"*I couldn't find it your honour, it mustn't be there*!" - tool errors, tool limitations and user error in digital forensics

Abstract

The field of digital forensics maintains significant reliance on the software it uses to acquire and investigate forms of digital evidence. Without these tools, analysis of digital devices would often not be possible. Despite such levels of reliance, techniques for validating digital forensic software are sparse and research is limited in both volume and depth. As practitioners pursue the goal of producing robust evidence, they face the onerous task of both ensuring the accuracy of their tools and, their effective use. Whilst tool errors provide one issue, establishing a tool's limitations also provides an investigatory challenge leading the potential for practitioner user-error and ultimately a grey area of accountability. This article debates the problems surrounding digital forensic tool usage, evidential reliability and validation.

Keywords: Digital Forensics; Crime; Tool Testing, Validation; Software; Errors

1 Introduction

Moore (2013) states that 'despite the maturity of the software industry, empirical research demonstrates that average software quality, when measured through the presence of software defects, is low'. Bugs are arguably almost certainly present in most software (Pan, 1999; Cusack and Homewood, 2013), only variances in their severity and the purpose for which the software was designed, determine whether they are tolerable. Arguably society has now reached a point where the acceptance of errors in software is becoming the norm. Such statements cause unease with the increasing reliance on the development and use of software for the processing of complex information within criminal and forensic realms. The need for accuracy does not sit well alongside knowledge of the fact that any software used to generate evidential data is unlikely to have been confirmed as 100% reliable by either a vendor or the practitioner using it. In legal disputes where the presentation of fact is required, flaws may exist in the tools used to interpret evidential data unbeknownst to all parties involved. Recent high-profile examples of questioned forensic software include the case of Mayer Herskovic, where it was stated that the accuracy of software which was used to examine the likelihood that a suspect's genetic material was present within a sample 'should be seriously questioned' (Kirchner, 2017).

The field of digital forensics (DF) provides no exception, with issues surrounding software functionality having been noted. As stated by SWGDE (2017) 'is a complex field that is heavily reliant on algorithms that are embedded in automated tools and used to process evidence where weaknesses or errors in these algorithms, tools, and processes can potentially lead to incorrect findings'. Cacheback and its scrutiny in the case of Casey Anthony surrounding the Mork file format provides arguably the most prominent example evidence misinterpretation (Alvarez, 2011; Wilson, 2011). Here, the number of online searches for the term 'chloroform' recovered from Mozilla Firefox's Mork history database was reported differently by two seperate tools, NetAnalysis (1 time) and Cacheback (84 times). It was subsequently determined that NetAnalysis's reported results were correct, despite Cacheback initially being relied upon. Whilst an example of an initial tool error (with incorrectly reported data), it also highlights the need for practitioners to have validated the accuracy of the tools they use both before and during investigations. In addition, the

importance of ensuring that findings from other relevant sources are taken into account is demonstrated, along with utilising techniques such as dual-tool validation for any obtained results. In most cases, DF finds itself in a scenario where often it is reliant on software in order to carry out its investigatory work (SWGDE, 2017) where digital data requires digital interpretation and display; there are often no viable manual analysis alternatives.

1.1 Digital Forensic Software

It is easy to approach this topic with the stance that wide-scale 'fundamental' (emphasis) flaws exist in DF software and practices. This is unlikely the case, however errors of varying severity can impact an investigation. The effect and consequences of unreliable tool use can be far reaching, both for those on the receiving end of inaccurate evidence and the practitioner involved (Brunty, 2011). As a result, where DF software is utilised, in terms of it's assumed reliability, practitioners 'can't simply accept the manufacturer's word for it; assumptions aren't permitted' (Sammons, 2012, p.33). This remains a sentiment almost certainly easier said than practically done. In order to establish whether a piece of DF software functions correctly, it must be validated, where Guo *et al*'s., (2009, p.13) definition states 'validation is the confirmation by examination and the provision of objective evidence that a tool, technique or procedure functions correctly and as intended'. In essence, a level of 'functional correctness' needs to be determined and assessed before a tool should be utilised in a live investigation (SWGDE, 2014; SWGDE, 2017; Becker et al., 2017), yet in practice it can be difficult to do this. The Forensic Science Regulator (2015) provides direction in the form of guidelines designed to support the the development of methods for validating forensic software with acknowledgement of the need to embed validation into laboratory practices in order to adhere to the ISO 17025 standard. Given that now in the United Kingdom, DF laboratories are required to obtain ISO 17025 by the Forensic Science Regulator, the development and effective implementation of adequate testing and validation methods is important. ISO 17025 seeks to achieve the development of robust practices and benchmark standards, ultimately improving practices within the field. Developed practices and procedures must be validated and maintained, where procedures for sustained validation must be in place.

Testing is a critical phase of any DF tool usage, but its undertaking is burdensome and error rates are rarely established (James, 2013). There is limited DF tool-testing research available (see work by Talib (2016) and Flandrin et al's., (2014) indication of the limited models in existence). The National Institute of Science and Technology's (2017) Computer Forensics Tool Testing (CFTT) Project provides possibly the only current public facing attempt at a comprehensive funded-level DF tool validation program since its introduction in the year 2000 (Guttman et al., 2011). Yet even the CFTT is limited, confined to only a few core DF tool functions (imaging, string searching and carving) and platforms. It should be noted, that private testing programs may be in operation, where results may be confined to specific entities such as law enforcement, however it is not possible to confirm and quantify the existence of such initiatives. DF tool vendors are arguably in a difficult position as practitioners maintain the highest expectations of investigative software (arguably rightly so given required evidentiary standards) and providers are expected to produce software capable of tackling numerous digital file formats and data structures. Varying data compositions and continuous target application iterations provide a challenge to vendors, and often their product is seen as the gateway to evidence acquisition and interpretation. Given the challenge of providing timely support to the practitioner, it seems unfair to expect

perfection from tool vendors, and in some sense, errors themselves are not the sole issue as they can often be mitigated or rectified; the trouble is detecting their existence.

At present, there arguably remains ambiguity around the assumed level of functionality provided by DF tools, and where unexpected results are returned from a DF tools operation this can not always be assumed to be as the result of a 'tool error'. In DF, a grey area exists, termed a 'tool limitation', essentially defining the limits of a tool's capability. Whilst this may seem commonly applicable to all software, the issue in DF is the potential vagueness in which a limitation may be defined, coupled with gaps in practitioner knowledge and experience to detect this limitation. In addition, the misuse of a tool through a failure to detect a limitation may be categorized as a user error. This article examines the difficulties surrounding DF tool usage, and the difference between tool errors, tool limitations and subsequent user errors.

2 The difference

It is necessary to first consider the difference between a DF tool error (TE), tool limitation (TL) and tool user error (UE). Whilst arguably it may seem theoretically obvious, distinguishing the three in practice provides a challenge. To provide clarity at this stage, the following definitions are offered before analysis takes place.

> Tool Error: "*A TE occurs when an application/software package misinterprets or misrepresents data which is the subject of its investigation*".
>
> Lyle (2010, p135) indicates that "typically there are two types of error, type I, also called a false positive (detecting it when it is not really there), and type II, also called a false negative (missing it when it really is there)". TEs arise when an application/software package (commonly referred to as a 'tool') claims to provide a certain functionality and following its operation within these confines, erroneous results are provided. To further define the scope of a TE, misinterpretation and misrepresentation provide two overarching categories. Misrepresentation involves the reporting of an inaccurate 'state' of data. This includes the failure to find information which is reported to be within the confines of its ability, or, the misreporting of information which is not present within a case (the latter being arguably a rarer situation). A misrepresentation involves the inaccurate parsing, conversion, translation or display of information classed as within the confines of the tools ability to process.
>
> It is key to note here that to be a TE, establishing both the confines of a tool's ability, and that the task carried out should have been within it, is imperative in order for any inaccuracies to be categorized as a TE. A tool cannot be expected to, and be held accountable for performing a task for which it never stated that it could. If the task is beyond the stated confines of the tool, use of the tool and subsequent reliance on inaccurate results must be categorized as a user error (UE).
>
> Tool Limitation: "*A TL defines the confines which an application/software package can be expected to reliably operate*".

A TL defines the boundaries of a tool's functionality. Inaccuracies generated from processes ran from within the confines of a tool's functionality are TEs (See Figure 1). A TL forms a line of accountability in the sand, where errors either side lead to differing forms of potential liability (see further discussion in Section 2.2).

Tool User Error: "*A UE defines the use of a process, procedure or tool for a purpose or in a way which it was not designed to be utilised*"

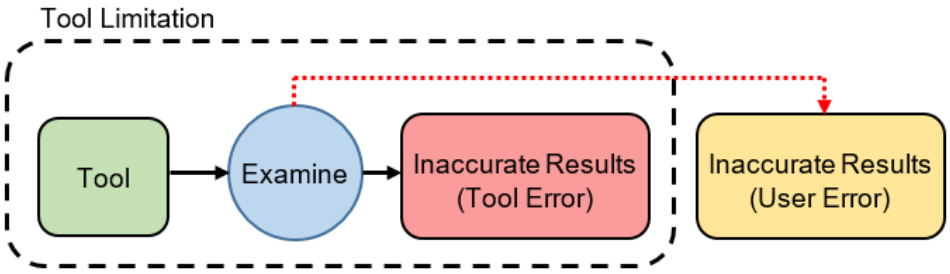

Figure 1: A visual representation of TEs and TLs

The incorrect use of a tool (a UE) can and may often lead to inaccuracies in investigation results (Aziz, 2014). UEs can appear avoidable at first glance, where individuals may point towards effective training, correctly implemented best practices, peer-review of work and the knowledge and competence of a practitioner, but this may not always be possible. UEs can occur due to the ineffective and ambiguous defining of a TL, leading to an genuine misinterpretation of results in belief that a tool is functioning in a way which in reality it is not. Whilst on face value the difference between TEs, UEs and TLs is obvious, the task of identifying and distinguishing them is not, and it is this factor which gives some cause for concern.

2.1 TEs and their avoidance

Software errors have been categorized by Chessman (2017) to include accidental errors, update errors, 'software rot' (performance degrade over time based on factors such as limited maintenance, support and code development), inadvertent and intentional bias and flawed self-test diagnostics. Avoiding TEs is difficult in the DF discipline as even detecting one has occured may not be possible without significant scrutiny and rigorous testing taking place. Existing quality control methods such as peer-review are unlikely to highlight TEs (James, 2013). Whilst this article is not suggesting that those in the field are not capable of doing such a task, or is it advocating that such processes should be done in every case, it is trying to debate what is realistic given existing industry pressures. DF evidence does not permit manual validation in the sense it cannot be touched, viewed and analysed without first undergoing tool-based interpretation and presentation. Therefore the discipline finds itself in the unusual situation of having to verify and validate the tools it uses, using these tools (see Figure 2). At which point, it becomes trapped in an infinite loop of potentially not being able to identify an issue. An accepted level of trust has to be drawn at some point, and often DF practitioners settle on the use of certain tools as an industry recognised standard to collect and present potentially evidential data. In addition, the DF field appears to be operating on the assumption that a tool which has a lot of users, may (by its sustained usage) have a low error rate (James, 2013). Whilst the Association of Chief Police Officers (2009) guidelines

promote the use of dual tool verification of results, this may not always be feasible in terms of resourcing and time. Dual tooling casework takes time, degrading existing computing power, potentially slowing down case processing. Crucially, dual tool verification does not guarantee reliability, it only improves the chances of it (Sammons, 2012; Cusack and Homewood, 2013).

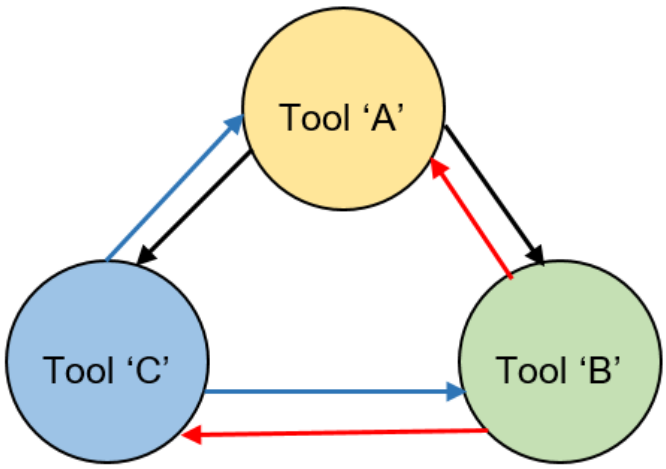

Figure 2: The circular testing of DF tools

There are a range of tools which at a high-level perform the same task, the parsing and displaying of a structured file system captured via a forensic data extraction (forensic imaging) process (consider OpenText's (2017) Encase, X-Ways's (2017) forensic platform, AccessData's (2017) Forensic Toolkit as well as open source platforms such as 'The Sleuth Kit'). Take for example a hypothetical set of tools which may perform identical acquisitions of a target digital device, inferring reliability in any results given. In reality the field has limited valid methods for manually verifying their performance. While each tool may produce an identical output (verified using hashes etc.), all may still be incorrect if they have utilised similar flawed parsing algorithms or code libraries. Although unlikely, the potential risk exists. Open source tools may offer support in the process of verifying results. Open source solutions may offer some support, and although their development and maintenance is onerous and potentially carried out using fewer resources than commercial equivalents, their underlying functionality is available for greater degrees of scrutiny by a practitioner who possesses the requisite knowledge to do so. Consideration should also be given to the use of tools constructed for use on non-Windows platforms (linux-based applications) in order to validate and verify performance and outputs of mainstream provider platforms which are often designed for utilisation in a Windows environment. Generally, the DF field has exhibited trust in the current processes used for acquiring and parsing file system data (See Figure 3).

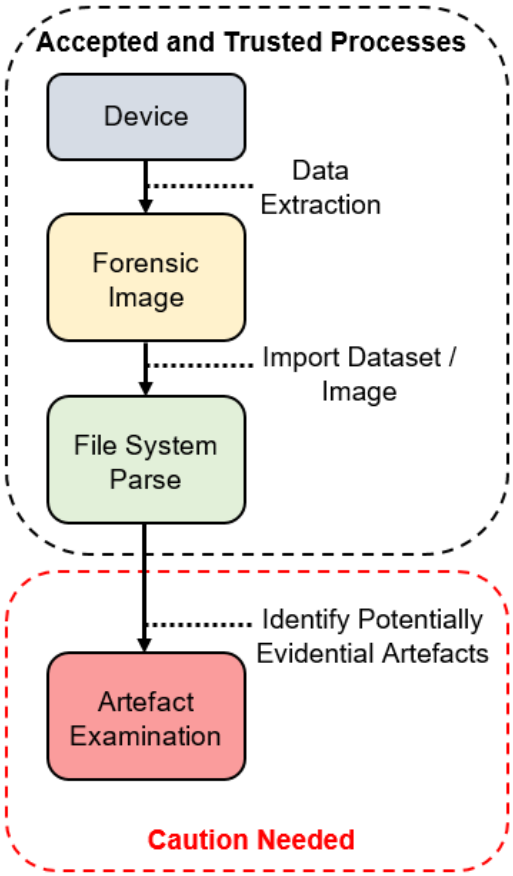

Figure 3: Exhibited trust in the stages of a DF examination.

Applications for more bespoke artefact-parsing and interpretation tasks arguably should be partially trusted. Tools which tackle artefacts which are well documented and understood (usually operating system artefacts), having been subject to long-term scrutiny and research are generally accepted. However, where artefacts are associated with new and emerging technology, caution should be utilized. Many tools offer 'support' for the interpretation of certain artefacts, but what is key to note, is that it is their interpretation of data based on their testing and their algorithm development. This is often a closed process and there is no way to assess the reliability and robustness of such work and the richness of their test dataset and their chances of exhausting all of the possibilities and data formatting which their tool may be confronted with. Whilst this process may be perfect (and this article in no way questions that), what practitioners receive is a 'product' proposed for tackling a certain investigative scenario. Arguably, the practitioner should test and validate a tool before utilising it and this guidance is enforced by both the vendors themselves (see Section 2.3) and overarching forensic bodies, but fundamentally, this may not be considered by some to be a core function of a practitioner's role. However, it should be made clear that part of a practitioner's role (particularly those who regularly engage with court processes and therefore must adhere to evidence admissibility and reliability governance) is to test and validate the tools they use during an investigation which is also encouraged by many forensic tools vendors. Practitioners should maintain confidence in the results that their software tools provide, regardless of vendor statements regarding their previous in-house test iterations. Once this level of confidence is attained, a practitioner can begin to effectively defend their findings. Those engaging in the defence of their investigation findings having

not carried out some level of methodological testing and verification may face significant scrutiny of their work.

Although a generalisation, arguably most practitioners are not capable of software testing (which when done effectively is often an extremely complex and lengthy procedure) and are not likely versed in the principles of software design best practice and evaluation principles. Even if they were, the amount of time available to do this would present an issue. Where practitioners pay for a product there is an expectation that it carries out its tasks to a sufficient standard, no different to the purchase of goods and service in any other area of life. In 2007, Beckett and Slay (2007) suggested there was significant reliance on a tool vendor to test their tools. Arguably some 10 years later, this situation remains. It is not possible to quantify the occurrence of TEs in DF because there is no method for robust vetting of practitioner results on a large scale. Typically, spotting errors in a single case comes down to only a small number of practitioners who have direct involvement.

2.2 The 'grey area': TE, UE or TL?

The distinction between TEs, UEs and TLs is an important one as it allows accountability to be established. Practitioners should not and cannot take their duties in cases of criminal prosecution lightly and mistakes need to be attributed to an individual entity (or where multiple practitioners are involved, those forming part of the erroneous process should be identified), laboratory or organisation to ensure they can be rectified and future harm can be prevented. The success or failure of a practitioner's investigation can be depended on both the quality of the tools they use and their effective implementation. Whilst the distinction in definition between TEs, UEs and TLs is clear, categorizing an event is not, due to the following reasons.

1. *Practitioner knowledge*: The phrase 'Nintendo forensics' is acknowledged by a number of practitioners (Carvey, 2009; Jones, 2016; Read et al., 2016) referring to the use of push-button parsing tools which require little to no understanding of their underlying workings. Whilst this is not applicable in all cases, arguably such mentalities do still exist, where it is easy to initiate a process and draw conclusions solely on its completion and results. Krone and Bell (2010, p.10) allude to this issue stating 'companies that develop forensic tools offer product features such as simplified searching and improved workflow with an emphasis on faster processing and turnaround time. However, these default product features can easily be misused by untrained individuals…..the default settings for several commonly used e-discovery tools are strongly "dumbed down"' (Krone and Bell, 2010, p.10).

   To start, a tool should arguably never be utilised where the practitioner does not fully understand its application both in terms of what it does, and, to the extent that it carries out this task (Sammons, 2012). Therefore where a tool indicates that it will recover information in relation to event 'A', a practitioner must maintain an understanding of what event 'A' is, its extent and coverage, and, whether the tool has effectively delivered on this task. For example, a tool may accurately state that it recovers Internet history, but omit to disclose that it cannot handle the recovery of content from compressed formats, certain browser types (and versions), certain system locations or process unallocated regions of media. In such a case, it would be wrong for a practitioner to assume that no Internet History exists on a device based

solely on the output of this tool alone. Arguably, this could be categorized at a UE. Further, a DF tool is likely never fully up-to-date. Where new releases of software maintain new bespoke structures beyond a tools ability to parse within the confines of a current release, a practitioner must intercept this potential issue. This can only be achieved when two things are in place, first, the practitioner is knowledgeable enough to respond to this issue, and second, the tool transparently defines its current set of limitations.

Tool design can also influence UEs. Whilst at the core of a UE's definition, the user is at fault, bad tool design can almost coerce or encourage mistakes, although research in this area is limited (see Hibshi et al., (2011) for an example of usability research for forensic tools). Vague terminology, poor graphical interface layouts and the over-incorporation of functionality may influence practitioner decision making for the worse, leading to confusion and subsequent tool UEs.

2. *Transparency*: DF maintains an unusual situation where it may not always be able to distinguish if an event can be classed as a TE or TL if a lack of transparency by tool vendor exists. As Daniel (2012) suggests, a DF tool must be predictable in order to be reliable, yet where a complete listing of its functionality is not available, this is not achievable. Documentation accompanying a DF tool is often comprehensive in terms of documenting tool setups and usage but can be lacking in areas of technical processing information. Whilst the effective use of a tool will inevitably prevent UEs it does not provide a full understanding of the confines in which it can be reliably operated. The key issue surrounds reliability and a practitioner needs to know when and when they cannot rely on their chosen tool to provide accurate results.

   A failure to disclose such information in enough detail creates a grey area of uncertainty, taking for example cases of basic data record recovery. This task is often binary; content is either present on a device, or not. A practitioner using a tool which ineffectively defines its usage constraints leads to a practitioner's inability to confirm that the results of their data recovery investigation is factually accurate. In this scenario, (let's assume no TEs are present) there are two potential outcomes. First, the initial volume of recovered records is correct, as the practitioner and tool have performed effectively. Second, the number of recovered records is incorrect as records exist beyond the capabilities of recovery of the tool, unbeknownst to the practitioner. In the former, despite accurate results, the practitioner cannot say with certainty from this process alone that result are accurate as they do not know to what degree their tool is operating, despite performing well. In the later, attributing blame is difficult for missed content is difficult. If a tool has not effectively defined its limitation, it would be difficult to establish that a TE has occurred as it is not possible to establish whether missed data is actually beyond its capability. This may occur where a vague TL is defined (for example, recovery of artefacts from Firefox browser but omits to define what versions). Arguably it could be categorized as a TE as a practitioner should never have placed themselves in a situation of using a tool which does not effectively define its limits. Yet it could be questioned as to whether this is fair if vague limitations have lead to poor decision making taking place.

2.3 Who is accountable?

Where an error occurs, accountability will always be an issue for debate, not necessarily for financial recompense or blame, but for the purposes of rectification, preventing further issues and developing better practice moving forward. Accountability can only be attributed when the facts of a circumstance are known, and in DF these may be difficult to ascertain. An examination of the End User Licence Agreements (EULA) for mainstream forensic tools help to establish the responsibilities involved in tool production and usage. EULAs provide a standard set of terms and conditions to be expected of a software provider, where the following provides some key excerpts from leading forensic software EULAs. Regarding OpenText's (2016) EnCase package (formally that of Guidance Software) the following is stated within the generic UK & Ireland EULA, of which it is assumed these terms would cover those utilising the EnCase software .

> "*Licensee will not disclose results of any benchmark or other performance, evaluation or test run on or related to the software. Licensee acknowledges that the software is not fault-tolerant…OT does not warrant that the software is error-free or will operate without interruption* "

Magnet Forensics (2017a) states.

> *"Magnet forensics specifically does not warrant that the operation of the software will be uninterrupted or error-free. Magnet forensics does not warrant that the software is free from bugs, errors, or limitations…or that all errors in the Software can be found or corrected, although Magnet Forensics shall use commercially reasonable efforts to do so...*
>
> *...Commercially reasonable efforts to correct any defects in the Software which prevent the Software from operating in accordance with its specifications (including the provision of Updates or Upgrades where necessary to correct a defect in the Software"*

X-Ways Forensics (2015) states.

> *"The user must assume the entire risk of using the program"*

Cellebrite (2016) states.

> *"Notwithstanding anything to the contrary in this EULA, Buyer shall not, alone, through a User, an Affiliate or a Third Party (or allow a User, an Affiliate or a Third Party to ...disclose any results of testing or benchmarking of any Software to any Third Party… or reverse compile, reverse assemble, reverse engineer or ActiveUS 133888744v.7 otherwise translate all or any portion of any Software;*
>
> *Cellebrite does not warrant that (i) the operation of any Software and/or Product will be error-free; (ii) all defects in any Software and/or Product will be corrected"*

Forensic software EULAs maintain many of the standard components of a typical software EULA governing software usage; after all, fundamentally they are of the same nature. What distinguishes forensic software, is its typical adoption in criminal contexts, where its usage

can lead to severe consequences for a recipient. Whilst clauses are designed to protect the organization, there are four areas of interest in need of highlighting.

1. *The practitioner is responsible*: Whilst this appears sensible from an organisational business point of view, it is the culmination of all of the surrounding issues of tool verification that leads this statement to potentially cause discomfort. Typically, EULAs warrant that the provided software may not be error-free. As a result, practitioners cannot rely on the tools they receive to be 100% reliable, regardless as to whether they are or not. It would seem that liability for investigative errors lies with the practitioner as in accepting the EULA, they are accepting the chance of errors. As many software providers publicly disclose the release of updates and bug fixes, it can be assumed likely that at any one given period of time, an error (potentially multiple) exists in the DF tools in operation, only their severity and impact on the software's reliability may never be accurately established. Even if practitioners are fortunate enough to identify an error, its impact on historic cases should be established.

   As the practitioner seems to assume responsibility for using the tool, it is implied that sufficient testing of the platform must take place prior to use but post-product. Infact, vendors such as Magnet Forensics (2017b) encourage practitioners to undertake their own forms of validation, providing some guidance on how to achieve this. The issue which exists here is twofold; first a lack of available and effective tool testing strategies for DF, and second, time and resources to effectively test tools makes carrying out the task infeasible for many industry practitioners. Effective testing can be a resource-intensive, costly and time consuming process (Atkinson, 2014), especially given the complexity of DF software. It is unlikely that anything other than top-level errors (likely GUI-driven issues or basic data parsing discrepancies) would be detected, where more complex, impactful issues (should they exist) are unlikely to be detected. Further, a lack of comprehensive test-datasets for software testing creates an issue (Becker et al., 2017), where despite calls for practitioners to self-generate these resources, such an approach is just not feasible in most cases. In an ideal scenario, a practitioner needs a tool to function reliably as depicted, straight 'out the box', a potentially impossible request.

2. *Reasonable updates will be made*: Whilst updates will be made, what is deemed reasonable still remains vague. Likely this will cover anything which would impact its successful usage, however it also raises questions as to what errors are acceptable in a tools sustained usage. It would be unlikely that any major vendor would be anything other than helpful during the process of rectifying an error discovered during operation. In such cases, practitioners and vendors must work together to identify and fix software issues, with both parties playing an important role in software maintenance and development.

3. *Potential restrictions on the publishing of error rates/tests*: This is perhaps potentially the most restricting in terms of stunting the discipline's pursuit of evidence reliability. Flandrin et al., (2014) suggest that very little work has been done regarding tool testing and validation in DF. Given that some EULAs may prohibit the disclosure of results from benchmark testing, there may be no way to accurately assess the volume of testing which is actually taking place (although some disclosures have

been made - see Newsham et al's (2007) work with EnCase and The Sleuth Kit). As a discipline, knowledge distribution, particularly in the context of procedural limitations is key to sustained effectiveness. Any limitations noted in a software EULA mean that practitioners or researchers who do undertake their own testing and subsequently find errors are left only with the option of reporting these findings back to the vendor. Whilst sensible, providing an opportunity for those best positioned to rectify the issue (if it is reasonable, as noted above), it prevents the timely dissemination of this information to fellow peers in the field. Whilst a tool vendor may opt to inform it's users of an issue, or rectify it in a future update, the inability to disseminate tests results performed on a platform may serve as a twofold-deterrent.

a. An inability to disseminate test results may lead to a failure to report such issues in the first instance. Practitioners may simply identify a work-around within the confines of their practice and incorporate this into their standard operating procedures, leaving others vulnerable to the sustained usage of a limited tool.
b. A practitioner who sees no method of disseminating testing results to their peers may opt not to undertake such tests in the first instance. Covert testing where results are reported back to the vendor with no way of seeking recognition from fellow peers may not appeal to the practitioner. A morally correct stance may take the view that testing is for the greater good of a chosen field, and therefore its undertaking should occur. However, in reality, it is rarely financially viable to undertake the significant workload of testing DF tools without some form of recompense.

Any limits placed on the disclosure of testing is a method of protecting the reputation and integrity of a product and this is clearly understandable. However, the disclosure of tests (if carried out in a robust and reliable way) can be a positive contribution helping to instill confidence in both a product where results are positive, but also where issues arise and there's a clear and prompt vendor acknowledgement and response. Practitioners and vendors should work together to further tool developments.

4. *Limited access to code*: It should first be noted that should code access be provided, it is unlikely that such a maneuver would result in the widespread effective testing of tools. The complexity and knowledge-base required would likely leave only a small number of individuals in a position to effectively scrutinise the code to the required level. Yet on face value, no code access means that the testing carried out by any individual beyond the vendor will always be limited to black-box approaches (and their associated limitations) (Chessman, 2017). However, effective testing and validation requires source code access. Chessman (2017, p.196) states that the 'strongest reason that defendants must have the opportunity to analyze source code is to identify unknown unknowns'. Essentially these are errors which lie beyond existing knowledge, where a tool is wrongly assumed to be accurately functioning, with reference drawn to challenges made against the Intoxilyzer 5000EN in 2015 (Chessman, 2017). At present, practitioners are often reliant on those who create their tools to test them, and arguably, they are at present best placed to carry out this task.

EULA's provide a standardised method of defining the terms and conditions of software usage, and whilst they offer protection to the vendor, they can be restrictive to the user (Electronic Frontier Foundation, 2005). Whilst it is easier to consider the stance that a software provider should be producing a tool which is error-free beyond reasonable doubt, it is not practical. Further, it should be questioned as to whether it is reasonable to expect such levels of liability to be shouldered by the vendor themselves; it would arguably make no commercial-sense. Instead the field would likely see a decline in the number of providers willing to shoulder the burden as financial incentives may not outway the potential penalties associated with error-discoveries. Alternatively it may lead to significant price increases, in order to compensate for greater penalties. The feasibility of tackling the generation of error-free DF tools must be measured against the challenge. DF tools are attempting to tackle a moving problem. They produce a static release followed by periodic updates but the rate of change in technology far exceeds what most, if not all vendors are most-likely capable of keeping up with. The question should be asked as to whether it is sensible to hold vendors to account for every issue, and, it is arguably not.

3. So what's the issue?

There are some trade-offs and challenges facing the field of DF which do not sit easily alongside existing tool usage practices and their associated testing. DF now forms a part of many criminal offences and is increasingly becoming the primary source of evidence in many cases. It is also more frequently forming the sole form of prosecuting evidence in cases in both cyber enabled and cyber dependant offences. As the impact of DF in criminal regulation is increasing, so must the standards which rigorously test and hold its evidence to account. Tool testing and verification is needed, as is reliably establishing the confines in which a tool can be reliably used. As digital evidence becomes more prominent, so does the level of reliance placed upon it, and scrutiny it is subject to.

Digital offences and subsequent workloads are also changing. No longer do practitioners have the time to investigate every file resident on a device (often viewed as a traditional approach); in fact it has been this way for some time (Pollitt, 2013). Time constraints now place significant emphasis on the speed and precision at which relevant evidential content can be identified and examined through triage processes with dependance on the use of automation has never being greater. In triage cases where the sustained regulation and evaluation of sex offenders is taking place, it is common for practitioners to have up to an hour to examine and make a decision on a given digital device in the field, with results generated from automated recovery scripts providing the crux of the analysis process. Where time-limited investigations are taking place, it is crucial that a tool's limitations are clearly and accurately disclosed to the user to ensure triage-orientated suitable decision making to take place.

Triage inherently comes with a risk of missing evidential content (Horsman et al., 2014) therefore it is imperative that practitioners can make informed decisions about the confines of their chosen tools performance and rely on the tool to operate correctly within these boundaries. The discipline is moving away from technical processing in favour of data sifting and triage informed decision making. To adapt a well known phrase, practitioners are tasked with finding an evidential needle in a data haystack. Only now, the haystack is continuously growing in size, the needles are arguably smaller and continuously evolving in shape, and

the time available to complete the work has shrunk despite the task being more complex. Whilst the field has not yet mastered the use of solely autonomous processing of data (where subjective and evidential decision making can be applied reliably), it is now relying on automated processing more than ever. As a result, the accuracy of these processes must be validated in order to ensure accurate results.

There are limited avenues which can be taken for driving forward current approaches to ensuring the reliability of results from DF tools, however suggestions for improvement should be made given the recent imposition of regulations governing validation and verification in UK laboratories in the form of ISO/IEC 17025.

3.1 A formalised error / tool limitation discovery repository?

On face-value this may breach some EULAs, but only if it is practitioner driven. Vendors may choose to provide an open repository of disclosures surrounding identified errors and crucially, limitations which define the boundaries in which their product can reliably operate. The level of detail in known limitations can never be enough (everything can be factored into sound decision making by a practitioner) and this information may be best delivered by a vendor as a proactive measure. Practitioners need the details of a tool's known-good operating confinements in order to rely upon its use. Perhaps a closest comparable example is the National Institute of Science and Technology's (2017b) 'National Vulnerability Database' where vulnerability information is consistently posted to warn others. Whilst beyond the confines of DF software as it offers a neutral form of reporting with no real vendor association, it does provide an example of a repository which can be queried for supporting knowledge and decision making with continuous up-to-date submissions, where a comparable structure for DF tools would arguably be welcomed by practitioners. A structure of this type would provide a source of known-good information regarding the operation of the tools they use and a single point of contact.

However, as with most knowledge-based systems, maintenance is an issue requiring significant investigation by those who wish to take on such a task. Magnet Forensics (2017c) 'Artefact Exchange' portal and the 'ForSci' repository (JISC, 2017) provide a useful example of encouraged knowledge exchange between practitioners and vendors, despite being beyond the confines of tool testing and limitation. In this case, to provide maximum practitioner benefit, a collaborative effort from all of the major tool providers would be needed, which due to natural industry competition may not be feasible.

3.2 Increased procedural and testing disclosures

Here, a practitioner would benefit from a greater transparency in the testing and validation procedures which have been undertaken at the time of a tool's development. Through an understanding of the testing that has taken place, a practitioner can make an informed decision as to the potential reliability of a tool. Whilst this would provide an ideal situation, it is arguably also an unobtainable one. Disclosure of validation processes would almost certainly see increased scrutiny of vendors which they may view unfavourably. Where as transparency would arguably increase trust in the software development process, it may also unearth practice limitations should they exist. Whilst such discoveries may benefit the DF field, a vendor is unlikely to subject themselves to such risk. Perhaps the issue here lies more with the relationship between tool vendors and practitioners and calls for greater forms of collaboration may be needed. While vendors may employ knowledgeable software

engineers for tool production, not all may possess dedicated DF knowledge. In turn, human error is always a factor given the complexities of software development in a DF context. In this scenario, engaging with practitioners during a tools development process for transparent and collaborative validation may lead to stronger relationships and a greater chance of preventing TEs and UEs before a tool enters the market. Despite beta versions of software providing some insight, engagement at an earlier stage of development would arguably be more beneficial where parsing algorithms can be subject to analysis. However, such sentiments may not sit well with the need to protect intellectual property and organisational information.

3.3 Increased functionality disclosures

DF tool vendors need to ensure that the disclosures made following the release of a product or update carry sufficient detail to warrant the correct identification of a TL. Whilst it would not be correct or necessary to single out particular vendors here, it is necessary to discuss in a hypothetical sense. Where a tool states support for a task or process, disclosed details should be scrutinised to ensure sufficient information is provided. As a proposed minimum the following items of information would offer practitioner support:

1. *Tool development timeframe*: Divulging the timeframe which a product or update has been subject to development helps a practitioner to identify limitations in its use. For example, a tool which has been under development from 01/01/2015 - 01/03/2015 can not reliably be expected to tackle evidential data produced by a software package or artefact which was developed beyond the actual DF tool's developmental stage. Whilst it may still function correctly, it cannot be expected to do so as developers would not have been able to test these subsequent iterations (and sole responsibility for its usage in these scenarios lies with the practitioner). Disclosure of a tool's development timeframe acts merely as a marker for a practitioner to make sound decision making around using the tool.

2. *Tested platforms*: Where a tool is aimed at tackling the parsing of data from a specific software platform of application, the tested platforms should be disclosed. For example, software which targets the interpretation of data from WhatsApp must disclose all versions which their tool has been tested against. Currently (at the time of writing), WhatsApp's most recent offering is version 2.17.426 having undergone a significant number of software iterations. A practitioner should be able to note which versions of WhatsApp their currently operated tool version has been successfully validated against, and which it has not. This requires documenting every minor as well as major release as even micro interactions can manipulate metadata structures, impacting the functionality of a tool in the future.

3. *Test conditions*: Practitioners need to also understand the setup of any testing which has taken place, including platforms used, test data used and functionalities tested. Many applications maintain varying internal functionalities. Defining that there is 'support for an application' is not sufficient, the specific functions of an application and associated data types should be disclosed. The defining of functionality support is an area which is often done well by many existing platforms.

3.4 Test data disclosures

As testing has already taken place, the disclosure of test data sets used by vendors would be valuable resource to the DF community and a platform from which to build self-testing procedures. It would not only provide a source of information from which to build on existing testing practices but also allow further iterations of tools to be manually checked by practitioners for reliability; removing the future burden of testing from vendors. As EULAs provide the need for testing, providing test data which was initially used would allow for this volume of scrutiny to take place and continue to take place once the tool is in the hands of the practitioner. It may also allow weaknesses in testing to be established as the richness of test data can be assessed to establish whether further work is required.

3.5 Alerts and error handling

Whilst vendors frequently release updates of their tools noting additional support, conversely, releases of terminated support would be of equal benefit. Noting when a tool or tool version is no longer capable of tackling a problem prevents a practitioner from mistakenly using it. A disclosure of 'no-support' does not have to be perceived a negative concept, as vendors could provide a timeframe for when future support will be offered, whilst during this period practitioners are prevented from mistakenly utilising the tool in their investigations. Further, practitioner would arguably support such disclosures and be encouraged to work more closely with vendors given their openness in support their working procedures. Error handling may also offer an opportunity to support a practitioner in their case-processing. Vendors who provide a greater level of detail in the error-handling provided by their tool stand to educate practitioners on the potential limitations of their tools usage. Tool dialog feedback is often results-focused, disclosure record information or in some cases the lack of. Errors which provide more descriptive information may be advantageous, focusing on whether analysed data was of the format to be expected, flagging any perceived anomalies in structure or inconsistency warnings.

3.6 External Factors

It is also necessary to consider the impact of of measures which can be utilised to counter potential tool usage issues. Whilst TEs may provide a concern, there are also non-technical practitioner UEs to be considered, with Sunde (2017, pg.17) alluding to factors such as practitioner competence, 'misinterpretations of the meaning, value or reliability of a piece of evidence, a biased decision, or essential evidential information being overlooked'. Examiner experience cannot be understated and in some cases, a practitioner's experience and understanding, gleaned from previous cases may protect against the occurrence of undetected tool issues and potentially allow any non-valid tool outputs to be detected and addressed before they form part of a final report. Conversely, a lack of experience can increase a practitioner's vulnerability to a tool UE.

Where a practitioner has experience of certain processes and their expected outcomes, results which are produced that sit in conflict with those which were expected can potentially be identified and subjected to additional scrutiny. In some cases, the application of basic technical knowledge may make it possible to establish valid conclusions about any recovered digital data. However, those with more limited experience may not be able to call upon such knowledge, and in these cases, may be reliant on peer-review processes from other members of staff in order to highlight and query such results. Training also maintains an important role, and practitioners should receive appropriate training for using the software and tools which they apply as part of their investigations. Despite such measures appearing

sensible, practitioners may not actually in practice receive training bespoke to each and all of the tools that they utilise in their work (potentially due to resource and costing issues). Those practitioners operating without formal tool training may find it more difficult to detect a TE or in turn, may be susceptible to a tool UE. Similarly, laboratories which prioritise speed of examination may subsequently increase their vulnerability to practitioner UEs (Pollitt et al., 2018). Finally, effective management and supervision may also limit or prevent tool UEs from remaining undetected through effective work scrutinisation policies by those with the requisite experience. Peer-mentoring schemes for inexperienced practitioners may also provide a 'safety net' for the detection of potential errors. however, such suggestions all may raise resourcing issues for organisations.

3.7 Implications

Whilst prior suggestions in Section 3 may improve disclosure and transparency for tool testing, they maintain an element of risk. Public disclosures, particularly of TLs and TEs increase the chances of subsequent exploitation of any weaknesses in existing tools. If vendors document the confines of testing, limitations in test scenarios may lead to the design of systems which can circumvent the constraints of current tools. Further, anti-forensic systems and software may seek to specifically design their structures in line with any documented existing weaknesses in forensic tools. As such vendor disclosures would benefit the DF community providing they are made solely to the DF community and associated entities. Distributing such information is in itself a challenge, and one which may ultimately stunt the development such approaches. Whilst vendor forums provide access to such communities, some remain unvetted, allowing access to anyone who wishes to register. Distribution of information to closed communities where access requires validated credentials may provide a solution but raises issue of attracting an appropriate population in order to ensure any information is distributed sufficiently wide-enough to practitioners.

There are also implications for the vendor in terms of competitive advantage. The disclosure of limitations may be directly utilised by other organisations to gain a competitive advantage; a concept which is likely to put organisations off from engaging in such schemes. As a result, such approaches would need industry 'buy-in' where all vendors agree to to confines in which disclosures are handled.

4 Conclusions

The process of creating robust DF tools is a difficult one, and the field is fortunate to be in a position where multiple vendors have taken up this challenge, and for this, practitioners are grateful. Whilst arguably there have been minimal instances of severe issues caused by tool usage, DF software is not immune from errors. As practitioners are wholly reliant on such software to carry out their work, the distinction between TEs, UEs and TLs needs to be made and has been in this work. Where mistakes occur, accountability needs to be established to prevent further harm from occurring and for errors to be rectified. For DF evidence to be reliable, the tool used to produce it should be validated and the confines of the tools usage should be clearly identified. This article has presented a discussion of areas surrounding tool testing, tool errors and limitations offering suggestions for improving current tool usage reliability.

References:

AccessData (2017) ‘Forensic Toolkit (FTK)’ Available at: https://accessdata.com/products-services/forensic-toolkit-ftk/ (Accessed: 1 December 2017)

Alvarez, Lizette (2011) ‘Software Designer Reports Error in Anthony Trial’ Available at: http://www.nytimes.com/2011/07/19/us/19casey.html (Accessed: 4 December 2017)

Association of Chief Police Officers (2009) ‘ACPO Managers Guide: Good Practice and Advice Guide for Managers of e-Crime Investigation’ Available at: http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_and_Advice_for_Manager_of_e-Crime-Investigation.pdf (Accessed: 1 December 2017)

Atkinson, J.S., 2014. Proof is not binary: the pace and complexity of computer systems and the challenges digital evidence poses to the legal system. Birkbeck L. Rev., 2, p.245.

Aziz, B., 2014. Modelling and refinement of forensic data acquisition specifications. *Digital Investigation*, *11*(2), pp.90-101.

Becker, C., Duretec, K. and Rauber, A., 2017. The Challenge of Test Data Quality in Data Processing. Journal of Data and Information Quality (JDIQ), 8(2), p.7.

Beckett, J. and Slay, J., 2007, January. Digital forensics: Validation and verification in a dynamic work environment. In System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on (pp. 266a-266a). IEEE.

Brunty, Josh (2011) ‘Validation of Forensic Tools and Software: A Quick Guide for the Digital Forensic Examiner ’ Available at: https://www.forensicmag.com/article/2011/03/validation-forensic-tools-and-software-quick-guide-digital-forensic-examiner (Accessed: 1 December 2017)

Carvey, H., 2009. Windows forensic analysis DVD toolkit. Syngress.

Cellebrite (2016) ‘EULA’ Available at: http://legal.cellebrite.com/us/Cellebrite-EULA.pdf (Accessed: 2 December 2017)

Chessman, C. (2017) A “Source” of Error: Computer Code, Criminal Defendants, and the Constitution’ Calif. Law Rev. 105, 101–193.

Cusack, Brian and Homewood, Alain (2013) ‘Identifying Bugs In Digital Forensic Tools’ Proceedings of the 11th Australian Digital Forensics Conference. Held on the 2nd-4th December, 2013 at Edith Cowan University, Perth, Western Australia

Daniel, L.E., 2012. *Digital forensics for legal professionals: understanding digital evidence from the warrant to the courtroom*. Elsevier.

Electronic Frontier Foundation (2005) ‘Dangerous Terms: A User's Guide to EULAs’ Available at: https://www.eff.org/wp/dangerous-terms-users-guide-eulas (Accessed: 4 December 2017)

Flandrin, F., Buchanan, W., Macfarlane, R., Ramsay, B. and Smales, A., 2014, September. Evaluating digital forensic tools (DFTs). In 7th International Conference: Cybercrime Forensics Education & Training.

Forensic Science Regulator (2015) 'Draft Guidance: Digital Forensics Method Validation' Available at: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/485037/2015_12_14_-_Digital_Forensics_-_validation_-_draft_guidance.pdf (Accessed: 15 March 2018)

Guo, Y., Slay, J. and Beckett, J., 2009. Validation and verification of computer forensic software tools—Searching Function. Digital Investigation, 6, pp.S12-S22.

Guttman, B., Lyle, J.R. and Ayers, R., 2011. Ten years of computer forensic tool testing. Digital Evidence & Elec. Signature L. Rev., 8, p.139.

Hibshi, H., Vidas, T. and Cranor, L., 2011, May. Usability of forensics tools: a user study. In *IT Security Incident Management and IT Forensics (IMF), 2011 Sixth International Conference on* (pp. 81-91). IEEE.

Horsman, G., Laing, C. and Vickers, P., 2014. A case-based reasoning method for locating evidence during digital forensic device triage. Decision Support Systems, 61, pp.69-78.

James, J.I., Lopez-Fernandez, A. and Gladyhsev, P., 2013, September. Measuring Accuracy of Automated Parsing and Categorization Tools and Processes in Digital Investigations. In *International Conference on Digital Forensics and Cyber Crime* (pp. 147-169). Springer, Cham.

JISC (2017) 'Developing ForSci: an improved tool for making forensic-based research openly accessible' Available at: https://www.jisc.ac.uk/events/developing-forsci-an-improved-tool-for-making-forensic-based-research-openly-accessible-29-jul-2017 (Accessed: 6 December 2017)

Jones, N., 2009. IT forensics: 22 years on. International Journal of Electronic Security and Digital Forensics, 2(2), pp.116-131.

Kirchner, Lauren (2017) 'Traces of Crime: How New York's DNA Techniques Became Tainted' Available at: https://www.nytimes.com/2017/09/04/nyregion/dna-analysis-evidence-new-york-disputed-techniques.html (Accessed: 4 December 2017)

Krone, Winston and Bell, Megan. (2010) 'Software Bugs in Common E-Discovery Search Tools' Available at: https://kivuconsulting.com/wp-content/uploads/2012/08/Software_Bugs_in_Common_Ediscovery_Tools_Kivu_Whitepaper_Oct_2010.pdf (Accessed: 4 December 2017)

Lyle, J.R., 2010. If error rate is such a simple concept, why don't I have one for my forensic tool yet?. Digital Investigation, 7, pp.S135-S139.

Magnet Forensics (2017a) 'END-USER LICENSE AGREEMENT' Available at: https://www.magnetforensics.com/legal/ (Accessed: 1 December 2017)

Magnet Forensics (2017b) 'Being Forensically Curious: The Process of Discovery' Available at: https://www.magnetforensics.com/blog/being-forensically-curious-the-process-of-discovery/ (Accessed: 4 December 2017)

Magnet Forensics (2017c) 'ABOUT THE ARTIFACT EXCHANGE' Available at: https://www.magnetforensics.com/artifactexchange/ (Accessed: 4 December 2017)

Moore, R., 2013. Standardisation: A tool for addressing market failure within the software industry. Computer Law & Security Review, 29(4), pp.413-429.

National Institute of Standards and Technology (2017) 'Computer Forensic Tool Testing (CFTT)' Available at: https://www.cftt.nist.gov/ (Accessed: 5 December 2017)

National Institute of Standards and Technology (2017b) 'National Vulnrebility Database' Available at: https://nvd.nist.gov/vuln/search/results?adv_search=false&form_type=basic&results_type=overview&search_type=all# (Accessed: 6 December 2017)

Newsham, T., Palmer, C., Stamos, A. and Burns, J., 2007, August. Breaking forensics software: Weaknesses in critical evidence collection. In Proceedings of the 2007 Black Hat Conference.

OpenText (2017) 'EnCase® Forensic' Available at: https://www.guidancesoftware.com/encase-forensic (Accessed: 1 December 2017)

OpenText (2016) 'End User Licence Agreement - UK & Ireland' Available at: https://www.opentext.com/file_source/OpenText/en_US/PDF/opentext-legal-eula-v3-uk-ireland-en.pdf (Accessed: 1 December 2017)

Pan, J., 1999. Software testing. Dependable Embedded Systems, 5, p.2006.

Pollitt, M.M., 2013. Triage: a practical solution or admission of failure. Digital Investigation, 10(2), pp.87-88.

Pollitt,Mark., Eoghan Casey, David-Olivier, Jaquet-Chiffelle and Pavel Gladyshev (2018) 'A Framework for Harmonizing Forensic Science Practices and Digital/Multimedia Evidence' Available at: https://www.nist.gov/sites/default/files/documents/2018/01/10/osac_ts_0002.pdf (Accessed 15 March 2018)

Read, H., Thomas, E., Sutherland, I., Xynos, K. and Burgess, M., 2016, January. A Forensic Methodology for Analyzing Nintendo 3DS Devices. In IFIP International Conference on Digital Forensics (pp. 127-143). Springer International Publishing.

Sammons, John (2012) 'The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics' Elsevier

Sunde, N., 2017. *Non-technical Sources of Errors When Handling Digital Evidence within a Criminal Investigation*(Master's thesis).

SWGDE (2014) 'SWGDE Recommended Guidelines for Validation Testing' Available at: https://www.swgde.org/documents/Current%20Documents/SWGDE%20Recommended%20Guidelines%20for%20Validation%20Testing (Accessed: 5 December 2017)

SWGDE (2017) 'SWGDE Establishing Confidence in Digital Forensic Results by Error Mitigation Analysis' Available at: https://www.swgde.org/documents/Current%20Documents/SWGDE%20Establishing%20Confidence%20in%20Digital%20Forensic%20Results%20by%20Error%20Mitigation%20Analysis (Accessed: 5 December 2017)

Talib, M.A., 2016. Towards early software reliability prediction for computer forensic tools (case study). SpringerPlus, 5(1), pp.1-12.

Wilson, Craig (2011) 'Digital Evidence Discrepancies – Casey Anthony Trial' Available at: http://www.digital-detective.net/digital-evidence-discrepancies-casey-anthony-trial/ (Accessed: 1 December 2017)

X-Ways (2015) 'License Agreement for Software Products of X-Ways Software Technology AG' Available at:http://www.x-ways.net/license.pdf (Accessed: 1 December 2017)

X-Ways (2017) 'X-Ways Forensics: Integrated Computer Forensics Software' Available at: http://www.x-ways.net/forensics/ (Accessed: 1 December 2017)