Document Title:	**Experimental Study of the Validity and Reliability of Digital Forensics Tools**
Author(s):	**Dr. J. Philip Craiger, Greg Dorn, Heath Crocker, Matt McLain, Kevin Kulbacki, Baber Aslam, Joe Cosmano, Scott Conrad, Dr. Marcus Rogers, Robert Winkworth, Francis Ripberger**
Document Number:	**311530**
Date Received:	**February 2026**
Award Number:	**2009-DN-BX-K237**

Final Technical Report

Experimental Study of the Validity and Reliability of Digital Forensics Tools

Award Number: 2009-DN-BX-K237

Dr. J. Philip Craiger
School of Engineering Technology
Daytona State College
National Center for Forensic Science
University of Central Florida

Greg Dorn
Heath Crocker
Matt McLain
Kevin Kulbacki
Baber Aslam
Joe Cosmano
Scott Conrad

National Center for Forensic Science
University of Central Florida

Dr. Marcus Rogers
Robert Winkworth
Francis Ripberger

Purdue University
Dept. of Computer & Information Technology
CERIAS

Abstract

Digital forensic techniques and tools, as with all other forensic disciplines, must meet basic evidentiary and scientific standards to be allowed as evidence in legal proceedings. In order to be admissible, evidence or opinion derived from scientific or technical activities must come from methods that are proven competencies to be "scientifically valid." Scientifically valid techniques are capable of being proven correct through empirical testing. In practice, this means that the tools and techniques used in digital forensics must be validated, and that crime laboratories, including digital forensic labs, should be accredited or otherwise proven to meet such scientific standards. This task is overwhelming for governmental agencies that lack funding to perform a full-scale validation of **all** forensics tools. Validation is often left to the individual examiners, who may lack the expertise and resources to conduct a scientific validation.

Our project directly addresses the National Academies' concerns related to measurement validity in the digital evidence domain. Researchers from the National Center for Forensic Science (University of Central Florida), Purdue University, and law enforcement digital forensic experts conduct tests to identify issues with the reliability and accuracy of the most accepted software and hardware in use by law enforcement forensic examiners. We conducted approximately 250 validation tests of hardware (write blockers) and software for both Windows and Mac OS X operating systems. Our research design was based on employing these tools to conduct the common forensic tasks across varying operating system and file system conditions following the Scientific Working Group on Digital Evidence (SWGDE) and National Institute of Standards and Technology (NIST) Computer Forensics Tool Testing Guidelines. We selected the

most commonly used commercial forensic suites employed by law enforcement for our test bed based upon feedback collected from a survey of over fifty members of the International Association of Computer Investigative Specialists (IACIS). Our research design includes the most frequently encountered file systems, and includes several file systems for each of Windows OS and Mac OS X. We have also included select hardware write blockers in our research design, as they are crucial to the forensic examiner's ability to duplicate media without changing the original evidence.

We used black box testing to identify issues with accuracy and reliability of our selected hardware and software. In black box testing, the software serves essentially as a "black box" and the performance of the application is evaluated against functional requirements. We created detailed test plans, scripts for installation of operating systems, scripts for forensic tool suite installation, scripts for evidence creation, and descriptions of hardware specifications used for testing in our research. Due to the number of documents, and file sizes, we are unable to include all documentation and reports in this report. These documents are publicly available for download from www.ncfs.org.

The extrapolation of the results of software validation is inherently limited for several reasons. First and foremost is that new versions of software may 'fix' previous detected faults ('bugs'), or even introduce new faults. In addition, these bugs may interact with the testing media (other software, hardware, media under examination, etc.), so that the bug may be apparent only under certain circumstances. *Thus, it is crucial that examiners test their own combination of software version and hardware to identify any discrepancies between expected and actual results prior to conducting a forensic*

examination, or relying upon the results. Accordingly, the results of our validation tests may be extrapolated to only the versions of the software versions used in our study.

Table of Contents

Executive Summary

Introduction

Digital forensic techniques and tools, as with all other forensic disciplines, must meet basic evidentiary and scientific standards to be allowed as evidence in legal proceedings. The requirements for the admissibility of scientific evidence and expert opinion were outlined in the precedent setting U.S. Supreme Court decision in the case of *Daubert vs. Merrell Dow Pharmaceuticals, Inc.,* 509 U.S. 579 (1993). In order to be admissible, evidence or opinion derived from scientific or technical activities must come from methods that are proven competent to be "scientifically valid." Scientifically valid techniques are capable of being proven correct through empirical testing. In practice, this means that the tools and techniques used in digital forensics must be validated, and that crime laboratories, including digital forensic labs, should be accredited or otherwise proven to meet such scientific standards. Obviously strict and accurate validation testing of new forensic tools is required if the results from such applications are to be acceptable as evidence in criminal cases.

As of 2009, when this proposal was written, there was not a substantial body of literature on forensic tool validation. A literature search found very few studies. The primary study again is the CFTT study described above. Other references include Lyle (2003), Bryson & Stevens (2003), Craiger, Pollitt, & Swauger (2005), and Swauger & Craiger (2005).

a. J. Beckett & J. Slay. Digital Forensics: Validation and verification in a dynamic work environment. In Proceedings of the 40th Hawaii International Conference on System Sciences – 2007.

b. P. Craiger, M. Pollitt and J. Swauger, Digital Evidence and law enforcement. In H Bigdoli, (Ed), *Handbook of Information Security*, New York, John Wiley and Sons, 2, pp. 739-777, 2006.

c. M. Meyers and M. Rogers. Computer Forensics: The Need for Standardization and Certification. *International Journal of Digital Evidence*, Fall 2004, Volume 3, Issue 2.

d. Institute for Security Technology Studies. (2002). *Law Enforcement Tools And Technologies for Investigating Cyber Attacks:* A National Needs Assessment Dartmouth College, Hanover, NH.

e. Lyle, J.L. (2003). NIST CFTT: Testing Disk Imaging Tools. *International Journal of Digital Evidence*, 1, 4.

f. Bryson, C., & Stevens, S. (2002). Tool Testing and Analytical Methodology In E. Casey (Ed.), *Handbook of Computer Crime Investigation: Forensic Tools and Technology. Academic Press.*

g. P. Craiger, J. Swauger, and C. Marberry. Digital forensic software tool validation In P. Kanellis (Ed) *Digital Crime and Forensic Science in Cyberspace*, Idea Group, 91-108, 2006.

h. B. Carrier. *Defining digital forensic examination and analysis tools*. DFRWS Workshop. 2002.

Methods

Note - Portions of the following were originally published in: P. Craiger, J. Swauger, & C. Marberry. Validation of Digital Forensics Tools. In P. Kanellis (Ed.), *Digital Crime and Forensic Science in Cyberspace* Idea Group

COTS (commercial off-the-shelf software) vendors often use white-box validation testing in order to affirm the soundness of their software. White-box testing (WBT) involves examination of the source code on which the application is built as well as tests comparing the performance of the software against requirements. A requirement is a specification of something that the software must do. Requirements are developed during the software design requirements phase, one of the first phases in the software engineering process. A formal examination of the source code is called a code walkthrough and has two major requirements, the primary of which is that the source code on which the application is built must be available for review. Most commercial vendors are reluctant to make source code available to external reviewers due to intellectual property concerns. Thus, code walkthroughs conducted by parties' external to a vendor are not common. From previous discussions with the large forensic software vendor we are of the understanding that they are not amenable to the release of their source code, even under the strictest non-disclosure agreements. Thus WBT is not the most attractive or feasible alternative for validation of COTS digital forensic software.

Black-Box Testing

Black-box testing (BBT) evaluates software by comparing its actual behavior against expected behavior. Unlike WBT, BBT assumes nothing about the internal structure of the application (i.e., the source code). In BBT the software serves essentially as a 'black box' and the performance of the application is evaluated against functional requirements. In a digital forensics context, BBT is performed using a tool to perform forensics tasks under various conditions, such as: different file systems, various digital artifacts, different hardware, and various software parameters (switches and settings, etc). The results of these tests across different conditions are compared against the software design requirements. If the tool performs as specified in the requirements then we have a level of confidence that the tool will work as expected under similar conditions. A positive outcome indicates we have validated the tool for the current task and conditions only; however, this confidence in the tool does not extend to conditions not covered in the test validation. BBT can be performed more quickly than WBT because it does not include a code walkthrough; however, it can still be a time consuming process as a thorough validation test may include several dozens to hundreds of test scenarios, each of which includes different combinations of hardware, test media, and software parameters. In a typical thorough validation it is crucial to exercise a tool over its full range of user selectable parameters and against a number of different data sets or test samples. Although one or two tests may produce positive results, there can always be situations where the tool will fail, situations that are unusual enough to have not been tested or addressed by the designers. Some peculiar combination of set-up parameters, operating criteria, etc., may reveal a hidden error

(i.e., software bug) that, while rarely occurring, may invalidate a tool's functionality for a particular set combination or variables.

Metrics and Errors

There are several validation metrics against which software may be tested, two of the most common of which are performance (speed) and errors. Typically speed will not be of utmost importance to the forensic practitioner. For the digital forensics practitioner the most significant metric will be whether the software performed as expected, as measured by the error rate of the tool.

In the *Daubert* decision, known or potential rates of error, and error type should be considered when evaluating a scientific technique. Two statistical error types of interest are false positive (Type I) and false negative (Type II) errors. False positive errors occur when a tool falsely identifies a positive result when none is present. For instance, using a validated reference data source, Tool X identifies a file as deleted when in actuality it is not. False negative errors occur when a tool fails to identify results that are actually there. For instance, Tool X fails to identify a file as deleted when in actuality it is. As an example, consider a forensic tool whose purpose is to scan digital media to detect .jpg graphic image files. The primary design requirement of the software, from a forensic point-of-view is to detect obfuscated jpg image files, (e.g., when a user changes a jpg extension as a means of hiding the files true type). The tool works by scanning file headers and footers (i.e., the first and last few bytes of a file that determine the files real type) and comparing the files true type with its extension. Normally, the file header/footer and extension will match. However, a simple way of hiding a file is by changing its extension.

Identifying Error Causes for Validation Testing

When a test of a software application results in test failures the most important task is to attempt to determine the cause of the test failure. In the examples above, the bit patterns of the files that were not correctly identified should be examined, and their location relative to disk sector or cluster boundaries should be reviewed. It could be that the tool is coded in such a way that it is not looking at the entire header or footer field, or has a coding error that allows it to misread the header, footer, or extension information. In addition, it may be possible that the tool has a problem accurately identifying these fields if they lay across cluster/sector boundaries, or if they lie in non-contiguous clusters. In the example used in this chapter above, further testing and analysis of the test hard disk should be performed to determine if any identifiable cause for the failures could be found. Further testing based on this and other scenarios should be performed to gather further data.

It should be noted that a limited number of failures does not necessarily completely discredit the use of the test tool software. The failure needs to be interpreted with respect to both the entirety of the test results and the nature of the failures. Depending on the manner in which the tool is to be used, a certain error rate may be acceptable if that error rate and the error types are known and taken into account in the analysis of the data recovered.

Test Scenarios

Selecting or creating test scenarios for full-blown validation testing is one of the most challenging aspects of validation testing. The set of test scenarios should consist of a number of heterogeneous examples that duplicate conditions that will be found in real

world forensic tasks. In addition to common types of data, the test scenarios must include boundary cases. Boundary cases are conditions or examples of things the tool must be capable of detecting even if they are rarely found in most situations. Recovering a 100GB file is an example of a boundary condition for the task of recovering a deleted file. If the tool correctly reports the results from real-world examples as well as boundary cases, then we can say with some authority that the software functions as expected.

A test scenario would ideally include test media containing a complete set of variables and data to thoroughly exercise the application. The advantage of running the tool against a known standard is that the results are known a priori given that the examiner knows what exists on the test media. The disadvantage is that time and effort to create the test media[i], which can be extensive, and the potential lack of knowledge about the range of variables that can exist. For example, consider the case of a test of a simple keyword extraction software package, which searches a hard disk for the presence of a keyword or keywords.

Work Description

Digital Forensic Tool Suites and Test Media

We conducted a survey of members of the International Association of Computer Investigative Specialists (IACIS), whose membership is limited to law enforcement computer forensic examiners. We received 51 responses to 31 questions regarding digital forensics tool use, frequency of encountering various operating and file systems, and common tasks conducted during the course of a computer forensic investigation. The results of the survey allowed us to tailor our testing to the most commonly

encountered file systems, operating systems, and forensic suites. In addition we selected several hardware write blockers, as they are a crucial part of a forensics investigation, allowing the examiner to read from data from media without the possibility of changing (and therefore tainting) the media.

The questions asked survey respondents about frequency of encountering various operating systems, file systems, and applications data during the course of their normal work. In addition they were asked the frequency with which they used particular digital forensics tools, which allowed us to tailor our tests to the most oft used tools in practice.

The results of our survey are included in the Appendix.

Subject Matter Experts

We selected a top-notch team of law enforcement practitioners to serve on our research team. Our subject matter experts provided us with extensive input and feedback regarding tool testing, functional requirements, as well as our reporting over the course of a two-year period during which planning and testing took place.

Sgt. Kevin Stenger (CFCE/EnCE) is head of the Computer Crimes Squad for the Orange County (Florida) Sheriff's office since 2002. He served as a member of the United States Secret Service Florida Electronic Evidence Task Force, and Supervisor of the Economic Crimes Squad (1998-2002). Sgt. Stenger has extensive training and experience (several hundred cases) involving computer crimes. He also teaches several courses in digital forensics for law enforcement and for the vendors of the tool suites tested in our research. Sgt. Stenger received a Master's of Science in Digital Forensics from the University of Central Florida in 2007.

Captain Dan Purcell, Sergeant (CFCE/EnCE) is the head of the Diversified Investigative Services Division (2011), and a former lead of the Economic & Computer Crimes Unit/Computer Forensics squad for the Seminole County (Florida) Sheriff's Office (2002-2011). He has an extensive training record for digital forensics as attested by his resume, and served as an instructor for the International Association of Computer Investigative Specialists (IACIS) from 2001-2005. Captain Purcell also serves an Agent for the United States Secret Service Electronic Crimes Task Force. Capt. Purcell received a Master's of Science in Digital Forensics from the University of Central Florida in 2007.

Test System Configuration

Digital forensic tool testing should include not only the operating system on which the tool runs, but also the *file system* of the evidentiary media. For the purposes of our testing, we developed a script that we followed to create 'evidence' for the most common operating systems (Windows, Mac OS X), crossed with the most common file systems for each operating system. (Note that we did not include Linux at this time as the results of the survey feedback suggested that Linux distributions are infrequently encountered by law enforcement examiners as of 2011) The file systems we tested include NTFS 3.5 (Windows) and HFS+ (Mac OS X). The scripts used in our testing are available for download from www.ncfs.org.

For PC-based operating systems we selected the three most commonly encountered by law enforcement examiners (based on our survey results): Windows XP, Vista, and 7. (Note that due to an error, Windows XP was inadvertently left out of the survey responses. However, additional research identified XP as the most

commonly employed operating system as of 2010). For Mac OS we chose 10.4 (Tiger), 10.5 (Leopard) and 10.6 (Snow Leopard) based upon survey results.

Hardware Testing

A key component of any forensic examination is the hardware write-blocker that allows the forensic examiner to create a duplicate of the original evidence without changing it. We identified the following write blockers for testing:

a. ULTRABLOCK series from Digital Intelligence (www.digitalintelligence.com)
b. Wiebetech's (www.wiebetech.com) Ultradock and Forensic Drivedock series
c. Tableau's (www.tableau.com) SATA/IDE Forensic Bridge

Each write blocker was subjected to multiple tests. Prior to duplication, we created a forensic (SHA-1) hash of the media to be duplicated using the Linux md5sum utility. We then connected the write blocker and the media, and subsequently create a duplicate using the duplication functions included in the forensic suites. We subsequently hashed the resulting duplicate image, and the original media, to determine whether a) the forensic duplication tool functioned properly, and b) whether the write blocker worked appropriately.

Testing Methodology

For the Windows-based tests we used the Scientific Working Group on Digital Evidence's '2009-01-15 SWGDE Recommendations for Validation Testing Version v1.1' testing procedures, which was specifically written for the validation of digital forensic tools. For the Mac-based testing we used an adaptation of the methodology developed by the National Institute for Standards and Technology (NIST) in 2001 for computer

forensic tool testing (CFTT) guidelines was used as the basis for the testing methodology.

Both the SWGDE and NIST approaches are based on well-organized methodologies for conformance testing and quality testing such as ISO/IEC Guide 2, *Standardization and Related Activities – General Vocabulary*, and the ISO/IEC 17025, *General Requirements for the Competence of Testing and Calibration Laboratories* (NIST, 2001). Note that the SWGDE and NIST validation testing guidelines are similar, involving the following steps for testing a computer forensic tool:

a. Establish categories of forensic requirements
b. Identify requirements for a specific category
c. Develop test assertions based on requirements
d. Develop test code for assertions
e. Identify relevant test cases
f. Develop testing procedures and method
g. Report test results

Validation Tests: Windows-based Forensic Tool Suites

Based on feedback from the IACIS survey we identified the most commonly used forensic tool suites, file systems, operating systems, and forensic tasks. In regards to the PC-based testing, the validation test encompassed examining the two most widely used digital forensics software tools (Encase 6.16.2 and FTK 1.81.6) by triangulating their ability to perform common forensic analysis tasks. The requirements (or forensic tasks) chosen for the study are:

- Identifying and recovering digital evidence related to web browsing (history, cookies, searches)
- Identifying and recovering of email correspondence (Outlook Client, Web Based)

- Identifying and recovering instant messages (Facebook only)
- Identifying and recovering encrypted files
- Identifying and recovering compressed files
- Identifying and recovering modified, access, and created times
- Identifying and recovering deleted files
- Identifying and recovering ASCII/UNICODE search strings
- Identifying and recovering of graphic files
- Identifying and recovering Window's registry information

Note that the purported functionality for each requirement was identified from the respective tool's user guide or manual (corresponding to the exact version of the tool used in testing) where available. The exact functionality tested is described in detail in each report.

The groupings of forensic functions contain the forensic requirement categories that are determined by expert examiners. A test assertion is a statement of behavior, action, or condition that is testable or measurable (NIST, 2001). The test cases are then created based on the test assertions. Procedures and methods are developed for each test case. The test cases are implemented; then the results recorded and evaluated.

Reporting

For the Windows-based forensic tool suites, an individual reporting document was created for each function tested, version of the operating system, and tool suite. Thus, for a single function, testing keyword search for example, six tests were run, and six reports were generated documenting the results: (Windows XP, Vista, 7) x (Encase, FTK).

During the initial discussions of our research team we came to the conclusion that it was important that the documentation of the steps used in our tests be as thorough and transparent as possible for several reasons. First is to clarify to the reader the specific steps used in our testing so that an examiner could replicate the same steps. Second is that, based on anecdotal evidence, forensic examiners (at least as of 2010) do not perform thorough tests of the tool suites prior to performing examinations. This is, more than likely, not due to a lack of desire or laziness on the part of the examiner, but rather a lack of training or understanding as to how to properly perform the tests.

Based on these two factors, we developed very detailed descriptions of procedures used in the tests in textual form, and also screen captures of each step. The latter was added as a means of clarifying the exact steps employed in the all portions of the research, including media preparation, evidence creation, and testing. As such, the reports may be used by forensic examiners or other interested parties as educational or training materials.

Results

Our researcher team (both at UCF and Purdue) completed several hundred validation tests. Each test has an associated report that corresponds to the test plan reporting as outlined in either the SWGDE validation guidelines (for the Windows-based tools), or the NIST validation guidelines (for the Mac-based tools). The total size of the reports is several gigabytes in size, which exceeds the limits of the upload capability of the NIJ site. As such, we have included all test reports, evidence creation scripts, installation information, hardware testing information, etc., on a DVD that was mailed to

NIJ. The information will also be available on NCFS's website (www.ncfs.org), where individual reports may be downloaded.

Limitations of the Study

The purpose of our study was to conduct tests to determine the reliability of digital forensic tools suites across multiple experimental conditions. There are two terms that are often used to describe these types of tests: *verification* and *validation*. SWGDE defined these terms in their SWGDE and SWGIT Digital & Multimedia Evidence Glossary (11/1/2007) as follows:

Validation Testing: An evaluation to determine if a tool, technique or procedure functions and as intended.

Verification: The process of confirming the accuracy of an item to its original; confirmation that a tool, technique or procedure performs as expected.

By these definitions, the two are closely aligned.

The extrapolation of the results of software validation is inherently limited for several reasons. First and foremost is that new versions of software may 'fix' previous detected faults ('bugs'), or even introduce new faults. In addition, these bugs may interact with the testing media (other software, hardware, media under examination, etc.), so that the bug may be apparent only under certain circumstances. *Thus, it is crucial that examiners test their own combination of software version and hardware to identify any discrepancies between expected and actual results prior to conducting a forensic examination, or relying upon the results.* Accordingly, the results of our validation tests may be extrapolated to only the versions of the software versions used in our study.

Conclusions

The general consensus of the tests is that the forensic suites work as expected. There are some exceptions, and the discrepancies between expected and actual results may be found by referring to the individual validation reports, which are available via download from www.ncfs.org in PDF format.

There are several issues we encountered for which we may judgment calls. For instance, we referred to the software manuals to identify the software capabilities. In some instances, we found the manuals sometimes lacking in detailed description of the capabilities and limitations of the tools. Therefore, we sometimes had to use our own expertise and judgment as to the expectation of what the tool's capabilities are. Major software vendors often provide software training for several thousands of dollars that provides 'hands-on' training, as well as additional information regarding capabilities and tool use. One item on our wish list for forensic examiners is more detailed user manuals, much like common office software, which would provide the less experienced forensic examiner with better information on the use and capabilities of the tools.

It is important to note that, from our own experiences and anecdotal evidence, many forensic examiners, particularly at the local level, are 'cops' first, and 'examiners' second. That is, at the local level you are more likely to find law enforcement officers that receive training to be forensic examiners, rather than trained and educated forensic examiners hired specifically for that purpose. As such, 'examiners' are less likely to have training in software tool testing. This is one of the driving motivations for the detail included in our testing reports. An examiner has a detailed example (hundreds of them) that can be used as a template for testing his/her own tools.

Digital forensics is a relatively new field, in contrast to physical and biological forensics; this is likely to change in the future as the field matures. We expect that software vendors will keep abreast of these changes, as they have in the past.

Implications for criminal justice policy and practice in the United States

Our project's primary goal is to provide scientific validation to commonly used digital evidence tools. The requirement for formal testing and validation has been echoed by the recent Academies of Science report, as well as being a crucial component of any *Daubert* challenge to scientifically derived evidence. The resultant testing methodology and metrics related to accuracy, validation and error rates will provide the first truly non-vendor data that is both up-to-date and scholarly. The fact that we are including LEO in the functional requirements phases and in oversight provides a voice to the community most directly impacted by this research. *Tool testing must be an ongoing activity to accompany changes in technology*. For instance, since this project started there are new versions of both Windows (8) and Mac OS X (10.7), and there have been updates to the forensic tool suites used in this research. As such, tool testing will always remain one step behind technology.

We believe project marks a significant shift in digital forensic science – moving away from direct reliance on vendors for disclosures related to accuracy, validation and error rates of their tools (Meyers & Rogers, 2004). This is an extremely important step toward maturing the field of digital evidence, as well as ensuring that the domain does not get relegated to the category of pseudo or junk science. The ability to have independent validated tools is also critical for those agencies seeking ASCLD certification for their

forensic labs that process digital evidence. In many jurisdictions this certification is or will be mandatory.

APPENDIX

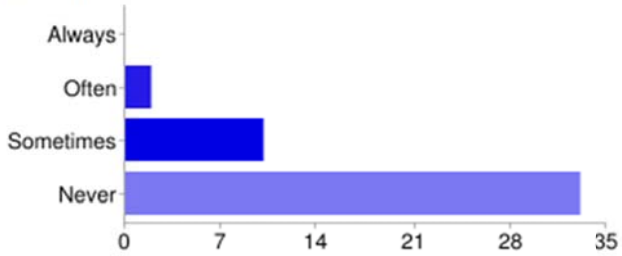
2009 IACIS Survey

We conducted a survey of members of the International Association of Computer Investigative Specialists (IACIS), whose membership is limited to law enforcement computer forensic examiners. We received 45 responses to 31 questions regarding digital forensics tool use, frequency of encountering various operating and file systems, common tasks done during the course of a computer forensic investigation. The results of the survey allowed us to tailor our validation testing to the most commonly encountered file systems, operating systems, and forensic suites. In addition we selected several hardware write blockers, as they are a crucial part of a forensics investigation, allowing the examiner to read from data from media without the possibility of changing (and therefore tainting) the media.

The results of our survey are below.

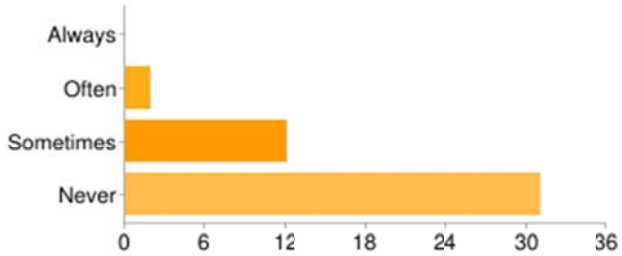
NTFS File System

How frequently have you encountered the following NFTS file system versions in your work? - NTFS 1.1 (NT 3.5)



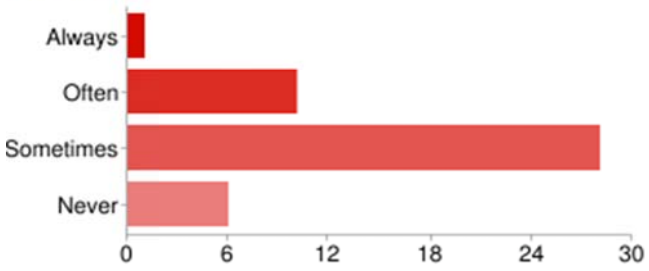
Always	**0**	0%
Often	**2**	4%
Sometimes	**10**	22%
Never	**33**	73%

How frequently have you encountered the following NFTS file system versions in your work? - NTFS 1.2 (NT 3.51)



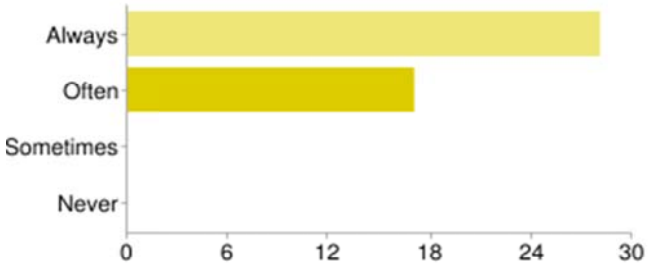
Always	**0**	0%
Often	**2**	4%
Sometimes	**12**	27%
Never	**31**	69%

How frequently have you encountered the following NFTS file system versions in your work? - NTFS 3.0 (Win 2000)



Always	**1**	2%
Often	**10**	22%
Sometimes	**28**	62%
Never	**6**	13%

How frequently have you encountered the following NFTS file system versions in your work? - NTFS 3.1 (XP, Vista, 7, Server)



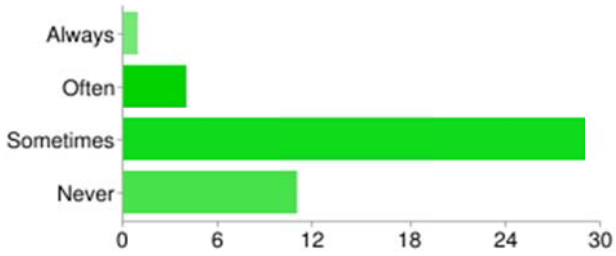
Always	**28**	62%
Often	**17**	38%
Sometimes	**0**	0%
Never	**0**	0%

Comments regarding NTFS file system

NTFS 3.0 and 3.1 only in the past 3-4 years

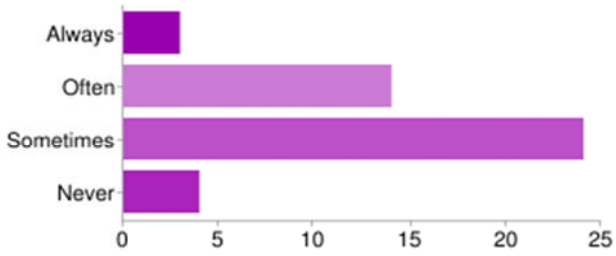
FAT File System

How frequently have you encountered the following FAT file systems in your work? - FAT12



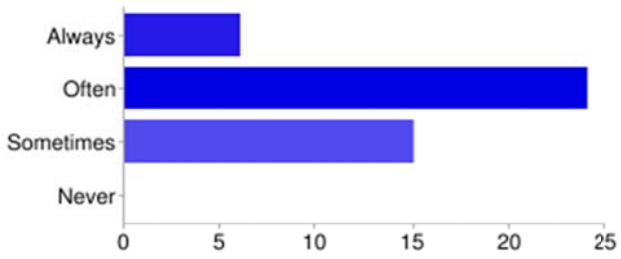
Always	1	2%
Often	4	9%
Sometimes	29	64%
Never	11	24%

How frequently have you encountered the following FAT file systems in your work? - FAT16



Always	3	7%
Often	14	31%
Sometimes	24	53%
Never	4	9%

How frequently have you encountered the following FAT file systems in your work? - FAT32



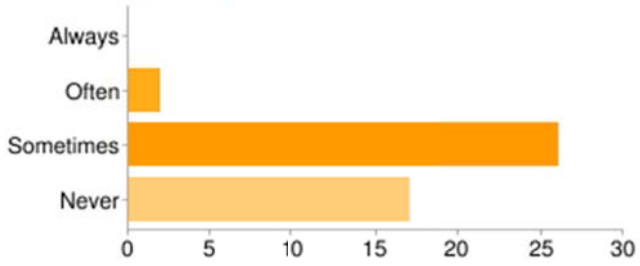
Always	**6**	13%
Often	**24**	53%
Sometimes	**15**	33%
Never	**0**	0%

Comments regarding FAT file system

I have not seen FAT12 in years and my most recent cases, last 3-4 years anything FAT has been very rare. Generally we see FAT 16 and 32 on media cards and sometimes on external hds Becoming ever more rare, with UBS External drives being the exception. I have also examined pieces of media which was formatted exFAT which is not listed above Mostly on flash memory (cards in cameras) or USB drives. FAT12 is extremely rarely (nearly never)

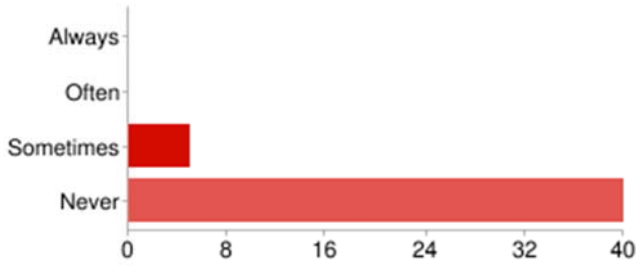
Linux/UNIX File Systems

How frequently have you encountered the following Linux/UNIX file systems in your work? - EXT (2,3,4)



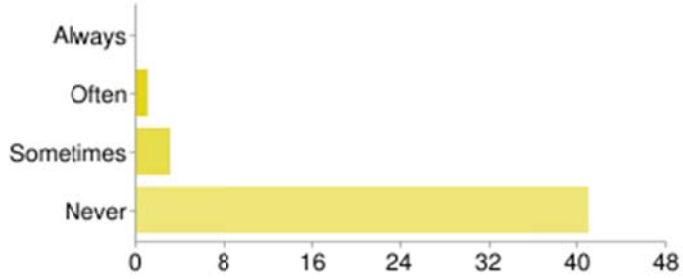
Always	**0**	0%
Often	**2**	4%
Sometimes	**26**	58%
Never	**17**	38%

How frequently have you encountered the following Linux/UNIX file systems in your work? - Reiser



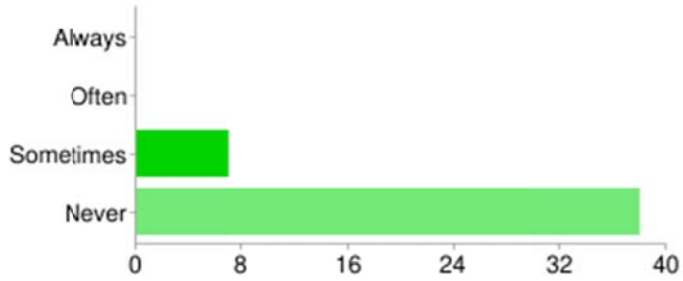
Always	**0**	0%
Often	**0**	0%
Sometimes	**5**	11%
Never	**40**	89%

How frequently have you encountered the following Linux/UNIX file systems in your work? - JFS



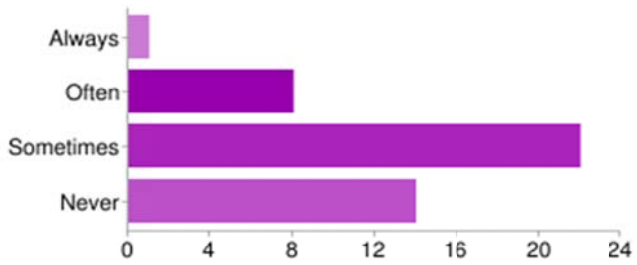
Always	**0**	0%
Often	**1**	2%
Sometimes	**3**	7%
Never	**41**	91%

How frequently have you encountered the following Linux/UNIX file systems in your work? - XFS



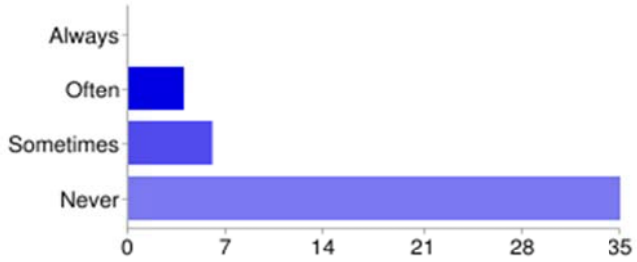
Always	**0**	0%
Often	**0**	0%
Sometimes	**7**	16%
Never	**38**	84%

How frequently have you encountered the following Linux/UNIX file systems in your work? - HFS/HFS+



Always	**1**	2%
Often	**8**	18%
Sometimes	**22**	49%
Never	**14**	31%

How frequently have you encountered the following Linux/UNIX file systems in your work? - UFS



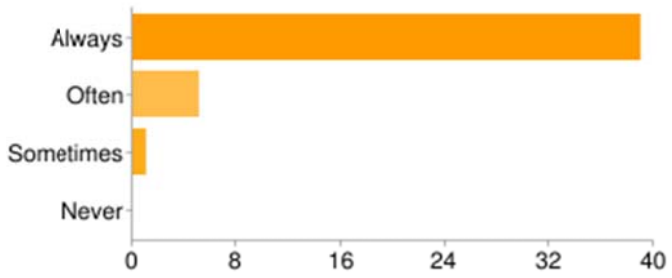
Always	**0**	0%
Often	**4**	9%
Sometimes	**6**	13%
Never	**35**	78%

Comments regarding Linux/UNIX file systems

I have never had a case where the subject OS was Linux. I did have one case with an AS/400 and it was AIX

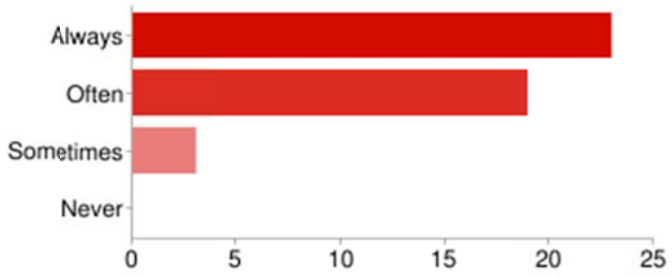
Reiser and XFS are extremely rarely (nearly never)

Rate the frequency of use of the following forensic tool functions in your work - Hashing (MD5, SHA-1, etc.)



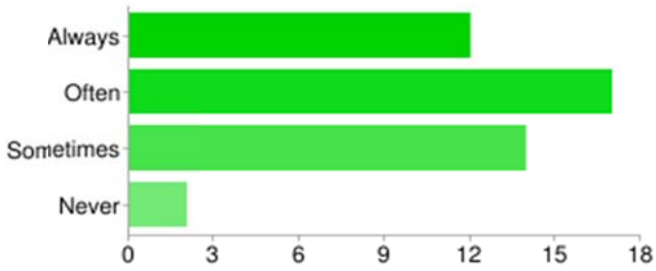
Always	**39**	87%
Often	**5**	11%
Sometimes	**1**	2%
Never	**0**	0%

Rate the frequency of use of the following forensic tool functions in your work - Recovering web browser activity



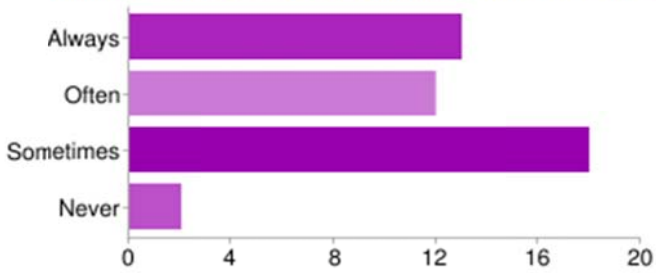
Always	**23**	51%
Often	**19**	42%
Sometimes	**3**	7%
Never	**0**	0%

Rate the frequency of use of the following forensic tool functions in your work - Recovering IM or chat activity



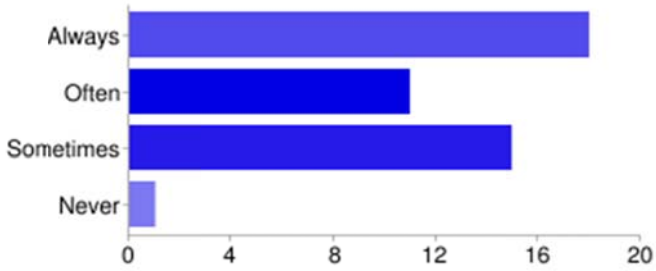
Always	**12**	27%
Often	**17**	38%
Sometimes	**14**	31%
Never	**2**	4%

Rate the frequency of use of the following forensic tool functions in your work - Identifying encrypted files

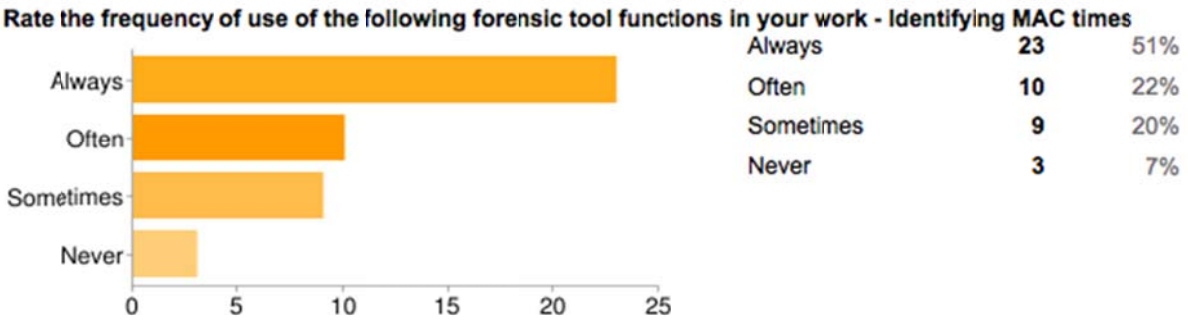


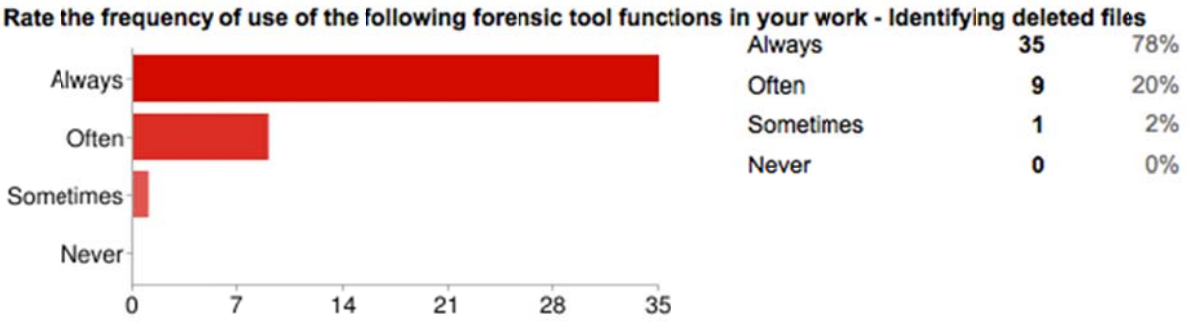
Always	**13**	29%
Often	**12**	27%
Sometimes	**18**	40%
Never	**2**	4%

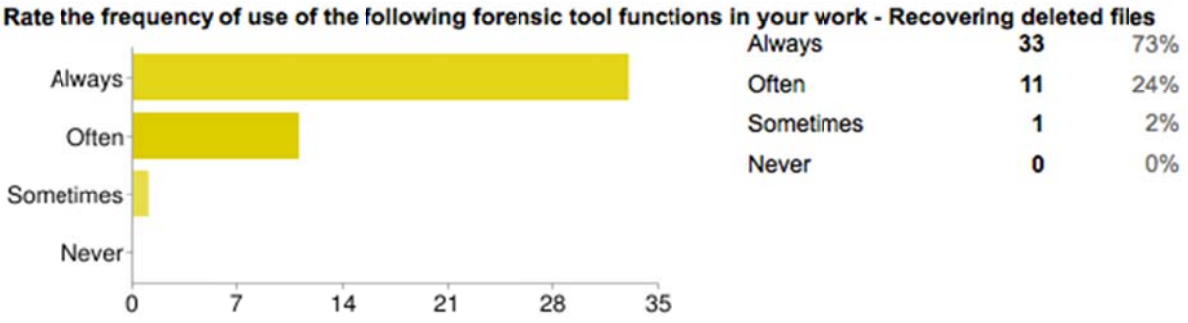
Rate the frequency of use of the following forensic tool functions in your work - Identifying compressed files



Always	**18**	40%
Often	**11**	24%
Sometimes	**15**	33%
Never	**1**	2%

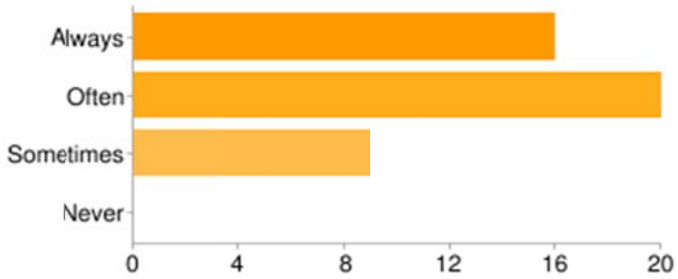






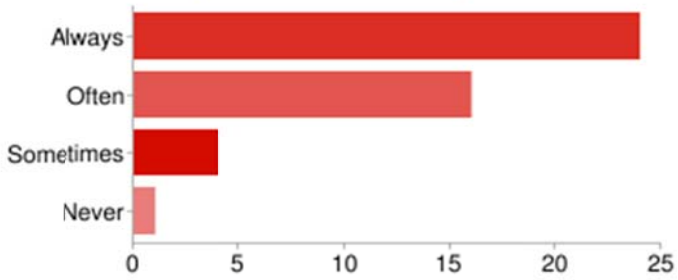
a. Résumés of key personnel (see the following pages for documentation)

Rate the frequency of use of the following forensic tool functions in your work - UNICODE search



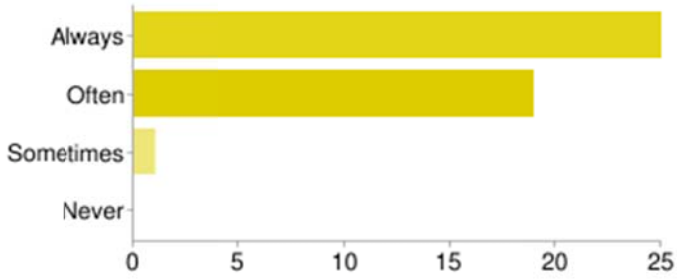
Always	**16**	36%
Often	**20**	44%
Sometimes	**9**	20%
Never	**0**	0%

Rate the frequency of use of the following forensic tool functions in your work - Graphic/picture search



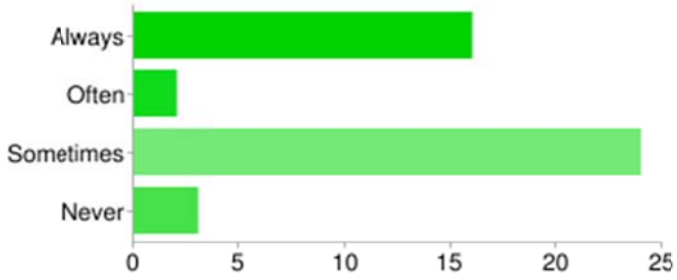
Always	**24**	53%
Often	**16**	36%
Sometimes	**4**	9%
Never	**1**	2%

Rate the frequency of use of the following forensic tool functions in your work - Windows registry search



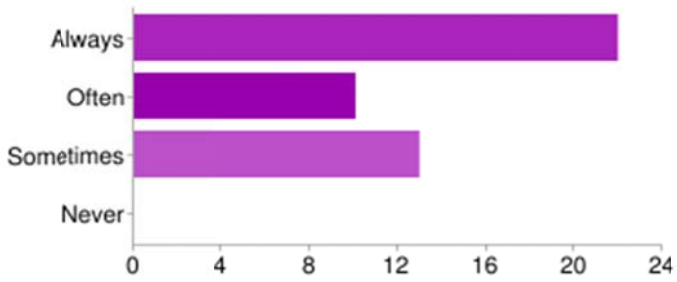
Always	**25**	56%
Often	**19**	42%
Sometimes	**1**	2%
Never	**0**	0%

Rate the frequency of use of the following forensic tool functions in your work - Identifying hidden partitions



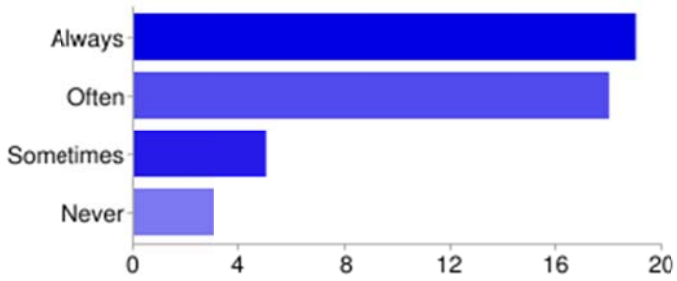
Always	**16**	36%
Often	**2**	4%
Sometimes	**24**	53%
Never	**3**	7%

Rate the frequency of use of the following forensic tool functions in your work - Recovering slack space



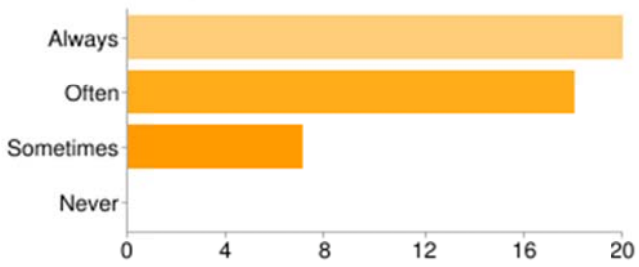
Always	**22**	49%
Often	**10**	22%
Sometimes	**13**	29%
Never	**0**	0%

Rate the frequency of use of the following forensic tool functions in your work - ASCII search



Always	**19**	42%
Often	**18**	40%
Sometimes	**5**	11%
Never	**3**	7%

Rate the frequency of use of the following forensic tool functions in your work - View spreadsheet file



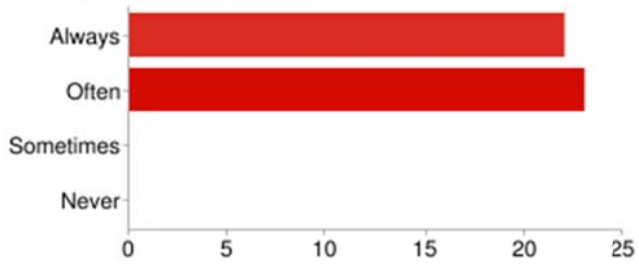
Always	**20**	44%
Often	**18**	40%
Sometimes	**7**	16%
Never	**0**	0%

Comments on use of forensic functions

-Hexadecimal View of files is an important forensic function (inclusive data interpreter; e.g. for timestamps)

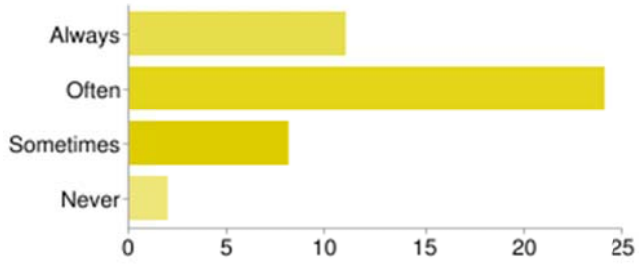
Web-Browsing Activity

In the past how frequently have you had to recover trace evidence associated with the following web browsers? - Internet Explorer



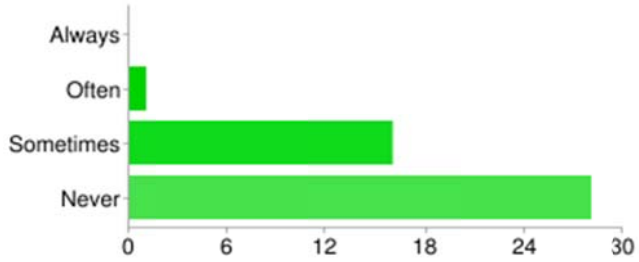
Always	**22**	49%
Often	**23**	51%
Sometimes	**0**	0%
Never	**0**	0%

In the past how frequently have you had to recover trace evidence associated with the following web browsers? - Firefox



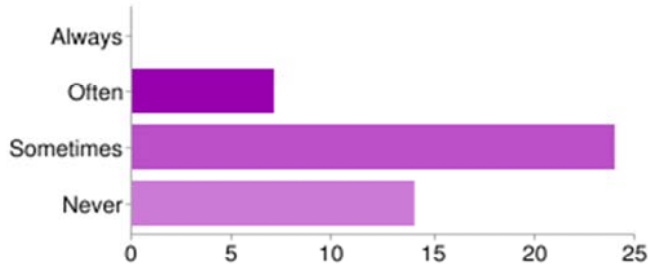
Always	**11**	24%
Often	**24**	53%
Sometimes	**8**	18%
Never	**2**	4%

In the past how frequently have you had to recover trace evidence associated with the following web browsers? - Opera



Always	**0**	0%
Often	**1**	2%
Sometimes	**16**	36%
Never	**28**	62%

In the past how frequently have you had to recover trace evidence associated with the following web browsers? - Safari



Always	**0**	0%
Often	**7**	16%
Sometimes	**24**	53%
Never	**14**	31%

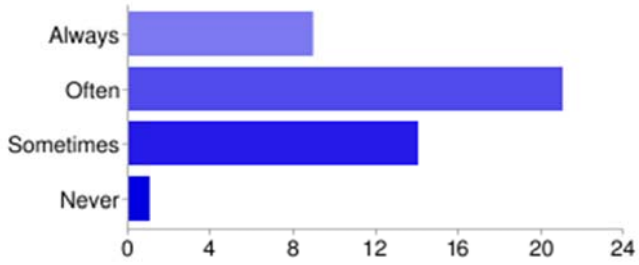
Comments on web browsers

Chrome is also becoming more popular

Starting to see an increase of Google Chrome usage

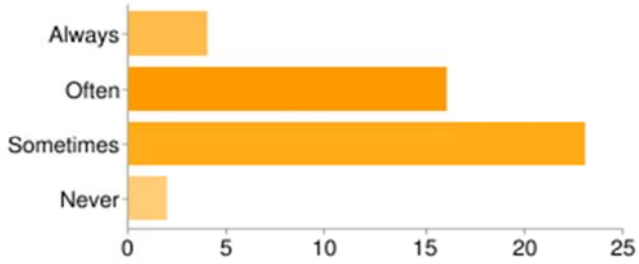
Email applications

In the past how frequently have you had to recover trace evidence associated with the following email applications? - Outlook



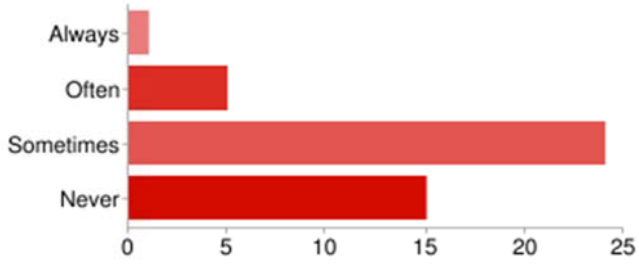
Always	9	20%
Often	21	47%
Sometimes	14	31%
Never	1	2%

In the past how frequently have you had to recover trace evidence associated with the following email applications? - Outlook Express



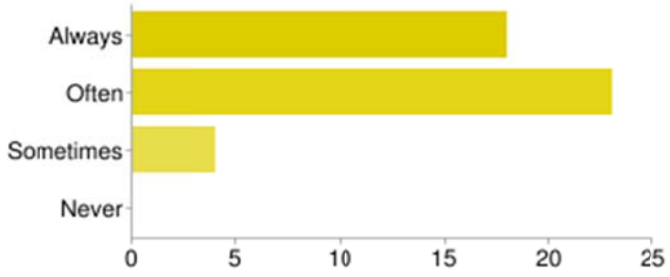
Always	4	9%
Often	16	36%
Sometimes	23	51%
Never	2	4%

In the past how frequently have you had to recover trace evidence associated with the following email applications? - Thunderbird



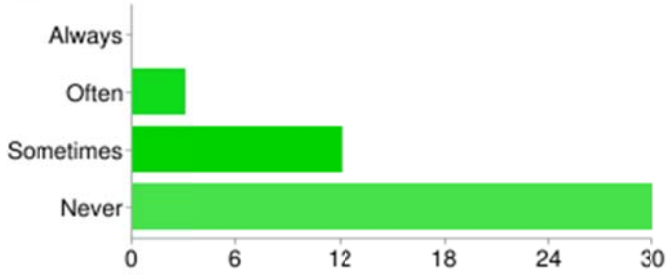
Always	1	2%
Often	5	11%
Sometimes	24	53%
Never	15	33%

In the past how frequently have you had to recover trace evidence associated with the following email applications? - Web mail (Yahoo!, Gmail, LiveMail, etc.)



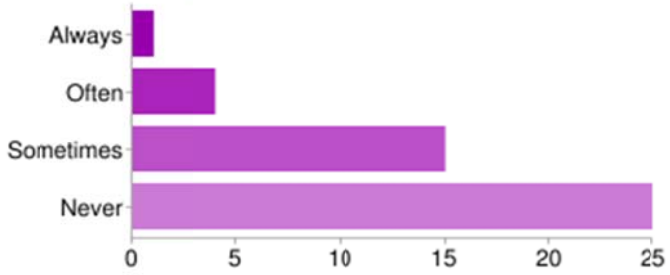
Always	**18**	40%
Often	**23**	51%
Sometimes	**4**	9%
Never	**0**	0%

In the past how frequently have you had to recover trace evidence associated with the following email applications? - Entourage (OS X)



Always	**0**	0%
Often	**3**	7%
Sometimes	**12**	27%
Never	**30**	67%

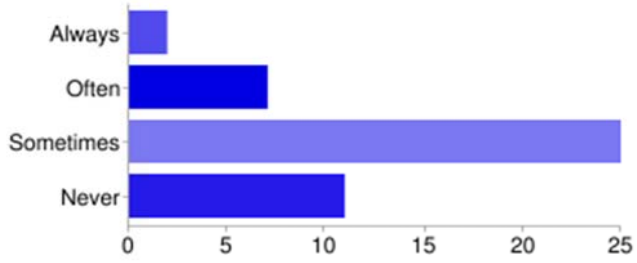
In the past how frequently have you had to recover trace evidence associated with the following email applications? - Apple Mail (OS X)



Always	**1**	2%
Often	**4**	9%
Sometimes	**15**	33%
Never	**25**	56%

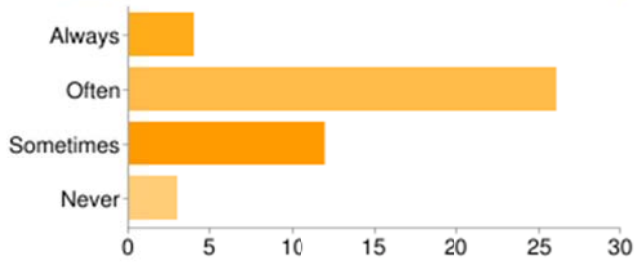
Instant Messaging/Chat Applications

In the past how frequently have you had to recover trace evidence associated with the following instance messaging/chat protocols? · AOL Instant Messenger



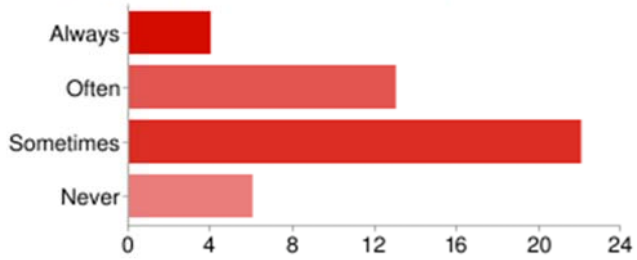
Always	**2**	4%
Often	**7**	16%
Sometimes	**25**	56%
Never	**11**	24%

In the past how frequently have you had to recover trace evidence associated with the following instance messaging/chat protocols? · Web-based (Facebook, My Space, Google, Yahoo!, etc.)



Always	**4**	9%
Often	**26**	58%
Sometimes	**12**	27%
Never	**3**	7%

In the past how frequently have you had to recover trace evidence associated with the following instance messaging/chat protocols? · MSN Messenger



Always	**4**	9%
Often	**13**	29%
Sometimes	**22**	49%
Never	**6**	13%

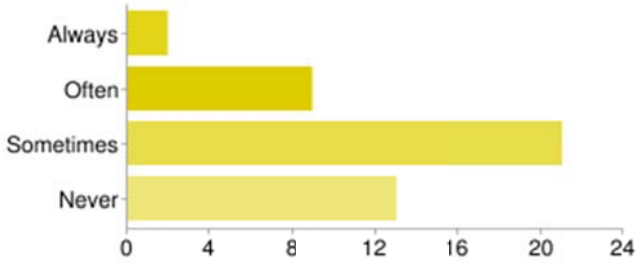
Comments on IM/chat applications

also ICQ, Yahoo!

Messenger and Skype are very important messengers

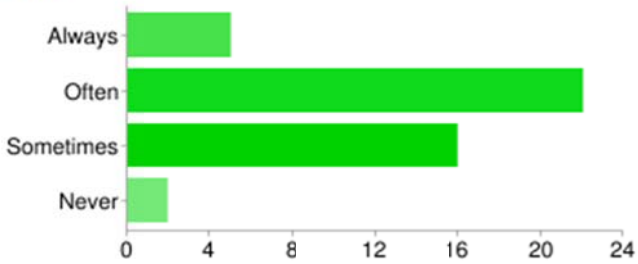
PC-based Operating Systems

In the past how frequently have you encountered the following PC-based operating systems? - Windows 7



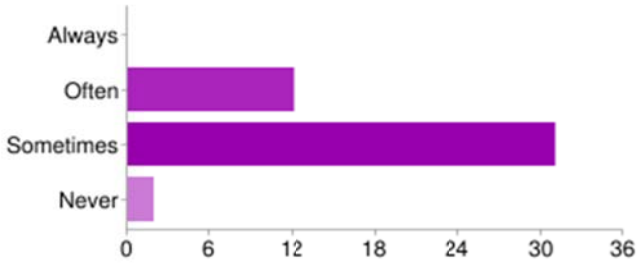
Always	**2**	4%
Often	**9**	20%
Sometimes	**21**	47%
Never	**13**	29%

In the past how frequently have you encountered the following PC-based operating systems? - Windows Vista



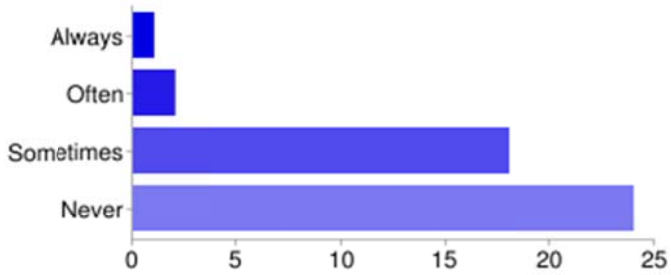
Always	**5**	11%
Often	**22**	49%
Sometimes	**16**	36%
Never	**2**	4%

In the past how frequently have you encountered the following PC-based operating systems? - Windows 2000



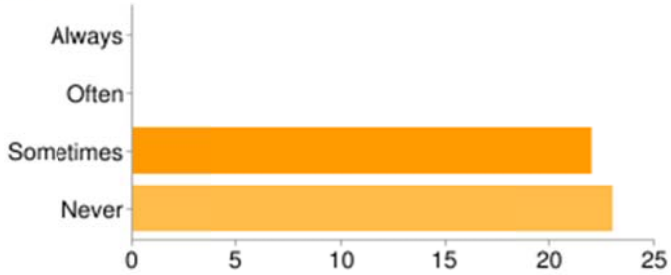
Always	**0**	0%
Often	**12**	27%
Sometimes	**31**	69%
Never	**2**	4%

In the past how frequently have you encountered the following PC-based operating systems? - Windows Server



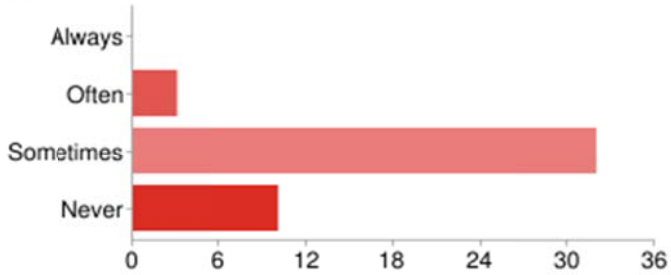
Always	**1**	2%
Often	**2**	4%
Sometimes	**18**	40%
Never	**24**	53%

In the past how frequently have you encountered the following PC-based operating systems? - Windows ME



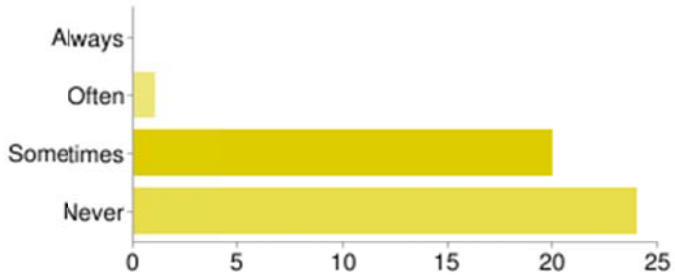
Always	**0**	0%
Often	**0**	0%
Sometimes	**22**	49%
Never	**23**	51%

In the past how frequently have you encountered the following PC-based operating systems? - Windows 98/98SE



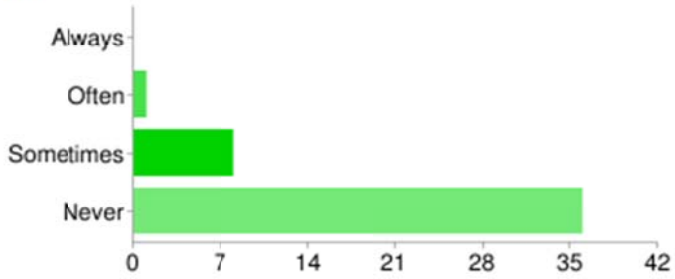
Always	**0**	0%
Often	**3**	7%
Sometimes	**32**	71%
Never	**10**	22%

In the past how frequently have you encountered the following PC-based operating systems? - Windows 95



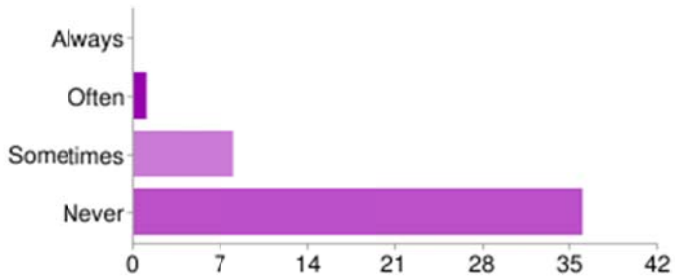
Always	**0**	0%
Often	**1**	2%
Sometimes	**20**	44%
Never	**24**	53%

In the past how frequently have you encountered the following PC-based operating systems? - Windows 3.1



Always	**0**	0%
Often	**1**	2%
Sometimes	**8**	18%
Never	**36**	80%

In the past how frequently have you encountered the following PC-based operating systems? - DOS (any version)



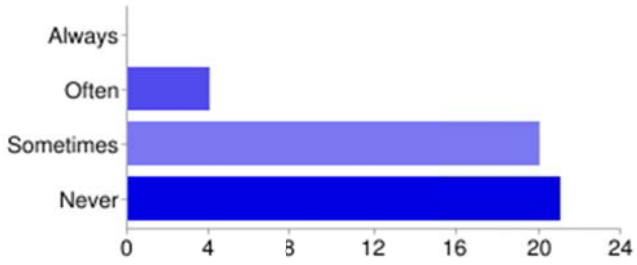
Always	**0**	0%
Often	**1**	2%
Sometimes	**8**	18%
Never	**36**	80%

Comments on PC-based operating systems

95% has been either Windows Vista or XP. Predominantly XP based systems The most common Operating System encountered is still XP with Vista a close second. These should be quickly overtaken by Windows 7. Almost everything we are seeing now is XP and Vista. I expect to see more Windows 7 soon. XP still dominates this realm and will so for the next year or so... about 1 of 100 is Windows 9x

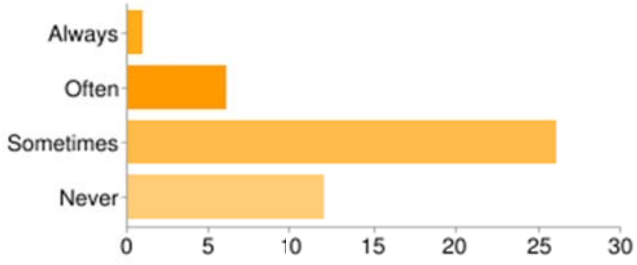
Macintosh-based Operating Systems

In the past how frequently have you encountered the following Macintosh-based operating systems? - OS X 10.6 (Snow Leopard)



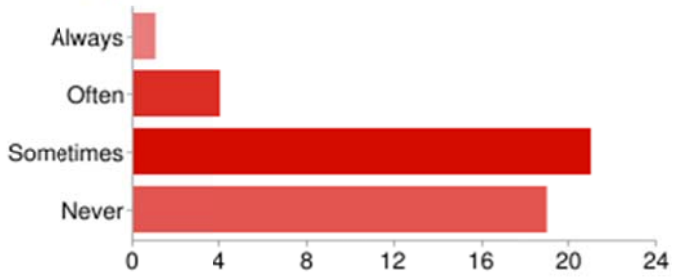
Always	**0**	0%
Often	**4**	9%
Sometimes	**20**	44%
Never	**21**	47%

In the past how frequently have you encountered the following Macintosh-based operating systems? - OS X 10.5 (Leopard)



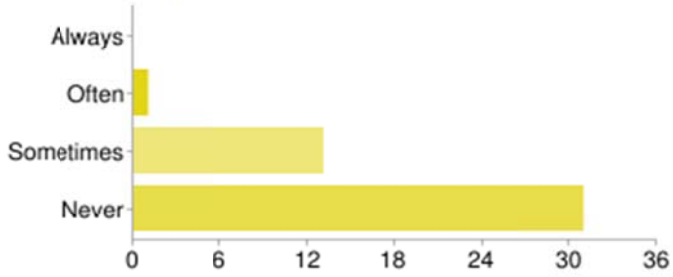
Always	**1**	2%
Often	**6**	13%
Sometimes	**26**	58%
Never	**12**	27%

In the past how frequently have you encountered the following Macintosh-based operating systems? - OS X 10.4 (Tiger)



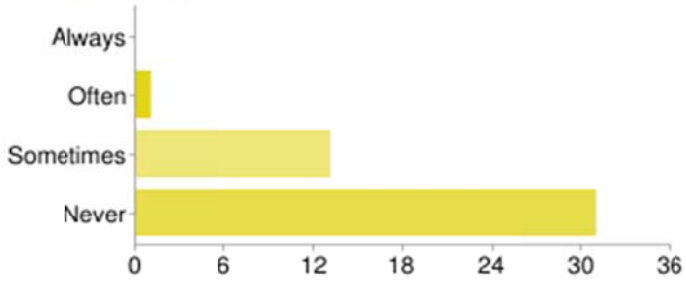
Always	**1**	2%
Often	**4**	9%
Sometimes	**21**	47%
Never	**19**	42%

In the past how frequently have you encountered the following Macintosh-based operating systems? - OS X 10.3 (Panther)



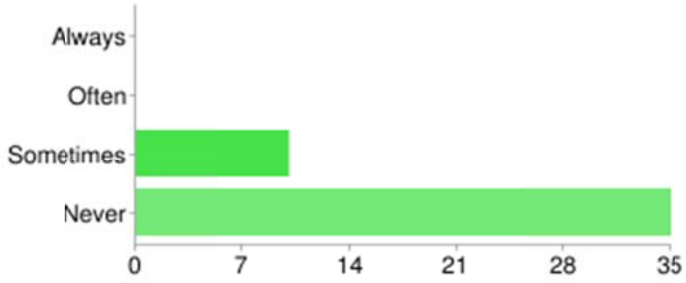
Always	**0**	0%
Often	**1**	2%
Sometimes	**13**	29%
Never	**31**	69%

In the past how frequently have you encountered the following Macintosh-based operating systems? - OS X 10.3 (Panther)



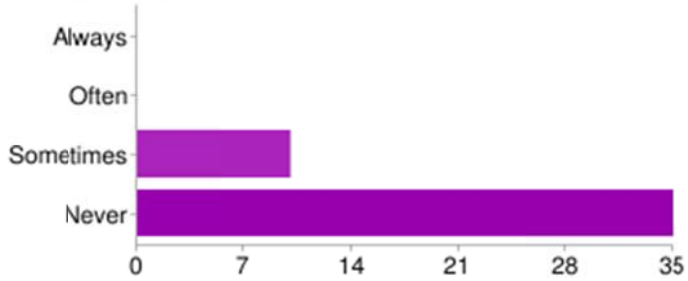
Always	**0**	0%
Often	**1**	2%
Sometimes	**13**	29%
Never	**31**	69%

In the past how frequently have you encountered the following Macintosh-based operating systems? - OS X 10.2 (Jaguar)



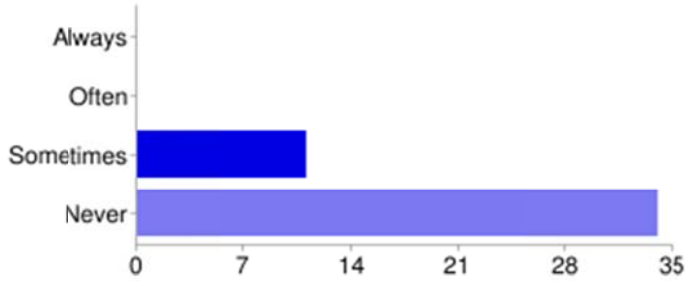
Always	**0**	0%
Often	**0**	0%
Sometimes	**10**	22%
Never	**35**	78%

In the past how frequently have you encountered the following Macintosh-based operating systems? - OS X 10.1 (Puma)



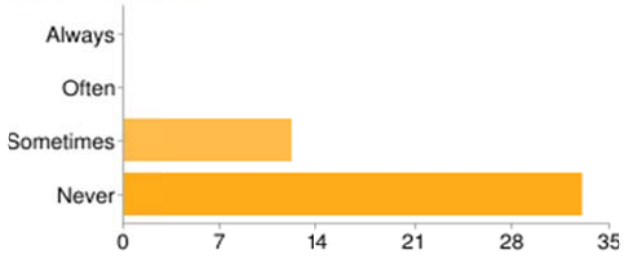
Always	**0**	0%
Often	**0**	0%
Sometimes	**10**	22%
Never	**35**	78%

In the past how frequently have you encountered the following Macintosh-based operating systems? - OS X 10.0 (Cheetah)



Always	**0**	0%
Often	**0**	0%
Sometimes	**11**	24%
Never	**34**	76%

In the past how frequently have you encountered the following Macintosh-based operating systems? - Mac OS 9 or earlier



Always	**0**	0%
Often	**0**	0%
Sometimes	**12**	27%
Never	**33**	73%

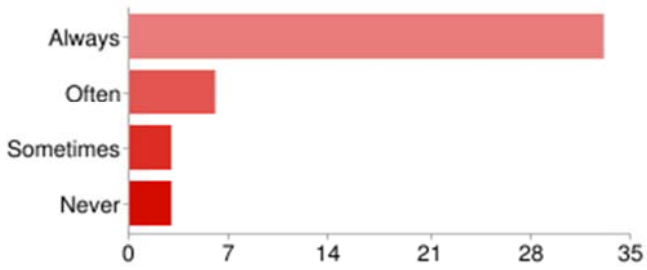
Comments on Macintosh-based operating systems

I have never had a case that involved a Mac I am not certified as a Mac examiner with my agency. Apple computers are rarely (about 1 of 50)

Forensic Tools Used

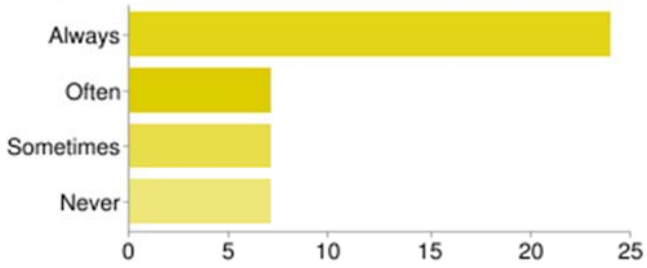
Please identify the forensic tools you use along with the frequency of use

In the past how frequently have you used the following forensic tools - Encase (any version)



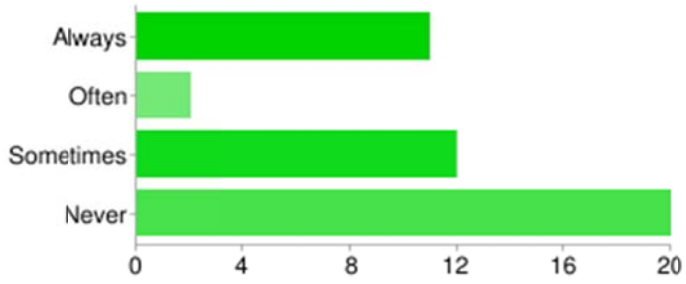
Always	**33**	73%
Often	**6**	13%
Sometimes	**3**	7%
Never	**3**	7%

In the past how frequently have you used the following forensic tools - Forensic Tool Kit versions 1.0-1.8



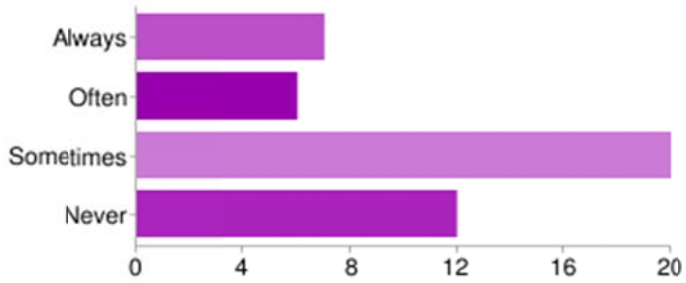
Always	**24**	53%
Often	**7**	16%
Sometimes	**7**	16%
Never	**7**	16%

In the past how frequently have you used the following forensic tools - Forensic Tool Kit versions 2 or 3



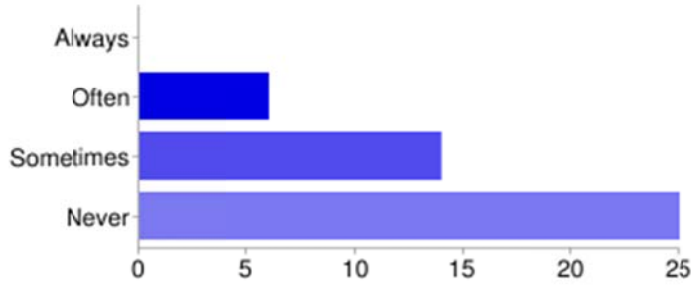
Always	**11**	24%
Often	**2**	4%
Sometimes	**12**	27%
Never	**20**	44%

In the past how frequently have you used the following forensic tools - X-Ways or Winhex



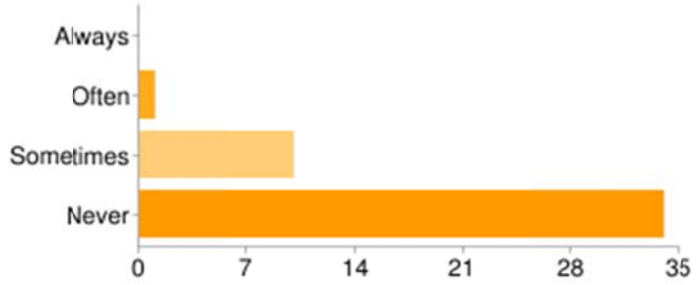
Always	**7**	16%
Often	**6**	13%
Sometimes	**20**	44%
Never	**12**	27%

In the past how frequently have you used the following forensic tools - Paraben



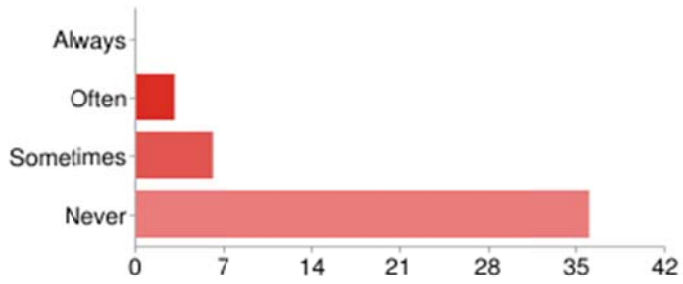
Always	**0**	0%
Often	**6**	13%
Sometimes	**14**	31%
Never	**25**	56%

In the past how frequently have you used the following forensic tools - BlackBag Forensic Suite (Macintosh)



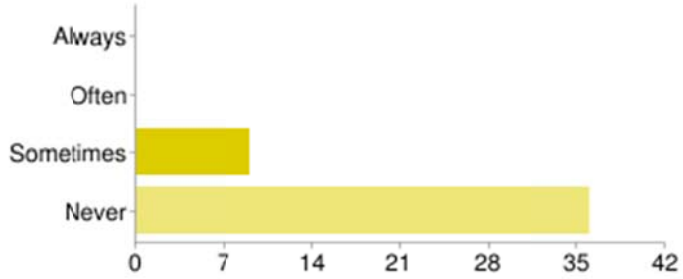
Always	**0**	0%
Often	**1**	2%
Sometimes	**10**	22%
Never	**34**	76%

In the past how frequently have you used the following forensic tools - MacForensicsLab (Macintosh)



Always	**0**	0%
Often	**3**	7%
Sometimes	**6**	13%
Never	**36**	80%

In the past how frequently have you used the following forensic tools - Sleuthkit (TSK)



Always	**0**	0%
Often	**0**	0%
Sometimes	**9**	20%
Never	**36**	80%

Comments regarding forensic tools used (including list of other tools)

Darren Freestone's Registry Browser Craig Wilsons Netanalysis MobileEdit for phones P2P Commander NetAnalysis ProDiscover Internet Evidence Finder, CacheBack, Net Analysis, VMWare Workstation, Metadata Assistant, Helix tools SimpleCarver Cache Back, Gargoyale, P2P Marshal, Mac Marshal, CacheBack Mac Marshal Gargoyle Intern Evidence Finder FTK is a piece of crap Infinadine DVD Inspector. Often Digital Detective (Craig Wilson) Net Analysis. Always Quick View Plus Often. Quicklook Often. USBDeview Always. Internet &Email Analysis Sometimes. Internet Evidence Finder v3.2.0 JAD Software Sometimes. Cach ...

Thank you for your time and effort! (Don't forget to hit the 'SUBMIT' button!)

Number of daily responses