A Lessons Learned Repository for Computer Forensics

Warren Harrison
Portland State University
David Aucsmith
Intel Corporation
George Heuston
Hillsboro Police Department
Sarah Mocas
Portland State University
Mark Morrissey
Portland State University
Steve Russelle
Portland Police Bureau

Abstract

The Law Enforcement community possesses a large, but informal, community memory with respect to digital forensics. Large, because the experiences of every forensics technician and investigator contribute to the whole. Informal because there is seldom an explicit mechanism for disseminating this wisdom except "over the water cooler". As a consequence, the same problems and mistakes continue to resurface and the same solutions are re-invented. In order to better exploit this informal collection of wisdom, the key points of each experience can be placed into a Repository for later dissemination. We describe a web-based Lessons Learned Repository (LLR) that facilitates contribution of Lessons, and their subsequent retrieval.

Introduction

Digital devices are becoming increasingly commonplace in society, and can be found almost everywhere. Such devices include, but are not limited to:

- PDAs
- Cell Phones
- Computers
- USB Flash Cards
- FAX Machines

While the proliferation of new techno-gadgets has perhaps improved the quality of life for society at large, they have stretched resources to the breaking point within the law enforcement community. The *preservation, identification, extraction, documentation, and interpretation of the information stored within these devices for evidentiary ... analysis* [Kruse and Heiser, 2001] is the province of the digital forensics specialist.

Unfortunately, every new hardware configuration poses a unique challenge to the forensics specialist trying to extract evidence to develop a criminal case. Typical issues are unfamiliar applications, file formats, or operating systems, and hardware devices or hardware problems not encountered before. Some configurations or devices occur infrequently enough that the solution, though known and previously used, is not clearly remembered and needs to be researched. Dealing with such issues can take days and weeks of "viewing the problem from different angles", following up dead ends and researching the technology.

For instance, in a recent case, one of the authors was asked to image a hard drive and provide the image file to another agency for analysis. He found that the hard drive exhibited some partition table errors that rendered it absolutely unavailable in the DOS or Windows environment. Ordinary tools for partition table examination could not identify the partition type, but the data area of the drive clearly contained a Windows operating system and typical applications for a home computer. Consequently, the examiner decided to make an image file using the Linux dd command. Unfortunately, the Linux dd command has literally dozens of combinations of command line switches. Luckily, someone had already researched the use of dd as a forensic tool, and posted a page (http://www.crazytrain.com/dd.html) providing exactly the information needed to put dd to use as a forensics tool. This lesson literally saved the investigator dozens of hours of research and trial and error.

We envision a mechanism where experiences such as the Linux dd lesson can be centrally maintained and indexed in a convenient manner for ease of location and retrieval. The lessons in this repository would be contributed by the users themselves to ensure that others benefit from their experiences.

A Repository of Lessons Learned

The goal of maintaining a Repository of Lessons Learned is broad dissemination of information about experiences that will discourage the use of work practices that lead to undesirable outcomes and encourage the use of work practices that lead to desirable outcomes. It is tempting to attempt to expand the focus of a Lessons Learned Repository to encompass other goals, thus it is important to list what a Lessons Learned Repository is not. It is not:

- A collection of general best practices
- A set of tutorials
- Unindexed links to a set of “how to” documents
- “Official Guidelines”
- Academic ideas about what should work

Rather, it is an organized method by which other digital forensics specialists can share their experiences – some that have worked, and some that have not, with colleagues in a manner that does not require a sifting through numerous irrelevant lessons in order to find one that is useful. Consequently, a viable Lesson Learned has three special attributes. A Lesson is:

- *Implemented by the Contributor*. The work practice or approach being described must have really been exercised – it should not be just a speculation. A speculative Lesson might be “my friend knows a fellow that works for Dell that says you can use a floppy boot disk to bypass the start-up files on the disk”.
- *Applicable to actual use*. The lesson should be phrased generally enough so that it is transferable, yet specific enough to identify a particular action that should be taken. For instance, simply exhorting the investigator to “look for words associated with identity theft” is really too general. A more appropriate Lesson might be “look for strings including ‘SS#’, ‘PIN’, ‘Visa’ and ‘AMEX’”.

- *Valid*. The contribution must have a significant impact on some outcome and be factually and technically correct. For example a Lesson such as, "to save time, we just analyze the original hard drive rather than wasting time making a copy – we've never had any trouble in court" is probably invalid since the common wisdom is to never work directly from the original media, but rather to work from an imaged copy.

The Key Issue for the Lessons Learned Repository

While many issues are involved in deploying a successful Lessons Learned Repository, the bulk of the issues can be summarized as motivation. For instance, what is the motivation of someone that contributes a Lesson? How can we encourage forensics specialists around the world to go to the trouble to contribute an experience, especially if it entails special formatting, indexing, etc.? Likewise, how can we motivate users to make use of the contributions?

Other repositories of user contributions have attempted to address these issues in different ways. The Xerox Eureka Project and Epinions are two different approaches to allowing contributions and use of experiences.

Xerox's Eureka Project is a repository of over 25,000 tips contributed by Xerox's service reps. Service reps often encounter undocumented problems, and are encouraged to submit their solutions to Project Eureka, so other reps can benefit from their experience when confronted with the same undocumented problem. Prior to publishing a tip, it is reviewed by a panel of peers who agree that the tip will work. Service reps are motivated to contribute tips because their names are associated with their contributions, and therefore they gain a significant amount of internal recognition from the dissemination and use of their tip.

Conversely, Epinions is a web-based Information Exchange in which contributors post advice, product reviews, opinions, and recommendations. Rather than filtering contributions via an organized review panel, users are encouraged to rate the degree to which they trust various contributors. Contributors are identified via biography, list of prior reviews and comments by users. Users assess how trustworthy a contributor is and therefore whether or not their review should be viewed. This is significant to contributors since unlike Project Eureka, contributions are motivated by a payment of one to three cents per view to the contributor.

It is unclear if any single answer to the motivation issue is the best or most appropriate. However, this is the key question to maintaining a successful user contribution repository. We are currently investigating ways to motivate the forensics community to become both Contributors and User.

Important Functions of a Lessons Learned Repository

The functions provided by a Lessons Learned Repository can be broken down into the following three activities:

- Collecting the Lessons
- Storing and Maintaining the Lessons
- Retrieving and Using the Lessons

These activities are closely intertwined, in that shortcomings in one adversely affect the others. Consequently, it is important that each of these functions undergo careful planning prior to deployment.

Collecting the Lessons

The value of a Lessons Learned Program is a function of how much experience people are willing to contribute, and how well the Lesson is packaged by the contributor. Packaging includes explaining the Lesson as well as indexing it so the Lesson can later be easily located.

There is some question as to the best way to represent a Lesson. We have identified at least a rudimentary collection of information that should be associated with a Lesson:

- A generalized 2-3 line summary of the Lesson that concisely represents the Lesson so the User can tell if it is applicable to their needs without reading the entire Lesson.
- A discussion of the Details relating to this Lesson. It is expected that this section convey the “wisdom” to be found in the Lesson. While this should not be a voluminous tome, some detail should be conveyed so the Wisdom can be replicated in another context.
- A series of "index categories" intended to aid Users in locating applicable Lessons. These indexes consist of *Beneficiary*, *Phase*, *Classification* and *Technology*. The index information is selected from a pre-defined list of choices if possible to avoid “index sprawl” and enforce consistency in Contributor indexing. Notwithstanding the importance of consistency, in the event the Contributor finds no pre-defined entry that meets their needs, they may select **Other**, at which point they have an opportunity to add a non-standard value. Multiple classifications may be associated with a particular Lesson. An initial collection of Beneficiaries includes {First Responder, Forensics Technician, Investigator/Detective, and Prosecutor}. An initial collection of Phases includes {Discovery, Seizure, Imaging, Analysis, Case Construction and Prosecution}. An initial collection of Classifications include {Fraud. Child Porn, Identity Theft, Homicide, Death Threats, and “Not Specific to a classification”}. An initial collection of Technology indexes is {Windows, Linux, Cell Phone, PDA, Laptop, and Network}. Naturally, each index allows the Contributor to add a new index term to the Repository, if no pre-defined index is suitable.
- Optional electronic mail/name/agency of the Contributor. This will be discussed more in a later section on “Policies”. Because of the potential desire for anonymity, this information is currently optional. However, we are also cognizant that Lessons lacking this information may be considered less credible by Users.

There are a number of opportunities for a good Lesson to be rendered useless if the Contributor is expected to provide all of this information. The Summary could be misleading or needlessly terse so the User doesn’t think the Lesson is worth pursuing. The Details could be poorly written so that either it is difficult to understand how to generalize the Lesson so it can be used in other contexts, or the User simply can’t tell what the Contributor is talking about. Poor indexing can result in the Lesson being missed by the User if they choose to retrieve Lessons based on the indexes.

Because it is not trivial to write a good Lesson, a Contributor can expect to invest a significant amount in time in crafting the Lesson they wish to add to the Repository.

Retrieving and Distributing the Lessons

Once Lessons are entered into the Repository, the User must be able to conveniently access them. Assuming an applicable Lesson exists in the Repository, there are several aspects that will affect the likelihood of the Lessons being retrieved and used:

- Appropriate Lessons appear, while inappropriate Lessons do not, so the User does not need to wade through a large number of "false matches".
- When an appropriate Lesson appears, the User recognizes it. Therefore, the Summary information must be descriptive enough to identify the Lesson as appropriate since Users are unlikely to read each Lesson in its entirety.
- The Lesson must be general enough so it can be used in a different context, while being specific enough so that the User can understand what to do.
- The User must trust the Repository in general, and the Lesson in particular.

Naturally this deals with both the way Lessons are entered into the Repository as well as the interface that is presented to the User. Several interaction approaches are possible, including index searches, full-text, "in-string" searches and simply allowing users to browse through the individual Lessons. Additionally, when using searches, providing some form of "matching score" so Users can tell how well a Lesson matches their criteria, so Lessons that just match one index out of several are flagged as being "less of a match" than Lessons containing every search term.

The issue of trust is equally significant. Users must trust a Lesson before applying it. Trust can be established in several ways. One approach is to allow Users to know the identities of the Contributors. However, there is a possibility that many Contributors may choose to remain anonymous. A second option to building trust is to learn that others have used the Lesson successfully. Consequently, by providing a feedback mechanism by which Users may associate comments with Lessons describing their experiences, they may become more comfortable using the Repository as a whole, as well as in using individual Lessons.

Storing and Maintaining the Lessons

The ease of retrieving Lessons is obviously a function of how they are maintained within the Repository. Clearly, they must be stored in a manner that allows both indexes as well as the body of the Lesson itself to be searched. There are several data structures that support these capabilities, from a simple flat file consisting of Lessons to a sophisticated database with inverted indexes and multiple keys.

Many times, however a User may wish to share a Lesson with someone else, or even save it for future reference. Representing Lessons within a database implies that the User must use the LLR interface to access specific Lessons. This is inconvenient if the User simply wants to look at a Lesson they found last week. Consequently, maintaining Lessons as individual XML pages appear to be a more logical choice for ease of use. The pages can be easily searched by serverside applications, while at the same time allowing individual Lessons to be bookmarked or e-mailed to others.

Repository Policies

The technology behind constructing a Lessons Learned Repository is fairly straightforward. A more open set of issues is the policies that may be instituted in its use and operation. These are vital issues because they impact the Contributor's willingness to add Lessons and the User's willingness to retrieve and use Lessons. Without Contributors and Users, the Repository cannot be a success. In the discussion on key policy issues that follows, we choose to simply describe the extremes of the possibilities rather than to advocate for one approach or another. We hope to perform additional research in the future to identify the impact of various policy choices on contributions and use.

Who can add a Lesson?

The "Contributor policy" has great significance as to how useful the Repository will be to others. This policy could be as loose as allowing anyone with access to an Internet connection and a web browser to add Lessons at will. On the other hand, one could imagine a much more restrictive policy in which individual users are "certified" as competent to add a Lesson.

In the first case, we could expect a relatively large number of contributions. However, many may turn out to be not implemented, not applicable or not valid. Some of these may be added either maliciously (or at least recklessly) while others may not be usable simply because the Contributor is not competent to be giving advice. We do not have an informed opinion as to what the ratio of unusable to usable Lessons would be using a least restrictive Contributor policy.

In the second case, most Lessons would likely contain a large amount of wisdom that would be of great utility to others. On the other hand, many potential Contributors that may not be willing to go through a "certification process" are still more than competent enough to make a valuable contribution. While we would expect the ratio of unusable to usable Lessons to be quite low, we would also expect that the total number of Lessons would be quite small using this much more restrictive Contributor policy. The use of a "Certification" for Contributors would also make the operation of the Repository much more expensive since we might assume the process would be non-trivial, and the operators would bear the costs of providing the "certification".

Who can read a Lesson?

The "User policy" primarily affects the degree to which the Repository is used as well as the effort required to operate it. In the least restrictive implementation of this policy, anyone could have access to the contents of the Repository. There is some danger that this may affect the willingness of some Contributors to participate. Many forensics specialists believe that their methods are too sensitive for access by the general population. The concern is that if the "bad guys" learn how forensics are done they will quickly arrive at ways to make forensic analysis more difficult.

Under the most restrictive implementation of the "User policy", only representatives of bona fide law enforcement organizations would be able to access the content of the Repository. This would make the operation of the Repository extremely expensive, since every user would have to be evaluated. Likewise, the process may very well be overly expensive from the User's standpoint since they would have to provide documentation in order to be given access to the Repository.

If the contents of the Repository are truly sensitive secrets, only to be known by the law enforcement community, this may be a difficult issue. However, if most of the techniques are well known throughout the general technical community and/or available through court records or trial transcripts, the effort may be moot.

Who (if anyone) filters Lessons?

The "Filter policy" is important from the perspective of User trust. It is extremely unlikely that most Repository users will know each other, so even if Contributors are identified, a User may not know if a contribution is trustworthy or not. Having a review of each Lesson prior to its publication in the Repository may increase the level of User trust.

However, a review procedure can add a great deal to the maintenance cost of the Repository. There is a great deal of variability in what comprises a "review". At one extreme, a review may simply be a cursory examination to determine if the Lesson is readable and makes sense. The other extreme may involve attempting to replicate a Lesson before it is added. We could also imagine reviews that check to ensure that new Lessons don't duplicate existing Lessons. This leads to a situation that could be potentially labor intensive.

Oddly enough, a review process may also make Contributors less willing to submit Lessons since they may view it as investing a great deal of energy and time to produce a Lesson that may not be accepted by the review board.

How much does a Contributor need to tell about themselves?

The policy on anonymous contributions is sensitive since many potential Contributors are concerned that they may be targeted by criminals for retribution if their work is revealed. Equally important is that many agencies may have policies about publishing information, that results in Lessons having to be approved prior to submission. The least restrictive implementation of this policy is simply allowing Contributors to remain totally anonymous. On the other hand, this adversely impacts the amount of trust Users have in the Lessons they retrieve, as well as reducing the opportunity to follow up with the Lesson author if a question arises.

The alternative to total anonymity is to require Contributors to associate their name, affiliation and even e-mail address with a Lesson. This way, potential Users have some sense of the qualifications of the Contributor, and well as a point of contact for follow-up questions. Since contributors are identified, this also provides a certain amount of recognition among Contributors. Because this is strictly a voluntary contribution of wisdom, the recognition that goes with a well-used Lesson may provide some motivation for Contributors.

A Prototype Forensics Lessons Learned Repository

In order to better explore many of the issues that must be addressed prior to deploying a stable Repository of Forensics Lessons, we have implemented a "proof of concept" prototype LLR. This prototype has provided a mechanism by which we have studied the mechanics of Lesson collection and retrieval. The prototype is at http://forensics.LessonsLearnedRepository.org and currently implements the least restrictive interpretation of each policy issue described above.

Future Work on LLRs

The most significant work that needs to be done with respect to a forensics Lessons Learned Repository is the effect of various policy decisions on usage. Because we choose to measure our success based upon usage and utility to practitioners, we view our policy decisions as being "market driven" rather than "prescriptive". To this end, we are currently considering an effort to evaluate various policy decisions from this perspective.

Regardless of policy decisions, people will seldom use an empty Repository. There is, of course, no value to Users and Contributors see no point in adding Lessons. Therefore, we are also investigating ways in which we can "prime the pump" by automatically extracting Lessons from various mailing lists, as well as building a group of forensics specialists willing to contribute a core set of Lessons.

Additionally, there is a great deal of promise in novel Lesson retrieval mechanisms. For instance, some on-line commerce sites recommend additional products based on products that the user has just purchased. We could imagine having additional Lessons recommended based on Lessons the User has just visited.

References

[Kruse and Heiser, 2001] Kruse, Warren G. II and Jay G. Heiser, *Computer Forensics : Incident Response Essentials*, Addison-Wesley Pub Co, 2001.

© 2002 International Journal of Digital Evidence

About the Authors

Warren Harrison is a Professor of Computer Science at Portland State University and a Police Reserve Specialist in Digital Forensics with the Hillsboro (Oregon) Police Department.

David Aucsmith is Architect for Microsoft's Security Business Unit. He is responsible for the security design of Microsoft products. He is a Police Reserve Specialist in Digital Forensics with the Hillsboro (Oregon) Police Department.

George Heuston retired as a Special Agent with the FBI after 21 years, seventeen of which were spent working high tech investigations. He is currently a Management Analyst for Hillsboro (Oregon) Police Department. He is currently an Adjunct Professor with the Portland State University Department of Computer Science.

Sarah Mocas is an Assistant Professor of Computer Science at Portland State University and a Police Reserve Specialist in Digital Forensics with the Hillsboro (Oregon) Police Department.

Mark Morrissey is a Computer Science Instructor at Portland State University and a Police Reserve Specialist in Digital Forensics with the Hillsboro (Oregon) Police Department.

Steve Russelle is a Detective in the Computer Crimes Unit (Forensic Data Analysis) of the Portland, Oregon Police Bureau. He is also an instructor for New Technologies, Inc. in Gresham, Oregon where he teaches specialized data hiding techniques and electronic eavesdropping considerations.