analysis results. Malware analysis is an error-prone task, such that it is easy to overlook some aspects. In this scenario, it is key to understand how analysts ensure the accuracy of their analysis results. One way to ensure correctness is to re-run analysts to verify if the results are reproducible. We asked the analysts how many runs they performed when analyzing malware samples. Table 14 shows responses broken down by frequency: `Always One` (A1); Typically one, but `Sometimes More` than one (SM); `Typically` a `Couple` runs (TC); and `Always` a `Couple` runs (AC).

We notice that only 24% of the analysts always run analyses multiple times, as a standard practice. Another 38% of the analysts

Table 14: Number of Typical Analysis Runs.

Category	A1	SM	TC	AC
Answers	0 (0%)	8 (38%)	8 (38%)	5 (24%)

report that they typically run more, but not always. This result shows that many analysts are aware of the possibility of multi-path malware, but do not have an established investigation method, basing the investigation of additional samples on their "feeling" about the need for additional inspection. Malware analysts would benefit from the development of solutions that point out the need for additional investigation and metrics to evaluate if the malware was properly explored (e.g., coverage [35]).

Do malware analysts test different sandboxes? We asked analysts if they run the samples in the same or multiple sandboxes to confirm the analysis results. Table 15 shows results broken down by usage frequency: `Always One` sandbox (A1); Typically one, but `Sometimes More` than one sandbox (SM); `Typically` a `Couple` of sandboxes (TC); and `Always` a `Couple` of sandboxes (AC).

Table 15: The Use of Different Sandboxes by Analysts.

Category	A1	SM	TC	AC
Answers	1 (5%)	8 (38%)	9 (42%)	3 (15%)

Most analysts (95%) are aware that malware behavior might differ from one sandbox to another and that analyzing in multiple sandboxes might be an interesting strategy to spot diverting behavior. However, only a few of them (14%) have multiple sandbox executions as an established methodological practice. In comparison to the previous case, fewer analysts run multiple sandboxes in comparison to the number of analysts who run samples multiple times in the same analysis session. An important academic contribution would be to formalize guidelines and criteria for the validation of malware results via structured methodologies [48].

What do malware analysts change in the analysis environment? Previous results pointed out that analysts: (i) want more fine-grained configuration in their sandboxes, and (ii) do not have uniform criteria to decide to pursue deeper samples investigation. We then question what analysts want to change in the sandbox environment and if there is a uniform criterion for that. We asked the analyst about the most typical changes sandboxes enable to understand if and how they are used. Table 16 shows the results broken down by motivation: Changing `Operating System` (OS); `Architecture`; `Both`; or `None`.

Table 16: Environment Configuration by the Analysts.

Category	Both	Arch	OS	None
Answers	5 (24%)	2 (10%)	0 (0%)	14 (66%)

Most analysts (66%) do not vary any of these most popular sandbox parameters. Instead, they prefer to change the sandbox solution as a whole to verify if different results appear due to the use of different collection mechanisms. Among the analysts that change the sandbox settings, most (24% of total) change both OS and architecture. A few of them change only the architecture. They report being particularly concerned about x32 vs. x64 malware. The emerging threat of multi-platform malware [34] was voluntarily expressed by one of the analysts as a concern for the future (see Section 7).

What multipath exploration strategies do malware analysts use? Previous results pointed out that analysts are aware that malware samples might have multiple execution paths. Even though there is no uniform way of checking for that among analysts, they end up doing this verification in some cases. Thus, we would like to understand which tools and technical strategies analysts use when they do so. Table 17 shows results for both the rate of analysts that responded to have performed multipath exploration at least once and the frequency in which the tools were used in these cases.

Table 17: Most-Used Path Exploration Strategies.

Category	Fuzzing	Symbolic	Concolic	Forced	Manual
Answers	9 (42%)	7 (33%)	5 (23%)	14 (66%)	19 (90%)
Rate	35%	41%	29%	49%	73%

As one could hypothesize due to the limited level of automation, most analysts (90%) relied upon manual sample inspection to discover new paths. No technique was used by the analysts in more than 50% of the samples. This result shows that many of the techniques reported in the literature for multipath exploration still need to be transitioned to practice and popularized among analysts to cause a real impact in the malware analysis field. Transitioning research to practice is very important because although many of these techniques are well-described in the academic literature [7, 35, 55], no commercial sandbox solution implements them.

How do malware analysts compare multiple traces? When malware analysts find multiple paths in a sample or run the same sample in multiple sandboxes they need to compare the results to give a final verdict. We asked them how they perform such a comparison. Table 18 categorizes strategies as: considering `All Versions`; based on `IOCs`; or comparing their `Graphs`.

Table 18: Most-Used Trace Comparison Strategies.

Category	All Traces	IoCs	Graphs
Answers	6 (28%)	13 (62%)	2 (10%)

Most analysts (62%) report base their comparison on the IOCs [3], i.e., if the IOCs are different in the multiple sandboxes or tools, then the sample has hidden paths. Comparing only the IOCs facilitates the analyst's work because it allows comparison to be performed at a high level. The comparison of malware traces at a low level (e.g., instruction level) is still an open research problem. Another significant part of the analysts (28%) does not verify differences in the traces. Instead, they just consider all of them together. When signatures are written, they write signatures to cover all the traces at the same time, without differentiating specific root causes of divergence. Only 10% of the analysts reported using some graph algorithm. Graph-based program comparison is an aspect widely studied in academic research [41, 54, 60] and that has the potential to help the practice of malware analysis if properly transitioned.

5.3 Analysis Procedures Performance

How fast are the tools used by the analysts? Previously, the analysts revealed complaints about the configuration of popular solutions like sandboxes, what do they think about the performance of these solutions? Also, since the majority of the analysts run tools on their own machines, are the performance of these tools compatible with the analysis requirements? We asked analysts to evaluate the performance of the tools they use. Table 19 shows the responses broken down by analysts' opinions: Tools are `Slow`, and could be `Faster` (SF); Tools are `Slow`, but this is `Intrinsic` to the malware analysis problem (SI); or the tools are `Fast Enough` (FE).

Table 19: Analysts' Perception About Tools Performance.

Category	SF	SI	FE
Answers	10 (47%)	3 (15%)	8 (38%)

We notice that the analysts are divided, which leads to two different but complementary conclusions: On the one hand, most analysts (62%) believe that the tools are slow, which suggests that increasing the performance of analysis tools would be very appreciated by the field, thus being an important research direction. On the other hand, if we consider that 15% of the analysts believe that the slowness is inherent to the tool and that 38% believe that the tools are already fast enough, we have that 53% of the analysts do not envision significant changes toward performance improvement. In sum, performance seems to be a desirable but secondary characteristic of the analysis tools for most analysts.

How useful would it be for malware analysts to have faster sandboxes? The practical divide found in the previous answer requires a deeper investigation to understand what are the analyst's impressions of performance. To gain more insights about performance characterization, we asked analysts how useful it would be to them to have faster sandboxes. We investigated sandboxes because it is likely the most popular malware analysis solution and there is rich academic literature on the topic. Table 20 shows the responses according to the usefulness levels: `Very` Useful; Useful only in `Specific` cases; or it makes `No Difference`.

Table 20: The Usefulness of Faster Sandboxes.

Category	Very	Specific	No Diff
Answers	10 (48%)	11 (52%)	0 (0%)

All analysts agree that faster sandboxes would be beneficial, but they disagree to the extent. The practical divide (48%-52%) remains between those who believe it is very useful and useful only in specific cases. The correlation between those who previously answered that tools are slow and the ones who now reported that faster sandboxes would be very helpful is high (0.76).

5.4 Analysis outcome performance

What are the typical analysis outcomes? We asked the analysts about the types of outcomes of their analysis procedures. Table 21 summarizes the results. We notice that producing reports is a key malware analysis task, being performed by 90% of the analysts.

Table 21: Most-Frequent Analysis Outcomes.

Category	Both	Reports	Signatures
Answers	10 (48%)	9 (42%)	2 (10%)

Another 58% of all analysts also produce signatures, which was revealed to be an important procedure. The importance of signatures for malware analysts highlights the fact that this task is not often covered in the academic literature. Whereas multiple proposals tackle malware analysis and detection via machine learning, automatic signature generation still needs to be more studied and developed by academia. We found no correlation between signature writing and working in AV companies, such that analyst working in other types of companies also write signatures and write reports.

How important is performance in the writing of signatures? Malware analysts have to make multiple project decisions when writing signatures. We asked the analysts how important is performance when writing signatures. Table 22 shows the responses broken down by the importance reported by the analysts: `Same` priority for accuracy and performance; `Accuracy` comes `First`; or `Only Accuracy` matters. We notice that most analysts (80%) care about performance, such that efficient signature matching should be an important academic research topic (e.g., efficient YARA matching [16]). However, as for tool performance, a significant part of the analysts (67%) consider performance a secondary aspect, not only for the tools they use but also for the signatures they write.

Table 22: Required Properties for Signature Generation.

Category	Same	Acc. First	Only Acc.
Answers	7 (33%)	10 (47%)	4 (20%)

We highlight the fact that even analysts who reported in the previous question to not write signatures were allowed to answer about the characteristics of the signatures they write. The interpretation for that is that whereas in the previous question they were answering about their typical experience, in this question they answered about their experience as a whole, which might have involved the writing of signatures in other epochs.

6 TOOLS EVALUATION

What are the tools most used by malware analysts? The tools used by the analysts significantly shape the analysis practices and they are also the most direct way research outcomes can be transitioned to practice. Thus, it is key to understand which tools are used by the analysts. We asked the analysts to rank the usage of different tools. Table 23 shows the rate of analysts that reported using each tool and the rate of samples analyzed using them.

Analysts' responses ranged from 0% to 100% for all tools but AntiViruses (AVs). The use of AVs illustrates how the use of tools is shaped by the scenarios. The only professionals to largely use AVs in the analysis procedures were the CSIRT analysts (correlation=1.0). This use is explained by the need for collecting information (e.g., labels) for incident response. In turn, AV analysts rarely use AVs in their investigations (correlation=-0.95). This is explained by the fact that the AV analysts are creating the detectors themselves.

The most used solution used by analysts is a disassembler. Interestingly, this is not the most studied subject in the academic literature, even though it presents multiple challenges [4, 52]. The second most popular solutions are decompilers, another class that presents key challenges [14, 72] to be addressed by future research.

How do malware analysts rate debuggers? We delve into the details of the different tools to understand how academic research

Table 23: Tools Usage.

Category	Similarity Hash	Debugger	Sandbox	Decompiler	Unpacker	AntiVirus	Disassembler
Answers	16 (76%)	18 (86%)	20 (95%)	19 (90%)	19 (90%)	11 (52%)	20 (95%)
Rate	47%	57%	58%	61%	49%	58%	66%

can be transitioned to the practice of malware analysis. We started our investigation with debuggers, one of the most popular tools in this field. We asked analysts how they rate the use of debuggers. Table 24 shows analyst's responses broken down by the level of satisfaction: (i) they are essential but require the analyst to perform `Repetitive` tasks; (ii) they are essential and `Enough` for the analysis procedures; or (iii) they are `Not Essential` for malware analysis.

Table 24: Analysts' Perception about Debuggers Usefulness.

Category	Repetitive	Enough	Not essential
Answers	15 (71%)	4 (19%)	2 (10%)

Most analysts (90%) perceive debuggers as essential for malware analysis at some level. Although essential, 71% of the analysts believe that the debuggers could be better, especially in avoiding repetitive tasks. Whereas the academic literature has multiple proposals for new debuggers, only a few of them are focused on malware analysis [27, 50], and most do not focus on usability for analysts, an open development field.

How do the malware analysts rate the role of debugging plugins? Since the analysts previously reported that debuggers can be enhanced, we investigated how they rate the use of plugins. Debugger plugins are popular for malware analysis because they increase usability, analysis capabilities, and reduce repetitive tasks. Developing plugins will be the most straightforward manner to transition academic research to practice. We asked analysts to rate the use of plugins. Table 25 shows responses broken down by the satisfaction level: Plugins are `Essential` for malware analysis; Plugins help in `Specific` cases; or Plugins make `No Difference`.

Table 25: The Role of Debugger Plugins for Malware Analysis.

Category	Essential	Specific	No Difference
Answers	9 (42%)	12 (48%)	0 (0%)

We notice that all analysts (100%) agree that plugins help to some extent. However, in another practical divide (42%-48%), most analysts believe that plugins help only in specific tasks, whereas many ones can be performed using the native debugging capabilities. The correlation between those who believe that debuggers require repetitive tasks and those who believe that plugins are essential is significant (0.62), thus explaining the different views analysts have about debuggers. An analyst summarized the relation with plugins as: *"Plugins are amazing, but they should be part of the tool."* and exemplified: *Current plugins to disasm Go or LLVM are non-native"*. This shows that the path to move the field forward in the analyst views is to integrate features from the plugins into the debuggers designed specifically for malware analysis.

How do the malware analysts rate the role of decompilers? Since analysts referred to decompilers as one of the most useful tools for their tasks, we asked the analysts to better detail their experience with decompilers. Table 26 shows responses broken down by the satisfaction level: Decompilers are `Very` Useful; Useful in a `Minor` part of the cases; or `Not Useful` at all.

Table 26: The Role of Decompilers in Malware Analysis.

Category	Very	Minor	Not Useful
Answers	17 (81%)	4 (19%)	0 (0%)

All analysts agree that decompilers are useful tools for their tasks. Most (81%) agree that decompilers are very useful, which is somehow surprising since current decompilers still present many drawbacks [14], that must be addressed by academic research. We can understand this result as the benefit brought by a solution being proportional to the complexity of the problem they address, i.e., decompilation is hard, but reading assembly code is also hard for humans [19], so even small decompilation advancements cause significant advances in malware analysis.

A known limitation of most decompilers is in the generation of functional code. In many cases, the decompiled code is readable but not compilable. Whereas this is a major problem for multiple fields, malware analysts explain that this drawback has to be put in the context of the malware analysis task: *"It is useful to get a snippet code from malware to use externally. For that, function correctness is important, but whole code correctness is not."*. In this sense, academic research developments might focus more on making parts of the decompilation functional [14].

On the exception side, those who do not consider decompilers very useful justify their preferences on the amount of information displayed by the disassemblers. As explained by one of the analysts: *"In most cases, the disassembly is enough. I use decompilers only to speed up analyses, but they are not really required to understand the malware. The disassembly tells me more information, such as calling convention and use of XORs"*.

What new tools would malware analysts like to exist? Once we identified the limits of existing tools for the practice of malware analysts, we gave a step ahead and asked the analysts which tools they would like to exist to help with their tasks. The complete answers are presented in Appendix E.

We classified analysts' answers in two groups: (1) Engineering Solutions., i.e., solutions that require implementation developments, but whose basis are already established; and (2) Scientific Solutions, i.e., solutions that require new knowledge development, even though their final outcome might be a product. Analysts' requests fitting these two categories highlight that not only academic research need to be transitioned to practice, but also there is room for improvement on the existing tools used by the analysts.

The common source for the need for new scientific developments is the request for more automation. Analysts request automatic ways to (1) Configure symbolic executors; (2) Select Sandbox Hooks; (3) Dump memory artifacts; and (4) Identify functions in binaries. All these tasks require new research developments because they all require the bridge of the semantic gap between the data collection and the data interpretation. In other words, they are cross-domain applications. These solutions require not only being able to handle the malware sample but also to infer what the analyst considers as "interesting" about them. The emerging AI developments have a significant potential to address these challenges.

7 FUTURE INSIGHTS

This work's goal is to identify which developments academic research needs to make to contribute to the future of malware analysts. We asked analysts' opinions, as follows.

What do malware analysts think the role of AI in malware analysis is? We started our investigation with the use of Artificial Intelligence (AI), since it is a growing trend in the security field and it was voluntarily expressed by some analysts as an important concept. We asked the analysts what will be the role of AI in the future of malware analysis. Table 27 shows the responses categorized by the reported impact: AI `Solves` the problem; AI `Helps` to solve, but not completely; or AI causes `No Change` to the field.

Table 27: Analysts' Impressions about AI usage.

Category	Solve	Help	No Change
Answers	1 (5%)	19 (90%)	1 (5%)

We notice that most analysts recognize the potential of AI but are cautious about their real impact. Only one analyst thinks that AI will have no impact in the field. Also, only one analyst thinks that AI will completely solve the malware analysis problems. Most analysts (90%) think that AI will help, but not completely solve the problem. The major reason for that pointed out by the analysts is that they think someone will still have to feed and train AI models and discover new malicious features (their thoughts are detailed in the next question).

What do malware analysts think about the future of malware analysis? The analysts freely commented about the multiple aspects of malware analysts they considered important for the future. The complete analysts' responses are presented in Appendix F. We here summarize the reasoning flow of most analysts as follows:

1. **From analysts to intelligence.** Multiple analysts mention that analyzing individual malware samples is not enough. The analysis products must evolve from individual analysis to intelligence (e.g., data correlation) to allow the detection and prediction of attacks at scale. AI plays a key role in it.
2. **AIs will not replace humans.** Multiple analysts agree that AI has a key role in moving the field forward, but most of them agree that human analysts are still required. The main reason for that is the need to update AI with the latest attack movements, that only humans can spot.
3. **We need better education of human analysts.** The analysts that will feed AI with data must be well-prepared to recognize unknown constructions. Most analysts agree that we need better training since now.

8 RELATED WORKS

Human-Focused Cybersecurity. The ultimate goal of any cybersecurity system is to protect the human stakeholders affected by the given system. Whereas cybersecurity research has initially put an almost exclusive focus on technical aspects of the studied system, the field has made progress toward the investigation of the human aspects involved in the system's operations, which is key for addressing security risks in a broad sense. It is essential to acknowledge how human factors play a key role in the security field. Even concepts such as malware, this paper's topic, are not agnostic to human definitions, as shown by recent research that investigates how humans and machines classify malware differently [5].

Malware analysis is a task that largely relies on human decisions and each human being approaches reverse engineering problems very differently [70]. A study showed that the decisions made by reverse engineers during their (non-malware analysis) reversing tasks are varied to the point of some preferring to start analysis onwards whereas others prefer backwards [36]. This same behavior diversity is likely to be found in malware analysts reverse engineering malware samples. Understanding how humans perform security tasks is not only important as a way of developing base knowledge on how humans operate, but it can also lead to concrete improvements to practical tools. For instance, evaluations in how humans decompile [19] can be used to develop improvements on the readability of code produced by decompilers [23].

Malware Analysis Landscape. Whereas investigating the malware analysts' practices is key to enhancing them, a few works present comprehensive evaluations of the current malware analysis landscape. An industry survey [51] recently presented findings to support that security companies need more automation in their malware analysis practices. Whereas this result provides some interesting direction for research in the field, it does not cover the needs of human analysts, as they were not individually interviewed. In this sense, the closest work to ours is the survey with 21 malware analysts [73] that investigated their malware analysis practices, such as how they use sandboxes. Whereas providing valuable insights about the field, the work does not focus on the tools the analysts would like to be developed to help in their practices.

Security Tools Landscape. Whereas evaluating how security experts interact with state-of-the-art security tools is essential to identifying how to make the tools better, the current literature is limited in evaluation reports. Our literature review found: (i) an analysis of offensive tools capabilities [66], which certainly provides some general insights but that lacks usability evaluations; and (ii) an HCI-focused evaluation of reverse engineering tools [38] that identified several limitations in existing tools (e.g., lack of analysis methods selectors) but that is not focused in malware analysis.

9 CONCLUSION

We investigated the practice of malware analysis by surveying 21 professional analysts and presented a critical analysis of the survey results to pinpoint current challenges and existing development opportunities from a research perspective. We discovered that: (1) Most malware analysis tasks are performed manually, such that developing automation mechanisms is a promising avenue to contribute to the field; (2) existing automation tools (e.g., public sandboxes) are not enough because they are not configurable and do not provide fine-grained information about analysis outcomes, such that developing methods to explain malicious paths is key to streamlining automation procedures; (3) decompilers are very popular tools among analysts, but significant developments are still possible in this domain; and (4) Many SOTA solutions described in the academic literature are still not widespread in practice, such that they need to be transitioned to practice to help analysts.

ACKNOWLEDGMENTS

We thank all the survey responders for sharing their invaluable time and knowledge with us. We also thank all members of our professional network who helped to distribute the survey among their malware analysis teams. Finally, we thank NSF for the support via the CNS 2327427 grant.

REFERENCES

[1] Hojjat Aghakhani, Fabio Gritti, Francesco Mecca, Martina Lindorfer, Stefano Ortolani, Davide Balzarotti, Giovanni Vigna, and Christopher Kruegel. 2020. When malware is packin'heat; limits of machine learning classifiers based on static analysis features. In *Network and Distributed Systems Security (NDSS) Symposium 2020*, Vol. 1. IFIP, US, 1.

[2] Olusola Akinrolabu, Ioannis Agrafiotis, and Arnau Erola. 2018. The Challenge of Detecting Sophisticated Attacks: Insights from SOC Analysts. In *Proceedings of the 13th International Conference on Availability, Reliability and Security* (Hamburg, Germany) *(ARES 2018)*. Association for Computing Machinery, New York, NY, USA, Article 55, 9 pages. https://doi.org/10.1145/3230833.3233280

[3] Bio Akram and Dion Ogi. 2020. The making of indicator of compromise using malware reverse engineering techniques. In *2020 International Conference on ICT for Smart Society (ICISS)*. IEEE, IEEE, Indonesia, 1–6.

[4] Dennis Andriesse, Xi Chen, Victor Van Der Veen, Asia Slowinska, and Herbert Bos. 2016. An In-Depth Analysis of Disassembly on Full-Scale x86/x64 Binaries.. In *USENIX Security Symposium*. USENIX, US, 583–600.

[5] Simone Aonzo, Yufei Han, Alessandro Mantovani, and Davide Balzarotti. 2023. Humans vs. Machines in Malware Classification.

[6] Erin Avllazagaj, Ziyun Zhu, Leyla Bilge, Davide Balzarotti, and Tudor Dumitras. 2021. When Malware Changed Its Mind: An Empirical Study of Variable Program Behaviors in the Real World.. In *USENIX Security Symposium*. USENIX, US, 3487–3504.

[7] Roberto Baldoni, Emilio Coppa, Daniele Cono D'Elia, and Camil Demetrescu. 2017. Assisting Malware Analysis with Symbolic Execution: A Case Study. In *Cyber Security Cryptography and Machine Learning*, Shlomi Dolev and Sachin Lodha (Eds.). Springer, US.

[8] Davide Balzarotti, Marco Cova, Christoph Karlberger, Engin Kirda, Christopher Kruegel, and Giovanni Vigna. 2010. Efficient Detection of Split Personalities in Malware.. In *NDSS*, Vol. 1. IFIP, US, 1.

[9] Parth Bhatt, Edgar Toshiro Yano, and Per Gustavsson. 2014. Towards a framework to detect multi-stage advanced persistent threats attacks. In *2014 IEEE 8th international symposium on service oriented system engineering*. IEEE, IEEE, UK, 390–395.

[10] Steve Bono. 2005. Thinking Like an Attacker. In *19th Large Installation System Administration Conference (LISA 05)*, Vol. 1. USENIX, US, 1.

[11] Marcus Botacin, Hojjat Aghakhani, Stefano Ortolani, Christopher Kruegel, Giovanni Vigna, Daniela Oliveira, Paulo Lício De Geus, and André Grégio. 2021. One Size Does Not Fit All: A Longitudinal Analysis of Brazilian Financial Malware. *ACM Trans. Priv. Secur.* 24, 2, Article 11 (jan 2021), 31 pages. https://doi.org/10.1145/3429741

[12] Marcus Botacin, Fabricio Ceschin, Ruimin Sun, Daniela Oliveira, and André Grégio. 2021. Challenges and pitfalls in malware research. *Computers & Security* 106 (2021), 102287. https://doi.org/10.1016/j.cose.2021.102287

[13] Marcus Botacin, Paulo Lício de Geus, and André Grégio. 2019. "VANILLA" malware: vanishing antiviruses by interleaving layers and layers of attacks. *Journal of Computer Virology and Hacking Techniques* 1 (2019), 1. https://doi.org/10.1007/s11416-019-00333-y

[14] Marcus Botacin, Lucas Galante, Paulo de Geus, and André Grégio. 2020. RevEngE is a Dish Served Cold: Debug-Oriented Malware Decompilation and Reassembly. In *Proceedings of the 3rd Reversing and Offensive-Oriented Trends Symposium* (Vienna, Austria) *(ROOTS'19)*. Association for Computing Machinery, New York, NY, USA, Article 1, 12 pages. https://doi.org/10.1145/3375894.3375895

[15] Marcus Botacin and André Grégio. 2021. Malware MultiVerse: From Automatic Logic Bomb Identification to Automatic Patching and Tracing. https://doi.org/10.48550/ARXIV.2109.06127

[16] Michael Brengel and Christian Rossow. 2021. YARIX: Scalable YARA-based Malware Intelligence. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, US, 3541–3558. https://www.usenix.org/conference/usenixsecurity21/presentation/brengel

[17] Sarah Brown, Joep Gommers, and Oscar Serrano. 2015. From Cyber Security Information Sharing to Threat Management. In *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security* (Denver, Colorado, USA) *(WISCS '15)*. Association for Computing Machinery, New York, NY, USA, 43–49. https://doi.org/10.1145/2808128.2808133

[18] David Brumley, Cody Hartwig, Zhenkai Liang, James Newsome, Dawn Song, and Heng Yin. 2008. Automatically identifying trigger-based behavior in malware. *Botnet Detection: Countering the Largest Security Threat* 1, 1 (2008), 65–88.

[19] Kevin Burk, Fabio Pagani, Christopher Kruegel, and Giovanni Vigna. 2022. Decomperson: How Humans Decompile and What We Can Learn From It. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 2765–2782. https://www.usenix.org/conference/usenixsecurity22/presentation/burk

[20] Marcus Carpenter and Chunbo Luo. 2023. Behavioural Reports of Multi-Stage Malware. arXiv:2301.12800 [cs.CR]

[21] Fabrício Ceschin, Heitor Murilo Gomes, Marcus Botacin, Albert Bifet, Bernhard Pfahringer, Luiz S. Oliveira, and André Grégio. 2020. Machine Learning (In) Security: A Stream of Problems. arXiv:2010.16045 [cs.CR]

[22] Binlin Cheng, Jiang Ming, Jianmin Fu, Guojun Peng, Ting Chen, Xiaosong Zhang, and Jean-Yves Marion. 2018. Towards Paving the Way for Large-Scale Windows Malware Analysis: Generic Binary Unpacking with Orders-of-Magnitude Performance Boost. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 395–411. https://doi.org/10.1145/3243734.3243771

[23] Steffen Enders, Eva-Maria C. Behner, Niklas Bergmann, Mariia Rybalka, Elmar Padilla, Er Xue Hui, Henry Low, and Nicholas Sim. 2022. dewolf: Improving Decompilation by leveraging User Surveys. https://doi.org/10.48550/ARXIV.2205.06719

[24] Mohammad Reza Farhadi, Benjamin CM Fung, Yin Bun Fung, Philippe Charland, Stere Preda, and Mourad Debbabi. 2015. Scalable code clone search for malware analysis. *Digital Investigation* 15 (2015), 46–60.

[25] Mariano Graziano, Davide Canali, Leyla Bilge, Andrea Lanzi, and Davide Balzarotti. 2015. Needles in a Haystack: Mining Information from Public Dynamic Analysis Sandboxes for Malware Intelligence. In *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, Washington, D.C., 1057–1072. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/graziano

[26] Panos Kampanakis. 2014. Security Automation and Threat Information-Sharing Options. *IEEE Security & Privacy* 12, 5 (2014), 42–51. https://doi.org/10.1109/MSP.2014.99

[27] Mohammad Sina Karvandi, MohammadHosein Gholamrezaei, Saleh Khalaj Monfared, Soroush Meghdadizanjani, Behrooz Abbassi, Ali Amini, Reza Mortazavi, Saeid Gorgin, Dara Rahmati, and Michael Schwarz. 2022. HyperDbg: Reinventing Hardware-Assisted Debugging. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, US, 1709–1723.

[28] Yuhei Kawakoya, Makoto Iwamura, and Mitsutaka Itoh. 2010. Memory behavior-based automatic malware unpacking in stealth debugging environment. In *2010 5th International Conference on Malicious and Unwanted Software*. IEEE, France, 39–46. https://doi.org/10.1109/MALWARE.2010.5665794

[29] Eujeanne Kim, Sung-Jun Park, Dong-Kyu Chae, Seokwoo Choi, and Sang-Wook Kim. 2020. A Human-in-the-Loop Approach to Malware Author Classification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 3289–3292. https://doi.org/10.1145/3340531.3417467

[30] Dhilung Kirat, Giovanni Vigna, and Christopher Kruegel. 2014. Barecloud: Bare-metal analysis-based evasive malware detection. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. USENIX, US, 287–301.

[31] Alexander Küchler, Alessandro Mantovani, Yufei Han, Leyla Bilge, and Davide Balzarotti. 2021. Does Every Second Count? Time-based Evolution of Malware Behavior in Sandboxes.. In *NDSS*. IFIP, US, 1.

[32] Giuseppe Laurenza, Leonardo Aniello, Riccardo Lazzeretti, and Roberto Baldoni. 2017. Malware Triage Based on Static Features and Public APT Reports. In *Cyber Security Cryptography and Machine Learning*, Shlomi Dolev and Sachin Lodha (Eds.). Springer International Publishing, Cham, 288–305.

[33] Giuseppe Laurenza, Riccardo Lazzeretti, and Luca Mazzotti. 2020. Malware Triage for Early Identification of Advanced Persistent Threat Activities. *Digital Threats* 1, 3, Article 16 (aug 2020), 17 pages. https://doi.org/10.1145/3386581

[34] Martina Lindorfer, Matthias Neumayr, Juan Caballero, and Christian Platzer. 2013. Poster: Cross-platform malware: write once, infect everywhere. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, Germany, 1425–1428.

[35] Yuying Liu, Pin Yang, Peng Jia, Ziheng He, and Hairu Luo. 2022. MalFuzz: Coverage-guided fuzzing on deep learning-based malware classification model. *Plos one* 17, 9 (2022), e0273804.

[36] Alessandro Mantovani, Simone Aonzo, Yanick Fratantonio, and Davide Balzarotti. 2022. RE-Mind: a First Look Inside the Mind of a Reverse Engineer. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA,

2727–2745. https://www.usenix.org/conference/usenixsecurity22/presentation/mantovani

[37] Lorenzo Martignoni, Mihai Christodorescu, and Somesh Jha. 2007. Omniunpack: Fast, generic, and safe unpacking of malware. In *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*. IEEE, IEEE, US, 431–441.

[38] James Mattei, Madeline McLaughlin, Samantha Katcher, and Daniel Votipka. 2022. A Qualitative Evaluation of Reverse Engineering Tool Usability. In *Proceedings of the 38th Annual Computer Security Applications Conference* (Austin, TX, USA) *(ACSAC '22)*. Association for Computing Machinery, New York, NY, USA, 619–631. https://doi.org/10.1145/3564625.3567993

[39] Vasileios Mavroeidis and Audun Jøsang. 2018. Data-Driven Threat Hunting Using Sysmon. In *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy* (Guiyang, China) *(ICCSP 2018)*. Association for Computing Machinery, New York, NY, USA, 82–88. https://doi.org/10.1145/3199478.3199490

[40] Brad Miller, Alex Kantchelian, Michael Carl Tschantz, Sadia Afroz, Rekha Bachwani, Riyaz Faizullabhoy, Ling Huang, Vaishaal Shankar, Tony Wu, George Yiu, Anthony D. Joseph, and J. D. Tygar. 2016. Reviewer Integration and Performance Measurement for Malware Detection. In *Proceedings of the 13th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment - Volume 9721* (San Sebastián, Spain) *(DIMVA 2016)*. Springer-Verlag, Berlin, Heidelberg, 122–141. https://doi.org/10.1007/978-3-319-40667-1_7

[41] Jiang Ming, Meng Pan, and Debin Gao. 2013. iBinHunt: Binary hunting with inter-procedural control flow. In *Information Security and Cryptology–ICISC 2012: 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers 15*. Springer, Springer, South Korea, 92–109.

[42] Najmeh Miramirkhani, Mahathi Priya Appini, Nick Nikiforakis, and Michalis Polychronakis. 2017. Spotless sandboxes: Evading malware analysis systems using wear-and-tear artifacts. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, IEEE, US, 1009–1024.

[43] Abhijit Mohanta and Anoop Saldanha. 2020. *Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware*. Springer, US.

[44] KA Monnappa. 2018. *Learning Malware Analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware*. Packt Publishing Ltd, US.

[45] Andreas Moser, Christopher Kruegel, and Engin Kirda. 2007. Exploring multiple execution paths for malware analysis. In *2007 IEEE Symposium on Security and Privacy (SP'07)*. IEEE, IEEE, US, 231–245.

[46] Nuthan Munaiah, Akond Rahman, Justin Pelletier, Laurie Williams, and Andrew Meneely. 2019. Characterizing Attacker Behavior in a Cybersecurity Penetration Testing Competition. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ACM/IEEE, Brazil, 1–6. https://doi.org/10.1109/ESEM.2019.8870147

[47] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad. 2015. Webwitness: Investigating, categorizing, and mitigating malware download paths. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*. USENIX, US, 1025–1040.

[48] Cory Q. Nguyen and James E. Goldman. 2010. Malware Analysis Reverse Engineering (MARE) Methodology & Malware Defense (M.D.) Timeline. In *2010 Information Security Curriculum Development Conference* (Kennesaw, Georgia) *(InfoSecCD '10)*. Association for Computing Machinery, New York, NY, USA, 8–14. https://doi.org/10.1145/1940941.1940944

[49] Anita Nikolich. 2019. Grey Science. In *Enigma 2019 (Enigma 2019)*. USENIX Association, Burlingame, CA, 1. https://www.usenix.org/node/226466

[50] Igor Novkovic and Stjepan Groš. 2016. Can malware analysts be assisted in their work using techniques from machine learning?. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. ACM/IEEE, Taiwan, 1408–1413. https://doi.org/10.1109/MIPRO.2016.7522360

[51] OPSWAT. 2022. State of Malware Analysis. https://info.opswat.com/hubfs/opswat-2022-state-of-malware-analysis.pdf.

[52] Roberto Paleari, Lorenzo Martignoni, Giampaolo Fresi Roglia, and Danilo Bruschi. 2010. N-version disassembly: differential testing of x86 disassemblers. In *Proceedings of the 19th international symposium on Software testing and analysis*. ACM, China, 265–274.

[53] Kyuhong Park, Burak Sahin, Yongheng Chen, Jisheng Zhao, Evan Downing, Hong Hu, and Wenke Lee. 2021. Identifying Behavior Dispatchers for Malware Analysis. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security* (Virtual Event, Hong Kong) *(ASIA CCS '21)*. Association for Computing Machinery, New York, NY, USA, 759–773. https://doi.org/10.1145/3433210.3457894

[54] Younghee Park and Douglas Reeves. 2011. Deriving common malware behavior through graph clustering. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, China, 497–502.

[55] Fei Peng, Zhui Deng, Xiangyu Zhang, Dongyan Xu, Zhiqiang Lin, and Zhendong Su. 2014. X-force: Force-executing binary programs for security applications. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. USENIX, US, 829–844.

[56] Daniel Plohmann, Sebastian Eschweiler, and Elmar Gerhards-Padilla. 2013. Patterns of a cooperative malware analysis workflow. In *2013 5th International Conference on Cyber Conflict (CYCON 2013)*. IEEE, Estonia, 1–18.

[57] Edward Raff, Richard Zak, Gary Lopez Munoz, William Fleming, Hyrum S. Anderson, Bobby Filar, Charles Nicholas, and James Holt. 2020. Automatic Yara Rule Generation Using Biclustering. In *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security* (Virtual Event, USA) *(AISec'20)*. Association for Computing Machinery, New York, NY, USA, 71–82. https://doi.org/10.1145/3411508.3421372

[58] Md Omar Faruk Rokon, Risul Islam, Ahmad Darki, Evangelos E Papalexakis, and Michalis Faloutsos. 2020. SourceFinder: Finding Malware Source-Code from Publicly Available Repositories in GitHub.. In *RAID*. Springer, 2020, 149–163.

[59] Hassen Saıdi, Phillip Porras, and Vinod Yegneswaran. 2010. Experiences in malware binary deobfuscation. *Virus Bulletin* 1, 1 (2010), 1.

[60] Stefano Sebastio, Eduard Baranov, Fabrizio Biondi, Olivier Decourbe, Thomas Given-Wilson, Axel Legay, Cassius Puodzius, and Jean Quilbeuf. 2020. Optimizing symbolic execution for malware behavior classification. *Computers & Security* 93 (2020), 101775. https://doi.org/10.1016/j.cose.2020.101775

[61] Shanhu Shang, Ning Zheng, Jian Xu, Ming Xu, and Haiping Zhang. 2010. Detecting malware variants via function-call graph similarity. In *2010 5th International Conference on Malicious and Unwanted Software*. IEEE, France, 113–120. https://doi.org/10.1109/MALWARE.2010.5665787

[62] Monirul I Sharif, Vinod Yegneswaran, Hassen Saidi, Phillip A Porras, and Wenke Lee. 2008. Eureka: A Framework for Enabling Static Malware Analysis.. In *ESORICS*, Vol. 8. Springer, Springer, Spain, 481–500.

[63] Narasimha Shashidhar and Peter Cooper. 2016. Teaching malware analysis: The design philosophy of a model curriculum. In *2016 4th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE, US, 119–125. https://doi.org/10.1109/ISDFS.2016.7473529

[64] Yan Shoshitaishvili, Ruoyu Wang, Christopher Salls, Nick Stephens, Mario Polino, Andrew Dutcher, John Grosen, Siji Feng, Christophe Hauser, Christopher Kruegel, and Giovanni Vigna. 2016. SOK: (State of) The Art of War: Offensive Techniques in Binary Analysis. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, US, 138–157. https://doi.org/10.1109/SP.2016.17

[65] Michael Sikorski and Andrew Honig. 2012. *Practical malware analysis: the hands-on guide to dissecting malicious software*. no starch press, US.

[66] TwoSixLabs. 2020. Edge of the Art in Vulnerability Research. https://apps.dtic.mil/sti/citations/AD1096948.

[67] Xabier Ugarte-Pedrero, Davide Balzarotti, Igor Santos, and Pablo G. Bringas. 2015. SoK: Deep Packer Inspection: A Longitudinal Study of the Complexity of Run-Time Packers. In *2015 IEEE Symposium on Security and Privacy*. IEEE, US, 659–673. https://doi.org/10.1109/SP.2015.46

[68] Xabier Ugarte-Pedrero, Mariano Graziano, and Davide Balzarotti. 2019. A Close Look at a Daily Dataset of Malware Samples. *ACM Trans. Priv. Secur.* 22, 1, Article 6 (jan 2019), 30 pages. https://doi.org/10.1145/3291061

[69] Marie Vasek and Tyler Moore. 2012. Do malware reports expedite cleanup? An experimental study. In *USENIX CSET*. USENIX Association, USENIX, US, 1.

[70] Daniel Votipka, Seth Rabin, Kristopher Micinski, Jeffrey S. Foster, and Michelle L. Mazurek. 2019. An Observational Investigation of Reverse Engineers' Process and Mental Models. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3313040

[71] Aaron Weathersby. 2021. Prevalence of PII within Public Malware Sandbox Samples and Implications for Privacy and Threat Intelligence Sharing: Student Paper Abstract. *J. Comput. Sci. Coll.* 37, 3 (oct 2021), 166.

[72] Khaled Yakdan, Sergej Dechand, Elmar Gerhards-Padilla, and Matthew Smith. 2016. Helping johnny to analyze malware: A usability-optimized decompiler and malware analysis user study. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, IEEE, US, 158–177.

[73] Miuyin Yong Wong, Matthew Landen, Manos Antonakakis, Douglas M. Blough, Elissa M. Redmiles, and Mustaque Ahamad. 2021. An Inside Look into the Practice of Malware Analysis. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, Republic of Korea) *(CCS '21)*. Association for Computing Machinery, New York, NY, USA, 3053–3069. https://doi.org/10.1145/3460120.3484759

[74] Katsunari Yoshioka, Yoshihiko Hosobuchi, Tatsunori Orii, and Tsutomu Matsumoto. 2010. Vulnerability in Public Malware Sandbox Analysis Systems. In *2010 10th IEEE/IPSJ International Symposium on Applications and the Internet*. IEEE, South Korea, 265–268. https://doi.org/10.1109/SAINT.2010.16

A LITERATURE REVIEW

To identify the knowledge gaps in malware practices in a representative way, we relied upon a Systematic Literature Review (SLR) strategy. More specifically, we refined a previous SLR on malware research [12] to cover the aspects specific to the analysts' practices. We chose this previous SLR because it follows a widely accepted SLR methodology (PRISMA); it was already peer-reviewed: and it covered 18 years of publications in top security conferences. We got access to the papers used in the previous SLR and refined the search criteria to include only the following keywords in the title, the abstract, or the text: *"malware practice"*, *"analysis practice"*, *"analyst's practice"*, *"practice of"*, *"reverse engineering practice"*, *"analysis process"*, and *"reverse engineering process"*. We then manually inspected each one of the matching papers to verify if the matches correspond to the description of an analysis practice. We understand as an analysis practice any action actively taken by malware analysts, either it being part of the decision-making or the hands-on reverse engineering process. We considered all analysis practices we identified in the literature.

Table 28 shows the distribution of the selected papers over the years and the venues, both for the Original SLR and the Refined one. We notice that whereas multiple solutions are proposed in the papers, analysis practices are only covered in small fractions. Among the conferences, the rate of papers covering analysis practices ranges from 6% to 21%, with a combined average of 11% of all papers. Over time, the rate of papers covering analysis practices in each ranged from 0% to 36%, which was achieved in 2005 (4 out of 11 malware papers).

In total, we found 55 papers mentioning some analysis practice. We discarded repeated topics and summarized the practices in the workflow presented in Appendix B. We reviewed the resulting workflow to identify existing Knowledge Gaps (KGs), i.e., analysts' practice problems claimed as open by the reviewed literature works. We conducted a second literature search in the works published after the original SLR to identify if more recent works answered some of the so-far open questions. The identified works are presented in Section 2 whenever appropriated. The remaining open KGs were translated to the 31 questions presented in Appendix C.

B MALWARE ANALYSIS WORKFLOW

The analysis practices we identified in the literature are not performed in isolation. In turn, they are part of an integrated set of reverse engineering practices; Some actions depend on previous ones or provide the basis for future developments. Therefore, we attempted to summarize the practices in a clear workflow, based on the placement of the practices in the literature works. We tried to respect the order of the practices as reported in the original papers, but we took the freedom to assume and reorder actions whenever the papers did not clearly state which practices they assumed. We acknowledge that malware analysts might play different roles and only perform parts of these actions, but our goal is to establish a general workflow that applies to most malware analysts. We summarized the tasks performed by malware analysts in the workflow shown in Figure 4. The steps are the following:

(1) One learns malware analysis techniques and is hired as a malware analyst.
(2) The employee assigns a task to the analyst.
(3) In most cases, but not all, it is a reverse engineering task.
(4) In reverse engineering tasks, the analyst receives (one or a set of) samples to analyze.

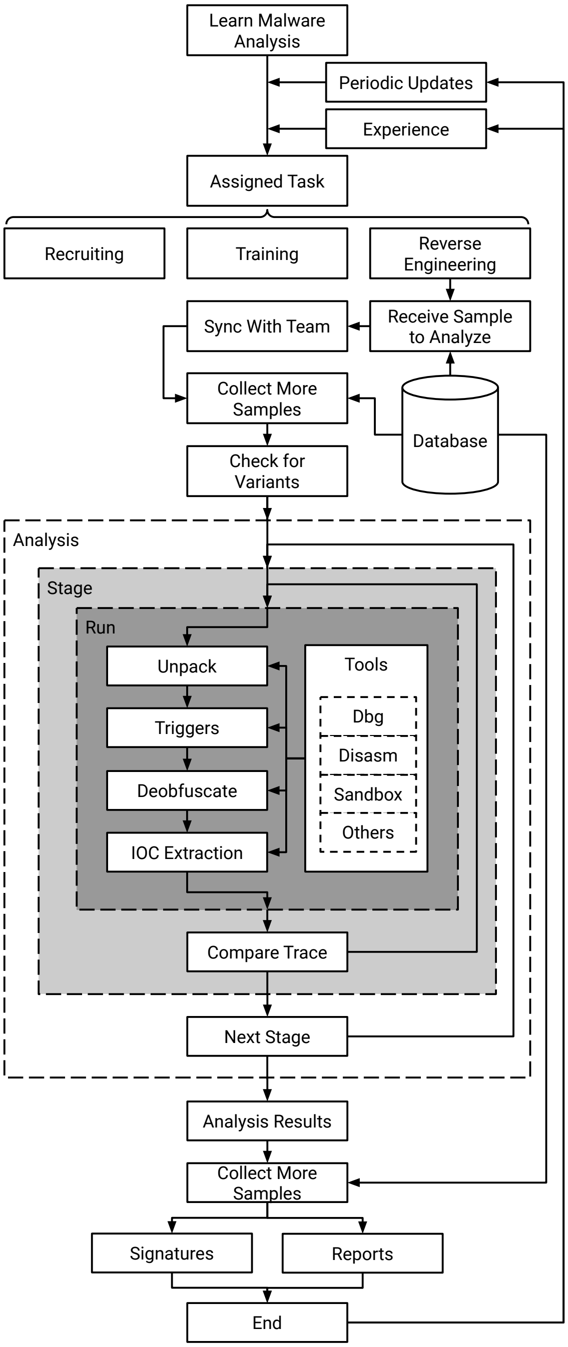

Figure 4: Malware Analysis Workflow.

(5) If working with a team, the analyst syncs with the team the samples to be analyzed.
(6) Upon starting to analyze, an analyst might collect additional (often similar) samples to enrich the analysis procedure.
(7) The analyst might recognize the sample is a variant of previously-analyzed samples. Specific actions might be taken in this case to speed up analyses.

Table 28: Paper Selection. Paper distribution per year (2000 – 2018) and per venue for the Original [12] and the Refined SLR.

Venue/Year	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		16		17		18		Total	
	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R
USENIX	1	0	0	0	0	0	0	0	0	0	1	0	1	0	6	2	2	0	3	1	7	1	8	1	10	1	12	0	9	2	7	0	9	3	13	1	6	0	95	12
CCS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	4	1	6	0	6	0	7	0	11	0	9	2	11	1	14	0	2	0	11	2	6	0	89	6
ACSAC	0	0	0	0	0	0	0	0	2	0	3	2	2	0	4	0	4	1	1	0	3	0	8	0	10	3	7	0	10	0	6	1	3	1	7	0	8	0	78	6
IEEE S&P	0	0	1	0	0	0	0	0	0	0	1	0	3	2	2	1	1	0	0	0	0	0	10	0	17	2	12	0	3	0	6	1	4	2	5	1	3	1	68	11
DIMVA	0	0	0	0	0	0	0	0	0	0	4	1	4	0	3	0	8	0	2	0	3	0	0	0	8	1	4	1	8	1	7	0	7	2	5	1	4	2	67	9
NDSS	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	3	0	3	1	3	1	3	0	2	0	4	0	5	0	4	1	9	1	7	0	3	1	49	5
RAID	0	0	0	0	1	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	3	0	5	1	5	1	3	0	4	1	3	0	3	0	31	3
ESORICS	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	2	1	1	0	0	0	0	0	2	0	3	0	3	0	0	0	1	0	1	1	0	0	14	3
Total	1	0	1	0	1	0	0	0	3	0	11	4	15	2	17	3	24	3	16	2	22	2	36	1	63	7	56	4	54	5	47	3	39	10	52	6	33	4	491	55

(8) The analyst starts reversing each sample.
  (a) The analyst starts the investigation by the first malware stage.
    (i) The analyst performs the first analysis run.
      (A) Unpack, Deobfuscate, and Find the triggers for each sample.
      (B) Use the appropriate tool for each task.
      (C) Extract the IOCs.
    (ii) Compare the results from multiple runs.
    (iii) If the results are not coherent, perform an additional run. Otherwise, stop.
  (b) The analyst verifies if the malware has a next stage to analyze. If so, repeat the process.
(9) Having analyzed all samples, the analyst is ready to report the outcomes.
(10) The analyst decides if the sample will only be reported or if a new signature is required.
(11) The analyst might collect more samples (e.g., variants) to amplify the impact of the analyses (e.g., more comprehensive signatures).
(12) The analyst writes the reports and the signatures.
(13) As the process is finished, the analyst and/or the company might store the analyzed sample for future queries.
(14) By the end of the process, the analyst is more experienced, and this experience is shared with other analysts to build more robust malware analysis teams.

C SURVEY QUESTIONS

The invited participants were asked to initially sign the consent form and then were presented to the following blocks of questions:

C.1 Professional Background (PB)

- PB1. *"Considering your daily job tasks. How often do you perform malware analysis tasks?"*
  - (A) I'm a full-time malware analyst.
  - (B) Most of my tasks, but not all, are malware analysis tasks.
  - (C) A reasonable number of my tasks are malware analysis tasks.
  - (D) I eventually do malware analysis tasks.
  - (E) I don't do malware analysis tasks.
- PB2. *"How do you characterize your current work/job?"*
  - (A) I'm part of a team and we analyze samples together.
  - (B) I'm part of a team, but we analyze samples independently.
  - (C) I'm an independent researcher.
- PB3. *"How much context about the malware collection do you often have about the analysis?"*
  - (A) I'm part of a team that analyzes regionalized/focused threats, thus I know the context of the infection.
  - (B) I'm part of a local analysis team that analyzes threats to our own company thus I know how malware was collected.
  - (C) I work with samples collected from a 3rd-party, thus I never know where they came from.

C.2 Professional Skills (PS)

- PS1. *"How did you learn to analyze malware?"*
  - (A) Post-grad in the field.
  - (B) Bsc in the field.
  - (C) Certification.
  - (D) Experience working (B.Sc. in another field).
  - (E) Self-taught.
- PS2. *"How long have you been working as a malware analyst?"*
  - Open answer.
- PS3. *"How do you get updated about new malware analysis developments? Rate accordingly:"*
  - (A) Academic Papers [0-100]
  - (B) White Papers/Blog posts [0-100]
  - (C) Youtube Videos [0-100]
  - (D) Security events [0-100]
  - (E) Training [0-100]

C.3 Analysis Practices (AP)

- AP1. *"Do you collect more samples than the one you are analyzing to perform the analysis tasks?"*
  - (A) I often collect more samples, to write better/broader signatures.
  - (B) I often collect more samples, to measure the impact on reports.
  - (C) I sometimes collect more samples, to understand how a new technique works.
  - (D) No, I only analyze the requested samples.
- AP2. *"Do you store the analyzed samples for further queries?"*
  - (A) They are always stored by the company I work for.
  - (B) They are always stored by me.
  - (C) They are eventually stored by me, just as a curiosity.
  - (D) They are never stored.
- AP3. *"Where and Who hosts analysis procedures?"*
  - (A) Mostly on my own computer.
  - (B) Mostly on public web services.
  - (C) Mostly on company-provided sandboxes.

- AP4. *"What's your relation with public malware analysis sandboxes?"*
  - (A) My employer allows using it, but I don't like using it.
  - (B) My employer allows using it, and I like using it.
  - (C) My employer doesn't allow me to use it.
- AP5. *"How often do you see malware variants to the point of recognizing that you analyzed a similar construction before?"*
  - (A) Rarely, the company triage system does a good job.
  - (B) Sometimes, but it is more common to see different families of malware.
  - (C) Very often, there are too many variants out there to be analyzed.
- AP6. *"How much of your work involves manual tasks? (e.g., debugging vs tracing)"*
  - (A) Most of my work is manual.
  - (B) Around 50% manual and 50% automated.
  - (C) I run fully automated analysis pipelines.
- AP7. *"How do you handle multi-stage/multi-format malware?"*
  - (A) With multiple tools, but mostly via automatedly integration.
  - (B) With multiple tools, but treating each stage as a completely new analysis.
  - (C) With multiple tools, but manually copy-pasting data from one tool to another.
  - (D) With a single tool that handles all formats at once.
  - (E) I only analyze a single infection vector.
- AP8. *"Considering your skills, how much do you struggle doing the following tasks?"*
  - (A) Unpacking malware [0-100]
  - (B) Identifying triggers [0-100]
  - (C) Deobfuscating malware [0-100]
- AP9. *"Considering your available time, how much do you struggle doing the following tasks?"*
  - (A) Unpacking malware [0-100]
  - (B) Identifying triggers [0-100]
  - (C) Deobfuscating malware [0-100]

C.4 Analysis Accuracy (AA)

- AA1. *"How many runs of a sample do you typically do on sandboxes and debuggers? It doesn't count restarting the debugger to set breakpoints or so. It means complete analysis sessions."*
  - (A) Always one.
  - (B) Typically one, but sometimes more.
  - (C) Typically a couple of runs.
  - (D) Always multiple runs.
- AA2. *"How many different dynamic analysis systems (sandbox, debuggers, so on) do you typically run a sample?"*
  - (A) Always one.
  - (B) Typically one, but sometimes more.
  - (C) Typically a couple of runs.
  - (D) Always multiple runs.
- AA3. *"When you change the analysis environment, do you change the sandbox execution environment across the runs?"*
  - (A) Always test multiple Architecture (32/64) and OS versions (e.g., Windows).
  - (B) Always test multiple architectures, but no OS versions.
  - (C) Always test OS versions, but no architectures.
  - (D) Neither OS nor architecture (single environment).
- AA4. *"Which techniques do you use for discovering new malware paths? Rate the amount of use."*
  - (A) Fuzzing [0-100]
  - (B) Symbolic Execution [0-100]
  - (C) Concolic Analysis [0-100]
  - (D) Forced Execution [0-100]
  - (E) Manual Inspection [0-100]
- AA5. *"When you perform multiple malware runs, how do you compare the different traces?"*
  - (A) Based on the IoCs.
  - (B) Graph-based comparison.
  - (C) Multiple versions are considered regardless of the differences.

C.5 Analysis Procedures Performance (APP)

- APP1. *"How do you rate the overall performance of the tools that you use?"*
  - (A) They are slow, but it's intrinsic to the malware nature.
  - (B) They are slow, and they could be improved.
  - (C) They are fast enough.
- APP2. *"Specifically about sandboxes, how do you think making tracing faster would help your work?"*
  - (A) Very helpful.
  - (B) Makes a difference only in specific cases.
  - (C) Makes no difference.

C.6 Analysis Outcomes Performance (AOP)

- AOP1. *"What are the typical outcomes of your analysis tasks?"*
  - (A) Threat report and signatures with the same frequency.
  - (B) Threat reports only.
  - (C) Signatures only.
- AOP2. *"How much do you worry about signature matching performance (matching time) when writing a signature?"*
  - (A) Performance is as important as accuracy.
  - (B) Performance is important, but accuracy first.
  - (C) Only accuracy is a requirement.

C.7 Tools Evaluation (TE)

- TE1. *"How often do you use these tools to analyze a malware sample? (0 means no sample, 100 for all samples)"*
  - (A) Similarity hashing [0-100]
  - (B) Debugger [0-100]
  - (C) Sandbox [0-100]
  - (D) Decompiler [0-100]
  - (E) Unpacker [0-100]
  - (F) Antivirus [0-100]
  - (G) Disassembler [0-100]
- TE2. *"How do you rate the current state of debuggers for malware analysis?"*
  - (A) They are essential but they require me to perform repetitive tasks.

  - (B) They are essential and I can perform all tasks with no problem.
  - (C) They are not essential to my work.
- TE3. *"How do you see the role of plugins in the debugging process?"*
  - (A) They are essential for malware analysis.
  - (B) They help, but they are not essential.
  - (C) They make no difference.
- TE4. *"How do you rate the current state of decompilers for malware analysis?"*
  - (A) They are very useful.
  - (B) They help in a minor part of the cases.
  - (C) They are far from being useful.
- TE5. *"What are the tools you would like to exist to help your malware analysis tasks?"*
  - Free-text answer.

C.8 Future insights (FI)

- FI1. *"How do you see the role of AI in the future of malware analysis?"*
  - (A) AI will solve the problem and eliminate analysts.
  - (B) AI will help in some tasks, but analysts will still be required in most cases.
  - (C) AI will not play a key role.
- FI2. *"What are your general thoughts about the future of malware analysis?"*
  - Free-text answer.

C.9 Voluntary Disclosure (VD)

- VD1. *"This survey is anonymous. However, if you disclose some information about yourself, it helps us to draw stronger conclusions about the findings based on your declared position, expertise, and so on. We can also reach out with follow-up questions. Now that you have filled out the survey, if you feel comfortable, you can voluntarily deanonymize yourself. Feel free to stay anonymous if you want."*
  - Free-text answer.

D FOLLOWUP SURVEY

The follow-up questions were designed individually to reach malware analysts who voluntarily disclosed their identities. We highlight that the follow-up questions were used to **explain** their responses, and not to **measure** prevalence, such that they do not impact the statistical results. All developed questions adopt the following template:

D.1 Follow-Up Questions (FU)

- FU-N. *"You mention to use the tool <NAME> to analyze malware. How do you use this tool?"*
  - Free-text answer.
- FU-N+1. *"You mention to never use the strategy <NAME> to analyze malware. Why not?"*
  - Free-text answer.

E ANALYST'S DESIRED TOOLS

We below present the analysts' answers about desired tools classified by how much new knowledge they require to be implemented. We identify the participants by a number (P_ID), according to Table 1 from Section 4.1, to highlight that their requests and needs are diversified.

Engineering Developments: Solutions that can be implemented using current knowledge.

Analysts want more scalability:

> P13. *"A Windows VM provided by Microsoft without many security things and tailored to allow me to change any characteristics of the machine without much trouble, like language, username, etc."*

Analysts want better Usability:

> P18. *"Better GUI based API tracer (similar like outdated API monitor)"*
>
> P8. *"I wish x64dbg could be called from the CLI and run a script with a sample."*

Analysts want more efficiency:

> P8. *"In Linux, I'd like to have more injection capabilities in strace and a Yara-like tool to match instructions."*

Analysts want to increase accuracy:

> P6. *"Better Unpackers."*

Scientific Developments: Solutions that require additional scientific developments, even though deployed via a product.

Analysts want increased accuracy:

> P5. *"Multi-Architecture Sandbox."*

`Challenge:` Whereas the construction of a sandbox is a well-studied topic, the correlation of data between multiple architectures is still an open problem.

Analysts want more usability:

> P7. *"A more automated angr."* (Ref. [64])
>
> P8. *"A good API logger that doesn't require me to choose which function calls I want to see. Something like strace but for Windows."*
>
> P14. *"A memory monitoring tool that you attach to a process before executing it and it automatically dumps anything interesting when allocated in memory (like PE files)."*
>
> P20. *"AI-assisted function identification for stripped binaries that actually works."*

`Challenge:` Whereas analysis and tracing are well-studied topics, the existing tools still require analysts to manually configure lots of key aspects of an analysis procedure (e.g., which functions to hook, how to explore multiple paths, where to stop analysis, where to put breakpoints, and so on). The analyst's request for automation is not only a matter of adding a new feature to the tools, but a request for the development of new reasoning tools, since to automatically hook a function, the analysis tool must be able to "guess" if that function is important in that given context. In this sense, the analyst's suggestion for AI assistance is in line with the problem challenge, since it requires additional reasoning.

The case of decompilers: They were reported as one of the most popular solutions among analysts. They were also the most commented solution, as they still have development gaps, as follows:

> P9. *"better decompilers to languages like delphi, go, rust."*

> P14. *"An easy-to-use decompiler based on the execution trace (for virtualized samples)"*

> P19. *"Improved decompilers with better types and static library detection; better ways to identify malware families."*

> P10. *"IA behavior analysis based on intermediate machine code."*

`Challenges:` The advancements requested by the analysts require additional scientific advancements, since most of these capabilities involve additional reasoning by the tools, in addition to better engineering support. The automatic identification of libraries or the best parts of the code to be decompiled requires the decompiler not only to know how to handle the code but also to interpret the goal and importance of the code pieces.

F ANALYST'S INSIGHTS ABOUT THE FUTURE

We below present the analysts' answers about their perceptions about the future of the malware analysis field. We identify the participants by a number (P_ID), according to Table 1 from Section 4.1, to highlight that their requests and needs are diversified.

Malware Tactics: It is increasingly important to identify **how** a malware works, not only if it is malicious or not.

> P5. *"It will require more and more skilled people. Malware evasion are common place now."*

> P7. *"Multi-stage, fileless, firmware and other types of samples that are difficult to analyze with traditional techniques will have a great impact on users' security, but at the same time will provide new opportunities for research in the field."*

> P5. *"With the increase of ARM devices, I believe we will have an increase of multi-architecture malware (recently I have seen an increase of multi-platform malware, but multi-architecture is still rare)."*

Developing Intelligence: Malware attacks appear in variants and can be stealth. It is increasingly important to develop knowledge about the attackers to map operations and anticipate movements.

> P15. *"Focus will change from file/code analysis on initial attack vectors (phishing, social engineering, network behavior etc.)"*

> P8. *"Being able to fully analyze a malware sample/family is not the most important thing IMHO. We have to have context and we need to extract intelligence from it, not only describe its features. Maybe we have to interact with its C2, track the actors, etc. So, malware analysis plays a key part on campaign/incident investigation, but it doesn't help much alone."*

The role of AI: The most trending technique at the moment was commented by many analysts. Their expressed views were:

> P12. *"I think the presence of a malware analyst will always be necessary. Perhaps there will be a day when an AI will be able to analyze with precision, but even in this case there will have to be a malware analyst to "feed" the AI with more inputs and progress the techniques and tools."*

> P13. *"AI will help and eliminate trivial tasks, but often is necessary to perform advanced tweaks to make the malware work, So, this needs to be done by a human being."*

> P14. *"AI will be useful for anomaly detection, but manual malware analysis will still be required to better understand how the attack works."*

> P18. *"AI will help in future more but there will be always a need for analysts."*

Education: More than any tool, current analysts express concern about training the next-gen analysts, a goal academia might supply.

> P19. *"We need better education, but it is a niche job."*

> P1. *"An ever-growing field with a great need for great and open-minded researchers. Start to think like attackers and combine it with the mindset of a defender and you'll more chance to win."*

> P9. *"Always will raise new challenger malware that will need skilled professionals and better courses will be a differential to prepare new professionals."*

G REPLICATION AND GENERALIZATION

Reaching a significant number of participants is the biggest challenge and threat to the validity of any survey study. We attempted to overcome this challenge and mitigate this risk in our main experiment by increasing the confidence in the representativeness of the analysts that we interviewed. To that, we limited the survey distribution to only analysts acknowledged to work in the field. We here take a further step. We reproduce the experiments with a different group of participants to evaluate how the answers generalize.

Participants. For the new survey round, we opted to distribute the survey via the Internet, as performed by related works. We distributed our survey via posts on social media and web forums to attract voluntary participants. The drawback of this approach is that we (and previous works) cannot guarantee that the responders are actually malware analysts and have the reported skills. The survey was open between October and December/23. In total, 20 analysts completed the survey. With this number of answers, we can compare the new results with the reference one on the same basis (we previously surveyed 21 analysts).

Population Differences. Whereas in the main experiment we were able to trace back analysts to their occupations, this is not possible in the reproduced survey due to the limitations of the Internet-based, open invitation strategy. The most significant difference noticed for the user population is that the new responders are younger. The average expertise years is now 4.33 vs. previously 7 years. The most experienced professional holds 15 years of experience in both cases. However, 3 professionals reported having this seniority in the first run vs. 1 professional in the replication study.

Methodology and Findings. We performed the same data analyses as described in the main paper. Due to space constraints, we opted to not reproduce all tables here, but only to compare the summary of each analysis.

Malware analysis is still one of the multiple tasks of most security professionals. The minority of analysts reported to be full-time malware analysts (prev. 14% vs. 10%). Most analysts position themselves on an intermediate number of analysis tasks. The fraction of eventual malware analysts is also constant (prev. 42% vs 40%).

Most analysts are individuals part of a team but analyze samples individually (prev. 76% vs. 80%). The only difference is a swap between those who were independent analysts (19% vs 10%) vs. teams that analyze samples together (prev. 4% vs. 10%).

Analysts know the context of the infections. Whereas previously most analysts reported analyzing regional threats (52% vs. 30%), now they report to analyze samples for their own local companies (14% vs. 50%). In common, the analysis of samples without infection context information is the less common scenario (7% vs. 20%).

Most analysts are self-taught (prev. 42% vs. 60%). The only noted difference is a swap between those who previously learned via postgrad in the field and those who learned via a certification process (10%). No analyst learned via formal undergrad courses.

Analysts remain updated mainly via whitepapers (prev. 46% vs. 34%). Academic papers are still a minority part (prev. 14% vs. 10%). The training category remained stable (prev. 12% vs. 13%). The newly interviewed analysts rely more on YouTube videos and events to remain updated (prev. 11% and 21% vs. 25% for each now).

Collected samples are important for signature and report generation. The fraction of analysts using previous samples to help generate signatures (prev. 38% vs. 30%) or writing reports (prev. 9% vs. 10%) is relatively constant. Now, fewer analysts collect more samples to understand their internal working (prev. 33% vs. 20%).

Samples storage depends on the company policy. The rate of analysts that never store samples is relatively constant (prev. 9% vs. 10%). When they do, part are due to company obligation (42% vs. 20%), and part due to curiosity (23% vs. 50%).

Most analysts run procedures on their own machines (prev. 85% vs. 70%) vs. company-provided sandboxes (prev. 9% vs. 30%).

The use of public sandboxes remains a controversial point. In this new study, the number of companies disallowing their use has grown (prev. 19% vs. 40%). Among the 60% allowed to use, 30% like and 30% dislike them (prev. 28%).

Analysts still see malware variants. The number of malware analysts rarely observing variants has grown in the replication study (prev. 5% vs. 30%), but most analysts still report seeing variants sometimes (prev. 42% vs. 60%) or very often (52% vs. 10%).

Most analysis tasks are still manual. Most analysts report that their analysis procedures are half manual (prev. 52% vs. 60%) or mostly manual (prev. 48% vs. 40%).

The analysis of multi-stage malware requires constant analysts intervention. Most analysts treat each malware stage as a new analysis (prev. 66% vs. 60%) and copy and paste analysis results from tool to tool (prev. 19% vs. 30%).

Knowing to deobfuscate samples is a key malware analyst skill. adn the task analysts most struggle with (prev. 45% vs. 54%).

Deobufscating malware samples takes the analyst's time and it is a time bottleneck for most analysts (prev. 56% vs. 59%).

There is no standard for the number of sandbox runs. Part of the analysts (prev. 38% vs. 50%) reported to typically run samples only once, but sometimes more. Another 50% of the analysts reported to typically run more than once, but not always (prev. 38%).

The discovery of new execution paths is still widely manual (prev. 73% vs. 57%). Forced execution is the second most common technique (prev. 49% vs. 30%).

Multi-path malware is often characterized via their IoCs (prev. 62% vs. 80%). The remaining analysts reported considering any malware trace that presents malicious behavior as representative (prev. 28% vs. 20%). No analyst reported a different strategy.

Tool's performance is another controversial point. Half of the participants (prev. 38% vs. 50%) believe that the tools are fast enough. Another half believe the tools are slow, but some (prev. 24% vs. 30%) believe it is an intrinsic limitation whereas others (20%) believe that the tools can be improved.

Sandbox performance is also a controversial topic. Most analysts (prev. 100% vs. 80%) agree that faster sandboxes would help. But, the practical divide between those who believe it helps in specific (prev. 52% vs. 60%) vs. in broader (prev. 48% vs. 40%) cases remains.

Signatures and Reports are the most frequent analysis outcomes. Most analysts report to produce both signatures and reports (prev. 47% vs. 50%). Another set produces only reports (prev. 42% vs. 30%). The production of signatures only is limited (prev. 10% vs. 20%).

Accuracy is more important than performance for signature generation. The minority of the analysts weights the same the accuracy and the performance of the written signatures (prev. 33% vs. 10%). Most analysts put accuracy first (prev. 47% vs. 70%), and a minority only worry about accuracy (prev. 19% vs. 20%).

Decompilers and AVs at extreme positions. Decompilers remain at the top of the most used solutions (prev. 61% vs. 65%), whereas AVs remain the less used ones (prev. 58% vs. 29%).

Debuggers require performing repetitive tasks. Most analysts consider debuggers key for malware analysis (prev. 90% vs. 90%). However, most analysts report that debuggers require them to perform repetitive tasks (prev. 71% vs. 80%).

Plugins help improve debuggers for malware analysis. Most analysts report plugins to improve debugger operation (prev. 100% vs. 90%). Once again, there is a divide between those who consider plugins essential (prev. 42% vs. 60%) or not (prev. 57% vs. 30%).

Decompilers are solutions with high cost-benefit. This result aligns with previous findings. Most analysts consider decompilers as very useful tools (prev. 81% vs. 80%).

Analysts believe AI will help and not replace them. This finding is aligned with the previous results. Most analysts believe that AI will affect their job (prev. 96% vs. 90%). Most of them believe that AI will help (prev. 90% vs. 90%), whereas a few to none believe it will completely solve current analysts' problems (prev. 4% vs. 0%).

Analysts want AI to automate tools. Analysts once again reported desire for (i) new engineering tools; and (ii) new conceptual tools. Among the conceptual tools, all analysts expressed a desire for AI to be used to automate repetitive tasks.