Validation for Digital Forensics

Robert F. Erbacher
Department of Computer Science
Utah State University
Logan, UT 84322
Robert.Erbacher@usu.edu

***Abstract*—This paper discusses the issues revolving around the need for validation and error in digital environments for the admission of evidence. Second, we discuss the how digital evidence intrinsically differs and conflicts with privacy protection that is intrinsic to traditional computer security. Finally, we begin to lay the theoretical foundations for the provision of validation and the computation of error in digital environments. This becomes particularly critical in software environments in which error has not traditionally been considered.**

Keywords- Validation; Error; Digital Forensics; Computer Security; Legal and Privacy Issues

I. INTRODUCTION

Computer security research has traditionally been concerned with issues of privacy as one of its principle goals. This field of research is concerned with protecting any information that might be considered private or that could lead to the compromise of privacy. This must also consider compromises that can be used to indirectly and slowly compromise data. For example, consider the recent attacks in which the secret questions used by some sites, should the user forget their password, are divulged through social networking sites [23]. In such a circumstance it is helpful to protect all information related to the user, including what sites they visit that could be used to glean such information.

Digital forensics on the other hand is focused on providing the evidence needed to prosecute individuals should they commit a crime using digital media. In this scenario, law enforcement must be able to accurately track user activity, including identifying what web sites they may have visited. This can be used to both validate that the perpetrator did in fact commit the crime but also identify how the perpetrator acquired the information necessary to perpetrate the crime.

As can be seen in this example, the needs of privacy and digital forensics do not necessarily coincide. At the very least, digital forensics will entail the protection of additional private information. However, the need for legal admissibility requires many additional requirements be placed on the data.

These requirements focus on the need for validation of the data. This is based on several requirements: attribution, validation, modification resistance and detection, and deletion prevention. Note that for this purpose privacy is not a requirement, though it often is treated as such to provide protection of the network.

Most of these requirements are similar to traditional security needs, specifically:

- **Attribution** – the identification of the origin of the events. This can include the source system, the application, and the user.
- **Modification resistance and detection** – the evidence log must not be easily modified and modifications must be detectible. In terms of forensics, these modifications would actually provide additional evidence.
- **Deletion prevention** – the evidence log must not allow entries to be deleted. When deletion does occur, the entries must be recoverable.
- **Validation** – in essence, each of the above requirements must be provably valid to a jury. In addition, all applications must be proven valid. In association with this is the identification of error rates associated with results to be presented in court such that juries can make the best decision possible as to the acceptability of the evidence. This will essentially require that the data, applications, and results be scientifically proven as valid.

While other researchers have identified the need for error [5][6], they have not gone to the full extent of identifying what is needed in terms of validation for legal admissibility. Additionally, no complete groundwork has been identified for the source of error in digital evidence. We lay out this groundwork, identify the ramifications of not having the validation or error and begin to identify how this error can be calculated. This paper will provide the foundation for future work in this area, research that is sorely needed to maintain the admissibility of digital evidence in the future. The presented foundation is designed to work with the results of other fundamental foundations examined by other researchers, such as Peisert, Bishop, and Marzullo [14].

II. APPLICABILITY

When discussing validation and error we must consider two domains. First is civilian legal admissibility. Second is military need. With civilian legal admissibility, the goal is to provide the scientific validation necessary and associated error for the evidence to pass Frye hearings [21]. This validation must also be sufficient to convince a jury as to the validity of the evidence that is generated through digital

means, essentially making prosecutions more robust. This will include the raw generation of the data and the results of any analysis processes and tools applied to the data to aid in the analysis of the data.

In terms of military needs, the goal is to provide evidence to reduce mistakes and to provide validation that the decisions were appropriate due to the provided error rates should mistakes occur. This will arise during after action reviews or court martial proceedings. Court martial proceedings could essentially revert to legal requirements. The goal is to have sufficient validation and understanding of error rates such that it can be proven that an action was appropriate, especially given the data available.

In both military and civilian uses, it is necessary to provide a complete comprehension of the error being generated, including its sources and magnitude. For this reason, it is typically necessary to include mean error rate and standard deviation. The mean error rate and standard deviation will aid in understanding the implications of the error rate. For instance, an error rate of .75% may not seem significant. However, if the mean error rate is .1 and the standard deviation is .05 then .75% would be significant simply due to its deviation from typical error rates. Thus, the standard deviation essentially provides an indication as to the magnitude of the error and provides an indication as to whether the error is truly significant or not. This additional detail will better allow decision makers to interpret the data and respond appropriately.

The error rates and standard deviation within military uses will allow decision makers to make better decisions as to the validity of the results. In terms of legal admissibility, these values would provide juries with a better understanding of the validity of the evidence. Such error rates are expected in other forms of evidence and providing such values with digital evidence would increase the acceptability, validity, and admissibility of digital evidence.

III. Validation versus Error

Intrinsically, we must consider the implications of both error and validation when employing digital evidence for legal admissibility. Error essentially identifies the likelihood that the result is wrong; how precise or not the result is. Validation identifies whether the actual solution is correct in terms of acceptance by the scientific community. Validation has implications throughout the forensic analysis process, such as:

- How do we know the error rate is correct?
- How do we know any assumptions are correct?
- How do we know the formulas and algorithms are correct? Here correct would imply both the use of the appropriate algorithm as well as a correct implementation of the algorithm.
- How do we know the software is correct?
- Can we rely on the integrity of the data?
- Can we rely on the integrity of the process?

Many other examples for the need for validation can be found. The issue is how we can validate the entire process and all components of the data generation and results generation sufficiently in a scientifically accepted manner. In the end, error must also be appropriately generated. Many examples of failed validation can be found which have led to the inadmissibility of evidence and the dropping of charges against suspects.

IV. Failure Examples

Multiple examples of failures related to digital forensic have recently come into play. The first is from the release of breathalyzer source code. This source code had numerous bugs and poor coding. Even though these issues likely do not affect a suspect, they will be used by defense attorneys to argue the invalidity of breathalyzers as evidence. Even more problematic is the algorithm used to compute the actual alcohol content [17]. Even though it appears the algorithm may be valid, it has never been scientifically validated. This lack of scientific validation for the algorithm could invalidate it according to the Frye test and prevent breathalyzers from being admitted as evidence. This essentially invalidates for legal admissibility a technology that has been in use for decades, at least for the identified equipment.

A second example arises from FBI fingerprint analysis, namely the ACE-V methodology. As seen in [16], this method has been used extensively by the FBI but has recently been attacked because it has never been formally vetted scientifically. The ruling in fact criticizes the fact that the error rates associated with the technique are not known. Another issue that could be associated with FBI fingerprinting is that their system is supposed to return potential matches. However, it does not appear that they have ever validated that that the system will actually return all potential matches; i.e., failing to return a potential match is essentially not identifying a potential suspect and reducing the suspect pool. This may inappropriately place undue focus on another suspect. Thus, the scientific vetting of digital forensic techniques and identification of associated error rates is becoming ever more critical for digital forensic evidence and associated tools to remain legally admissible.

A third example arises in the use of radar guns to detect speeders. Such devices must be calibrated [1] to be legally admissible. Further, the devices used to calibrate the radar guns must also be verified. Finally, these devices do in fact have reported error rates based on the particular equipment in use. As lawyers and the public become more attune to such legal requirements, they are challenging the admissibility of such evidence to a greater extent. It is becoming far more common for suspects to specifically request the calibration records and documentation on the method and tools used in the calibration process. Digital forensics must be prepared for such increasing challenges to its admissibility.

Finally, a recent example has occurred with respect to LIDAR based speed guns [18]. Speeding tickets given because of LIDAR have recently been thrown out since LIDAR has not been scientifically proven accurate. Here, accurate would be completely dependent on consistency and error rate; i.e., error rate and standard deviation. Since these values are not currently available, judges have determined that LIDAR currently is not admissible in court. It was

specifically indicated that the fact that LIDAR is widely used to detect speeds is insufficient for determination of admissibility. Again, this validation process will likely be easily performed but without this validation, the data will not be legally admissible.

The importance of these examples is to consider what is going to happen as lawyers continue to become more knowledgeable with respect to digital evidence and further challenge existing capability. These challenges could make existing techniques inadmissible as most techniques have not been formally validated, do not have associated error rates or standard deviations, and in fact it's not clear how to calculate such values in most cases. This problem actually becomes more significant with software analysis processes and software generated data.

V. VALIDATION

Validation needs to be provided at all steps in the forensic process, including:

A. Data generation

This primarily relates to attribution and providing assurance that an identified identity did in fact generate the data. Attribution, if possible, must identify the system generating the data, the application creating the data, the user running the application, when the data was created, etc. The more parameters that can be validated the more certainty [6] that can be associated with the validity of the data itself and the more acceptable the data will be.

B. Data collection

After data is generated, a repository must collect the data. This will require ensuring that the data is not modified on the way to the repository and providing validation of temporal relationships. These needs for forensics would be insufficient in terms of security, which would also require that the data could not be read and examined in transit.

A related issue is ensuring that the data actually makes it to the repository. The loss of data is especially problematic when considering legal admissibility and forensic analysis. Standard database two-phase commitment protocols should identify and prevent loss of data.

C. Data storage

Once in place at the repository, the system must provide for deletion and modification prevention and recovery. Insertion of elements is not a concern since the level of validation provided for authentic elements ensures that inappropriately inserted elements can be identified. Thus, these inserted elements will in essence provide additional evidence. This again differs from typical security and privacy issues where viewing the data may be a primary concern, which it is not with forensics.

D. System validation

System validation is associated with data generation and requires the unique identification of systems, identification of system restarts, identification of changed system configuration and attributes, and validation that messages were in fact generated by the designated system. The goal is to be able to identify when a malicious system, application, or user may be infiltrating the network. More specifically, it must be noted when a known system's attributes suddenly changes substantially. Sudden changes would indicate that the system is in fact different, whether malicious or not. This would be identified in logs when the system first connects to the data storage repository.

E. Application validation

Application validation is similar to system validation except applied to specific applications running on a system. As with system validation, it must be verified that the application is expected to be sending the events and that the application itself matches known characteristics. Application restarts, the user starting the application, and application parameter settings can all be of critical importance in determining the validity of the events generated by the application.

F. User validation

User validation attempts to provide validation of the users of a system as discussed in relation to system validation. More specifically, it is important to verify the user that started the application that is generating events and specified its parameters. Secondarily, it is important to know other users active on the system to verify the integrity of the system as a whole.

G. Algorithm applicability

The goal of algorithm applicability is to validate that the chosen algorithm has in fact been scientifically proven to generate correct results and to be appropriate for the given application in digital forensics. This is akin to validating acceptability for legal admissibility under Frye standards [21]. Examples in section IV identify some failures related to this topic.

H. Algorithm implementation

Given that an algorithm itself has been validated, the implementation must be similarly validated. Errors often occur in the transcription from a theoretical algorithm to an implemented algorithm. For example, SSH uses a well-established protocol for initiation of a connection and for maintaining the security of that connection. This protocol is well validated. However, there have been well-known bugs in the implementation of the SSH protocol [26] that have allowed it to be compromised.

Formal proofs are the traditional mechanism to verify that an algorithm or process satisfies the expected or needed requirements definition. Additional proofs and verification are then needed to verify that the implementation matches the theoretical model. The difficulty is that formal proofs of theoretical models can be challenging and formal proofs of implementations are essentially unfeasible with no sign that this will improve. Failing formal proofs, the goal is to provide sufficient testing and evaluation to ensure that the implementation matches requirements specifications. However, software testing itself has not been formally

validated. It cannot be guaranteed that software testing will examine all possible cases and scenarios and ensure that a particular scenario will not result in an error. Eventually, software testing has the potential to provide a needed validation mechanism. Software testing could also potentially aid in the identification of error rates. Substantial advancement is still needed in software testing to achieve this goal.

VI. Formal Proofs and Verification

The goal behind having a formal proof and verification is to provide validation that the underlying theories are valid, proven, and accepted. This is geared towards meeting the Frye standard. Intrinsically, we must ensure that the underlying algorithms and equations used in software has been vetted and is scientifically justified if the results of those algorithms and equations must be legally admissible. Formal proof and verification can include, but is not limited to:

- Formal methods – Formal methods relates to formal proofs that an algorithm or specification is correct. This relates to verifying that design requirements are being met, i.e., pre-conditions and post-conditions.
- Testing – Testing refers to the internal testing of implementations by developers in order to identify, locate, and correct errors.
- Peer evaluation – This include publication in peer-reviewed scientific journals to externally validate the appropriateness and technical soundness.
- External evaluation – External evaluation by independent analysts as to the validity of the implementation and identification of weaknesses or lack thereof. External evaluations have been critical for the acceptance of electronic voting machines. Without such external validations, electronic voting machines have not been trusted.

The key question that arises is which technique(s) singly or in unison provide the best level of accuracy and greatest robustness in terms of reduction in errors and error rates while still being feasible.

VII. Calculating Error in Digital Forensics

As mentioned, error has not been previously considered at all in digital forensics or in computer systems in general. This foundation for the calculation of error in computer systems is designed to provide a starting point for more robust and formal derivations of error in the future. Error can occur in the points of the digital forensics process examined below.

A. Data collection

Data collection error refers to observed data such as network traffic data. Error in this case refers to missing data elements and is well-established [13]. Additionally, there may be other sensors, especially in military settings, that provide measuring data. Such measurements themselves can have error rates associated with the accuracy of the measurement. Such error rates are typically provided by the manufacturer of the sensor. The failure to provide error rates essentially resulted in the inadmissibility of LIDAR data as evidence.

B. Data age

The older the data is, the more likely it may be incorrect or irrelevant. This can occur for instance with meta-data associated with GIS data. Old data has led to the military attacking incorrect or inappropriate targets [7]. The impact of data age is very dependent on the particular domain. Identifying the impact of data age can be computed as follows:

error=data_age * (#_elements_changed / total_elements) / duration_of_observation * 100%

C. Temporal order

Temporal order becomes an issue with most network evidence as the exact order of events often cannot be guaranteed and slight changes in event order can change the interpretation of the events. This is also a well studied area [10][11]. However, possible errors in temporal ordering are never reported.

D. Precision

Computers are typically designed to use some form of fixed precision computation. Arbitrary precision libraries are available but not often used. Using such libraries leads to additional compute requirements. This leads to additional error for complex computations. While this may not be an issue with a single computation, with multiple computations the error of truncated values will accumulate until it does become significant.

E. Formulas and compute time

Statistical calculations intrinsically have their own well-established error rate. Similarly, other non-statistical formulas will have their own error rate. This becomes more complicated in scenarios where time is limited, such as military situations in which actionable intelligence must be recovered and acted upon very rapidly. In such circumstances, computation accuracy may be sacrificed for timeliness of results. Numerical analysis and related algorithms in particular would require this trade-off.

F. Software implementation

It is with software implementations that the greatest discrepancy currently exists. Typically, users consider results output by software implementations to be correct and do not consider that there may be an error associated with that output. In addition to the underlying error identified in the previous sections of this paper that the implementation may depend on, the software implementation may have its own error. This will often result from coding errors, incomplete mediation, time-to-check to time-of-use errors, etc. The difficulty here is that it typically is not known how many errors actually exist in a piece of code and what the potential ramifications of them are. It has become accepted that source code will have errors. Work has been done to reduce errors through formal development processes [4], which include

extensive testing, but this has not eliminated all errors. Testing can be used to identify the occurrence rate and potential ramifications of errors but the complexity of software systems essentially limits the feasibility of this approach.

Whole taxonomies of software errors have been created [19][24][25]. For our purposes, we can reduce the scope of error types, as we are not concerned with errors that cause a program to crash or errors that do not add to the generation of results. Surveys on the frequency of occurrence of errors can provide a starting point but will not provide specifics on a particular application. Part of the goal for creating such taxonomies is to better understand what software errors occur and with what frequency [8][15]. This provides a foundation for further research into understanding and preventing such errors in released software [20]. Such taxonomies also aid in ensuring that all errors, especially errors that could result in deviations of results, are thoroughly tested.

For validation purposes, third party validation must be provided. Typical dynamic and static program analysis techniques are designed to identify vulnerabilities and coding errors [2]. The goal is to have these techniques similarly identify error rates based on the aforementioned taxonomies of software errors. As no research on this specific area has previously performed, the first step that is needed is a statistical survey of the generation of incorrect results in software implementations. Given the survey of typical error rates, a first generation system would simply associate error rates with environment characteristics. Future generation systems would use static code analysis to more accurately identify the potential for error within a software implementation.

G. *Analysis process*

Error in the analysis process essentially identifies human error. This could relate to missed evidence or errors in interpretation. This type of error is likely impossible to quantify. The goal therefore is to develop techniques that reduce the likelihood of errors in the analysis process. Such techniques would reduce the demand on the analyst by reducing the amount of textual data needed, assist the analyst in managing the analysis process, provide guidance through the analysis process, and help ensure the process and tasks followed will be legally admissible.

Consider the case identified in [12] in which conflicting evidence as to an individual's car driving speed was identified; both GPS and radar data were available. The analysis process required a detailed mathematical analysis of both data sources to interpolate the drivers speed between readings. The article further states, "… the accuracy of the GPS system was not challenged by either side in the dispute". As this was a court proceeding, this may not have been appropriate as there is an error rate associated with GPS, 3-5 meters for differential GPS [22]. However, the error rate and scientific validity of GPS accuracy may not have been sufficiently substantiated or admitted as evidence.

Information visualization techniques provide an interesting avenue for aiding the analyst but with its own consequences. Visualization intrinsically is not precise and by its very nature cannot be associated with error in a traditional way. Consequently, visualizations can themselves lead to misinterpretations of the data, due to the lack of precision, or missed data, due to occluded data. Given the value information visualization has towards the analysis of large data repositories, research needs to be performed to identify the *formal* implications of information visualization in legal admissibility.

H. *Anti-forensics*

In order to provide complete validation we must consider anti-forensics, including obfuscation and misdirection techniques. Handling anti-forensics techniques also improves the appearance of robustness to the jury. This comes into play since expert witnesses merely need to propose anti-forensics techniques that have not been handled to raise reasonable doubt. In essence, the goal of anti-forensics is to raise doubt as to the validity of the forensic techniques, processes, and data. By considering anti-forensics, the techniques will have additional validation incorporated to ensure that the anti-forensics techniques cannot be considered viable.

I. *Total error*

When considering total error, it must be determined whether the points where error is occurring are sequential or in parallel with other points in which error is occurring; i.e., is the error additive or not? This must also consider if there was error in the determination of the error. In other words, how certain are we that the generated error and standard deviation are correct? The components of the total error must be related to the analyst such that they can interpret it correctly and in turn pass it on to the jury or other decision makers. The analyst must also be willing to stand behind the error rate.

VIII. Conclusions and Future Work

In this paper, we have identified the need for validation and determination of error within digital forensics. This is becoming critical as lawyers become well informed about digital evidence and the methods through which it can be challenged. This is already leading to more Frye challenges to digital evidence and is only going to get worse. When we consider traditional forensics evidence there is always an error rate and/or classification associated with the evidence. This is currently not the case with digital evidence, even though the evidence cannot be guaranteed to be valid. This exemplifies the needs for such error rates in digital evidence.

Most sources of error are well defined. However, error associated with software implementations is not well defined and research is needed to identify how error associated with software implementations can be generated. Many of these issues relate to data provenance [3] issues or extend the typical concept of data provenance.

Aside from identifying software implementation error, the main need for further improvement in legal admissibility of digital evidence is the actual deployment of validation and error generation. This validation covers the realm from

scientifically validating the algorithms, formulas, and equations being used to validating the correctness of the implementation. While some forms of validation are not yet feasible, such as formal proofs of correctness of software implementations, others are well established but simply not sufficiently deployed. As can be seen from recent research, there is growing interest in aspects of error and validation related to digital forensics. However, current research has not had the depth necessary to ensure legal admissibility or examined all aspects of validation and error in a coordinated platform. This needs to be enormously extended.

REFERENCES

[1] David W. Allan, Frank H. Brzoticky, "Calibration of Police Radar Instruments," NBS Special Publication 442, Report of the 60th National Conference on Weights and Measures 1975, pp.42-47. http://tf.nist.gov/general/pdf/87.pdf

[2] Nathaniel Ayewah, David Hovemeyer, J. David Morgenthaler, John Penix, William Pugh, "Using Static Analysis to Find Bugs," *IEEE Software*, Vol. 25, No. 5, Sep./Oct. 2008, pp. 22-29.

[3] Peter Buneman, Sanjeev Khanna, Wang-Chiew Tan, "Data Provenance: Some Basic Issues," *FST TCS 2000: Foundations of Software Technology and Theoretical Computer Science* (2000), pp. 87-93.

[4] Carnot, M., Dasilva, C., Dehbonei, B., Mejia, F., "Error-Free Software Development for Critical Systems Using The B-Methodology," in the Proceedings of *third IEEE International Conference on Software Reliability Engineering*, October 1992, North Carolina, IEEE Computer Society Press.

[5] Brian Carrier, "Defining Digital Forensics Examination and Analysis Tools Using Abstraction Layers," *International Journal of Digital Evidence*, Vol. 1, No. 4, Winder 2003.

[6] Casey, E., "Error, Uncertainty, and Loss in Digital Evidence," *International Journal of Digital Evidence*, 2002, Vol. 1, No. 2.

[7] Carl Conetta, "Operation Enduring Freedom: Why a Higher Rate of Civilian Bombing Casualties," *Project on Defense Alternatives Briefing Report #13*, 18 January 2002. http://www.comw.org/pda/0201oef.html#6

[8] N. Ebrahimi, "On the Statistical Analysis of the Number of Errors Remaining in a Software Design Document after Inspection," *IEEE Transactions on Software Engineering*, IEEE Press, Vol. 23, No. 8, Auguest 1997, pp. 529-532.

[9] Yinghua Guo, Jill Slay, Jason Beckett, "Validation and Verification of Computer Forensic Software Tools—Searching Function," *Digital Investigation*, Vol. 6, Supp. 1, Sept. 2009, pp S12-S22.

[10] Chet Hosmer, "Proving the Integrity of Digital Evidence with Time," *International Journal of Digital Evidence* (IJDE), Spring 2002, Vol. 1, No. 1.

[11] Lamport, L., "Time, clocks, and the ordering of events in a distributed system," *Communication of the ACM*, Vol. 21, No. 7, July 1978, pp. 558-565.

[12] Derek Moore, "GPS or not, teen must pay $190 speeding ticket," *The Press Democrat*, November 4, 2009. http://www.pressdemocrat.com/article/20091104/ARTICLES/911049901/1334/NEWS?tc=autorefresh

[13] Bruce J. Nikkel, **"**Improving evidence acquisition from live network sources," Digital Investigation, Vol. 3, No. 2, June 2006, Pages 89-96.

[14] Sean Peisert, Matt Bishop, and Keith Marzullo, "Computer Forensics In Forensis," *ACM Operating Systems Review (OSR),* Vol. 42, No. 3, April 2008, pp. 112–122.

[15] W. Peng and D. Wallace, *Software Error Analysis*, Silicon Press, February 1995.

[16] Judge Susan Souder, Maryland v. Rose, No. K06-0545 (MD Cir. Ct. Oct. 19, 2007), http://www.baltimoresun.com/media/acrobat/2007-10/33446162.pdf

[17] Lawrence Taylor, "Secret Breathalyzer Software Finally Revealed," *DUI Blog*, September 4th, 2007, http://www.duiblog.com/2007/09/04/secret-breathalyzer-software-finally-revealed/.

[18] Megan Twohey, "Speeding tickets: Use of laser guns in Chicago to catch speeders questioned," *Chicago Tribune*, November 09, 2009. http://archives.chicagotribune.com/2009/nov/09/sports/chi-speeding-tickets-09-nov09.

[19] Giri Vijayaraghavan and Cem Kaner, "Bug taxonomies: Use them to generate better tests," *Proceedings of the Software Testing, Analysis and Review Conference* (Star East), Orlando, FL, May, 2003.

[20] Dolores R. Wallace and Laura M. Ippolito, "Error, Fault, and Failure Data Collection and Analysis," *Proceedings of the 10th International Software Quality Week*, May 1997, San Francisco, CA.

[21] http://en.wikipedia.org/wiki/Frye_standard

[22] http://www8.garmin.com/aboutGPS/

[23] California Office of Information Security: Securing State Information, "Challenge or Secret Questions," Vol. 4 No. 1.

[24] *IEEE Std 1044.1-1995 IEEE Guide to Classification for Software Anomalies–Description*, IEEE Standards Association, http://standards.ieee.org/reading/ieee/std_public/description/se/1044.1-1995_desc.html

[25] Sources of Bugs, http://c2.com/cgi/wiki?SourcesOfBugs.

[26] US-CERT, "Vulnerability Note VU#945216: SSH CRC32 attack detection code contains remote integer overflow," 2001, http://www.kb.cert.org/vuls/id/945216.