Constructing and Evaluating Digital Evidence for Processes

CARSTEN RUDOLPH and NICOLAI KUNTZE

ABSTRACT Digital evidence cannot be restricted to single evidence records. A large part of forensic investigations is concerned with relating events to each other and relating events to persons that have initiated them. This paper proposes different options to use meta-data models and meta-data graphs to support this task and also discusses the advantages and disadvantages of the different approaches.

Introduction

The task of reliably documenting processes in technological systems is complex and is usually executed separately on different levels. I.e. network events are collected and evaluated independently from information on application level processes. Existing logging can provide audit trails for subsequent evaluation of what has happened. Nevertheless, even when considering only basic requirements on forensic readiness, such audit trails can usually not be considered to be sufficiently secured against manipulations in particular during collection of the information. However, recent work on creating secure forensic evidence shows how single data records can be produced and stored in a secure way and how digital traces of evidence can be constructed.

This paper proposes a technological basis to construct digital evidence for several linked events in order to reliably document a complete process. The linking of events can either occur through information available in the events themselves or by indirectly linking evens through meta-information that again needs to be recorded by another securely documented event.

One example for such a chain of events is the process leading to measurements done by calibrated devices. It is obvious that data records produced by such a device need to be protected in order to be suitable as digital evidence. However, the event of measuring also needs to be linked to the process of constructing and calibrating the device.

Another example is the documentation of processes in enterprise networks. Reliable meta-data information can link actions on a network level (e.g. network access, user login via a central authentication server,etc.) to application level events executed by the same user using a particular device. Current logging systems can provide forensic data for some events on the network level. However, the linking to application processes requires additional information that is not available in current enterprise networks.

The approach discussed in this paper combines hardware-based security solutions with so-called meta-data access protocols and meta-data graphs visualizing the information linking of events. This visualization can also support the evaluation of the collected evidence by clearly showing the relations between the different events.

The use of meta-data information provides a much broader view on what kind of information given by digital evidence. Furthermore, it shows the large scope of associations that can be expressed by digital evidence and that therefore needs to be considered in the evolution of digital forensics and also in the view of digital data by lawyers and courts.

Two Examples for digital processes

This section discusses two scenarios for digital processes. In both scenarios there are obvious requirements for documenting essential parts of the process. Furthermore, it is not straightforward to relate events to a particular instance of the process and to relate activities by a particular person to the instance of a process. Usual logging activities might not be sufficient for a forensic reconstruction of instances of the processes. In general, it is not obvious how events can be linked. One task of digital forensics is concerned with

Linking events and relating technical events to each other and to physical events. In particular, in the case of conflicts, relating technical events and digital documentation to actions executed by persons can become relevant.

The two scenarios show that different characteristics of a digital process can be relevant. The first process occurs in an enterprise network with remote access. All physical human activities in the process are concerned with people accessing a computer connected to the network. The second example also includes other technical activities like the installation of devices and the configuration and calibration of these devices.

Nested authentication in enterprise networks

This scenario assumes that a data-base application is installed on a server somewhere in a protected part of an enterprise network. Further, it is assumed that transactions on the data stored in the data-base can be critical. Examples for such critical actions can be access to confidential information, changes to accounting data, financial transactions, changes to banking and payment information, but also technical actions in industrial automation scenarios. The actual type of transaction is not relevant for the discussion on digital processes within this example. The actions discussed here are concerned with user authentication and access to particular computers on the network. However, it should be noted that also transactions within one application can be seen as a digital process where similar requirements can occur on the application level. The main goal of the digital evidence should be to prove which individual was responsible for invoking a particular transaction.

The example scenario consists of the following possible sequence of events. All human actors can be employees of a company. This is not about escalation of privileges or other typical security attacks. All actors can own the right to execute all transactions in the process. The goal is to be able to relate transactions to persons:

1. Actor Alice remotely logs in to the enterprise network using a virtual private network VPN. She uses her own credential to securely establish the VPN connection. The credential can be implemented as a password, as a digital credential stored on a SmartCard, or some combination of password with a physical token. After authentication, she has access to a protected part of the network behind a firewall, e.g. she can have access to a particular computer in the internal network.

2. In order to access the critical application and execute a transaction Alice establishes a remote desktop application to a server in the protected network in order to get access to the graphical user interface of the application. The server initiates a second authentication process, e.g. by requesting a pair of user-name and password. Now, Alice can

either use her own credentials or another username and password known to here for some reason (e.g. by her colleague Bob).

3. On the server, Alice can now start the application. Once more, authentication might be requested, this time by the critical application itself. Again, Alice can either use her own credentials or another username and password known to her.

4. Finally, Alice executes a critical transaction and terminates the connection.

All steps can be logged and we assume for the discussion that secure logging mechanisms exist and are used[1]. In the case of a conflict (or simply for documentation) it can be necessary to know who actually initiated the critical action using the application in question. The documented events show that someone was logged in to the application and the transaction can of course be related to the person whose credentials where used to log in to the application. Now assume that Bob's credentials where used and Bob denies having initiated this transaction. Various attack vectors are available to Alice to steal a password from Bob. Examples include key-loggers installed on Bob's computer[2] or simply spying on Bob which is perfectly possible if Alice is a co-worker. Also social engineering techniques can be applied. Examples for such techniques include phone calls apparently coming from a technical administrator asking for a password.

[1] See: *A theoretical framework for organizational network forensic readiness*. (2007) Published in: Journal of Computers, Volume 2, Issue 3, Authors: Barbara Endicott-Popovsky, D. Frincke, C. Taylor and *New Calibration testing of network tap devices* (2007) Published in: Advances in Digital Forensics, Authors: Barbara Endicott-Popovsky, B. Chee, D. Frincke

[2] See: *The rise and rise of the keyloggers,* Published in: Network Security Volume 2007, Issue 6, June 2007, Author: Simon Heron

Clearly, digital forensics needs to be used to resolve this conflict. In principle, all logging information to determine the actual sequence of events should be available on the system. However, relating these different events and showing the correct sequence is not straightforward and it is not obvious that the logged information is actually suitable to correctly relate events. One can however expect that events do contain some information that can relate events either directly (e.g. by comparing IP addresses, user-names, process IDs, etc.) or indirectly (e.g. via events logged by some authentication server or directory). Nevertheless, even though this information can be available, actually identifying the correct parameters and re-constructing the process is difficult. Furthermore, relations between events are seldom unique and just by evaluating events in the context of some static information on the enterprise network can result in ambiguous results.

Clearly, the re-construction of a digital process can be supported by meta-information (or meta-data) for the different events. This meta-data should identify those parameters that can be used to relate events and describe the relations that can occur. Parts of this meta-data can be available on so-called configuration management data-bases (CMDBs)[3]. However, the information in CMDBs is concentrated on the infrastructure and does not include meta-data for applications and other higher layers. Thus, additional meta-data can be necessary and this meta-data will most probably evolve faster than the information on the network infrastructure. Thus, no static meta-data set can be assumed and in addition to the logs also a currently valid meta-data set needs to be stored.

[3] http://www.itil-officialsite.com/

In addition to the construction, storage and evolution of meta-data sets, this information can also be used to analyze possible evidence on processes. Knowledge of the meta-data showing relations between events can be used to determine (at the time of developing the system/network) if available meta-data will be sufficient to provide all relevant relations between logged events.

The complexity of the forensic evaluation in this example should not be underestimated. Many other persons could have been logged in at the same time and even Bob might have used the critical application himself in parallel to the use by Alice impersonating Bob. Furthermore, once the actual process leading to the critical transaction has been identified, it is still not straightforward to conclude which action has been involved by which person. In addition to the identification of the correct sequence of events in the digital process, it is necessary to evaluate which conclusions can be drawn. This process can include a risk analysis for the different actions. One example can be to distinguish different security levels for the authentication. Obviously, digital evidence for processes can only be one element in the dispute resolution.

Another property of this example is, that relevant events occur in a rather short time-frame, restricted by the duration of authenticated sessions in all different parts of the network (VPN, remote desktop, application). The example in the following section is different as relevant events can be spread over a much longer time.

Digital speed cameras

This example revises a scenario previously introduced by the authors[4] . The scenario consists of a digital camera to

[4] *Securing Digital Evidence,* Published in: 2010 Fifth International Workshop on Systematic Approaches to Digital Forensic Engineering, Authors: Jenny Richter, Nicolai Kuntze,

record speeding and process picture data, a server collecting data records with the computed information on speed, number plate etc., network components for communication, long-term storage for data records, and finally, evaluation components.

The existing technical solution provides secure digital evidence bound to the status of the device. Furthermore, a framework exists that enables the control of the validity of the chain of evidence at run-time. However, in reality this system is integrated in a much more complex infrastructure and the reported valid chain if evidence does not provide sufficient information for all situations. Logging data and monitoring can provide information that is not directly related to the actual chain of evidence showing the validity of the measurement, the picture and the evaluation of the values. However, in addition to the actual chain of evidence it can be necessary to also consider supporting processes and document these processes. Such processes include maintenance of the camera, security management for the server and the infrastructure the server is running in, access control (technical, but also physical), and administrative processes like issuing speeding tickets and factorization.

Clearly, not all of the processes mentioned above are relevant for forensic evaluation and not all of them need to be related to the original chain of evidence. However, depending on the character of a dispute some events need to be considered and also related to each other and to particular instances of the main chain of evidence. Again, it might not be possible to maintain a meta-data graph at run-time for all the different events and store the complete history of meta-data. Therefore, different ways need to be found to store meta-data information such that they can be used to evaluate stored log data.

and Carsten Rudolph

In addition to the storage of meta-data sets to enable off-line calculation of the relations between events, also run-time aspects can be relevant. In particular in this scenario of speed cameras it can be relevant to continuously check that the collected information constitutes valid chains of evidence. Checking the validity of digital signatures on single data records is one part of the validation. However, as described above, all relevant parts of the process should be considered. Thus, a meta-data graph can be constructed at run-time that shows the relations between different events. Evaluation of this meta-data graph can show that collected evidence is sufficient when seen in the context of the current meta-data specification.

Digital evidence for processes using meta-data information

This section first revises existing technical solutions and then discusses two different ways to deal with meta-data information to construct and evaluate digital evidence for processes.

Technical building blocks

Trusted computing and digital evidence

One important aspect of the generation of digital evidence is the status of the device used in the process. The software and configuration used to produce evidence needs to be presented and linked to the individual record. One simple scheme hereby is to include software name and version number as a simple string of text in each evidence record. This first (and often used) approach allows for uncertainties with respect to updates and various attacks on the evidence records. Just naming the software is not sufficient if the device can be manipulated. Stronger means of protection are therefore required to reliably document the software and configuration of the particular evidence generator. To provide proof on the actual state of the evidence generator, trustworthy reporting in the device is required. The Trusted Platform

Module that is standardized by the Trusted Computing Group TCG[5] introduces a core root of trust for measurement which establishes the foundation to report on the status by creating a chain of trust[6]. This chain of trust can be reported to external entities to allow for a verification of the evidence generator. This verification process is called Remote Attestation.

Application of remote attestation allows for a session-based or per-record scheme to protect digital evidence. The *session-based* approach relies on an initial attestation of the system and a session bound to the individual evidence generator and status. Each evidence record is then cryptographically bound to this session and therefore to a particular system state. The second *per record* scheme involves an attestation process for each evidence record. As in the basic remote attestation, an external random number generator is involved, and longer delays as well as higher bandwidth utilization are to be expected. More advanced schemes allowing for scalable attestation schemes can be applied[7].

The approach to hardware-based evidence generation linking the evidence to the platform state was first presented in SADFE 2010[8].

[5] http://www.trustedcomputinggroup.org/

[6] See: *Dynamics of a Trusted Platform: A Building Block Approach,* Intel Press, 2009, Author: David Grawrock

[7] See: *Improving the Scalability of Platform Attestation,* Published in: Third ACM Workshop on Scalable Trusted Computing, 2008, Authors: Frederic Stumpf, Andreas Fuchs, Stefan Katzenbeisser, and Claudia Eckert

[8] See: *Securing Digital Evidence,* Published in: 2010 Fifth International Workshop on Systematic Approaches to Digital Forensic Engineering, Authors: Jenny Richter, Nicolai Kuntze, and Carsten Rudolph

Meta-data and IF-MAP for the construction of digital chains of evidence

A technical infrastructure for a secure collection of events and for storing them with matching meta-data was shown in SADFE 2011[9]. The interface to meta-data access points (IF MAP)[10] as an established industry standard and part of the Trusted Network Connect (TNC) protocol stack defines and supports event distribution and correlation in the domain of network access control but can easily be extended to support other types of events and create event graphs representing relations between different types of events. To use IF-MAP to relate events and correlate pieces of evidence was also first proposed in SADFE2011. By now, IF-MAP meta-data models have been extended to include events on mobile devices[11] and a first draft of a meta-data model for industrial control systems also exists. For the creation of digital evidence, these meta-data models need to be extended for each scenario. In particular, including information for events provided by particular software or domain-specific meta-data models for particular application domains can provide cross-layer relation of events.

Run-time evaluation of meta-data versus offline computation of meta-data graphs

The precondition for all evaluation of meta-data is the existence of a meta-data model that describes meta-

[9] See: *Secure digital chains of evidence* Published in: Sixth International Workshop on Systematic Approaches to Digital Forensic Engeneering, 2011, Authors: Nicolai Kuntze and Carsten Rudolph

[10] See: *TCG Trusted Network Connect - TNC IF-MAP Binding for SOAP Version,* http://www.trustedcomputing.org/

[11] http://www.esukom.de/

information for events and thus expresses relations between different events. In the IF-MAP standard, the meta-data model is described in an XML following a standardized XML schema. Relations between different entities in the meta-data model can be shown as a meta-data graph. The complete meta-data model in defines the biggest meta-data graph for a system. In reality, this biggest graph will never occur, but all meta-data graphs representing a concrete system state are sub-graphs of this biggest graph.

Computation of meta-data at run-time and storage of sub-graphs

The run-time evaluation of meta-data can have different goals. First, meta-data graphs can be used to validate digital evidence for processes at runtime and ensure that the stored evidence together with the meta-data information can indeed be considered valid evidence. Second, an evaluation component could subscribe to relevant events in the graph and store each change in the graph together with collected digital evidence records. Then, these smaller meta-data graphs can directly be used to re-construct digital processes from events and relate events of the process or the complete process to persons responsible for initiating the events.

The computation of meta-data at run-time has two advantages. First, it is possible to continuously check that collected digital evidence is not corrupted and can in case of a dispute indeed be used to re-construct processes. Second, forensic evaluation is much easier if relevant subsets of the meta-data graph are stored together with the actual evidence. It can also be used to automate the evaluation of digital evidence and to generate enriched meta-data graphs that provide direct access to the relevant data records or log entries.

However, the run-time monitoring approach also has restrictions. Additional reporting of logged events needs to be done as the server building the meta-data graph needs to be informed about all events that are logged. This requires additional technical implementations. Even worse, privacy regulations might require to keep different logging information strictly separated and only allows forensic investigations to relate different digital evidence records. In addition, the amount of meta-data to be stored in addition to actual evidence records can get very high. Experiments have shown that already a meta-data graph representing services running on one single smart-phone consists of several hundred nodes and edges. Furthermore, such meta-data graphs can be very dynamic. To support evaluation of digital evidence in the context of the current meta-data graph, all changes need to be stored. In larger enterprise networks, meta-data information would considerably increase the amount of logging data to be stored.

Off-line evaluation of meta-data

In contrast to the scheme described in the previous paragraph, in this scheme only the meta-data model is stored and not the dynamic meta-data graph. The meta-data model is also subject to changes, for example whenever the network infrastructure is changed, new applications are installed, or new organisatorial roles introduced. Off-line evaluation means that event logs are collected and stored together with the currently valid meta-data specifications. Each change in the meta-data specification needs to generate a revision of the stored meta-data specification and this change needs to be logged in a way that there for each entry in the event log files the correct meta-data specification can be identified. One possibility is to include in all logs *change meta-data* events with unique identifiers for the old and new meta-data specifications. However, these changes occur at a very low frequency when compared with the meta-data graph. Thus, storing all changes in the meta-data model can easily be stored together with the remaining digital evidence in a forensic data-base.

The consequence that results from the more efficient monitoring is a more complex evaluation process. Evaluating digital evidence in the context of a meta-data model means to construct a meta-data graph from the log information. In principle, this can be achieved by *replaying* or *simulating* all the logged events and construct the meta-data graph for this behavior. In contrast to the run-time construction of the meta-data graph, relevant log events cannot be chosen beforehand and all events need to be computed to explore the meta-data graph and then to re-construct the process leading to the particular events under dispute.

Mixed scheme: Meta-data computation at run-time with restricted storage of meta-data information

In the case that critical processes can be identified on the level of meta-data information, it is possible to construct a scheme that uses run-time computation of the meta-data graph in combination with off-line evaluation. The complete meta-data graph is computed at run-time (e.g. by using an IF-MAP server) but the graph is not stored. Instead, it is continuously evaluated using rules that identify potentially critical processes. Then, the logged events leading to this particular meta-data subgraph are identified and tagged as events belonging to one instance of this process. It should be noted that single events can belong to several such processes. Then, in the case of a forensic investigation, the stored meta-data model can be used for a targeted reconstruction of the meta-data graph for this process. This subgraph then shows how the log events are related without the need of a long-term storage of large meta-data graphs.

Conclusions

Standard components can support the development of systems that provide secure digital evidence and also can support digital forensics by relating events. If devices, software and applications support the generation and distribution of event information in a standardized format (like IF-MAP) all components of the framework do not interfere with the infrastructure. Standardized components exist and MAP clients reporting events in the standardized format are developed for PCs, network components (routers, switches), mobile systems, and also for components of industrial control systems. In the long run, this development enables the consideration of digital evidence for critical processes already at design time and in addition can support forensic investigations.