THE “ART” OF LOG CORRELATION

Tools and Techniques for Correlating Events and Log Files.

Dario Valentino Forte, CFE, CISM,
Founder of the IRItaly Project

Corso Magenta 43, 20100 Milano, Italy
Tel (+39) 340-2624246
Web: www.honeynet.it
Email: dario.forte@acm.org

ABSTRACT

Log file correlation is related to two distinct activities: Intrusion Detection and Network Forensics. It is more important than ever that these two disciplines work together in a mutualistic relationship in order to avoid Points of Failure. This paper, intended as a tutorial for those dealing with such issues, presents an overview of log analysis and correlation, with special emphasis on the tools and techniques for managing them within a network forensics context. In particular it will cover the most important parts of Log Analysis and correlation, starting from the Acquisition Process until the analysis.

Keywords: Network Forensics, Log Analysis, Flow Reconstruction, Log Integrity.

1 LOGS: CHARACTERISTICS AND REQUISITES FOR RELIABILITY

Every IT and network object, if programmed and configured accordingly, is capable of producing logs. Logs have to have certain fundamental requisites for network forensics purposes. They are:

- Integrity: The log must be unaltered and not admit any tampering or modification by unauthorized operators;
- Time Stamping: the log must guarantee reasonable certainty as to the date and hour a certain event was registered. This is absolutely essential for making correlations after an incident;
- Normalization and Data Reduction. By normalization we mean the ability of the correlation tool to extract a datum from the source format of the log file that can be correlated with others of a different type without having to violate the integrity of the source datum. Data Reduction (a.k.a. filtering) is the data extraction procedure for identifying a series of pertinent events and correlating them according to selective criteria.

1.1 THE NEED FOR LOG INTEGRITY: PROBLEMS AND POSSIBLE SOLUTIONS

A log must guarantee its integrity right from the moment of registration. Regardless of the point of acquisition (Sniffer, Agent, Daemon, etc.) a log usually flows like this (Fig.1)

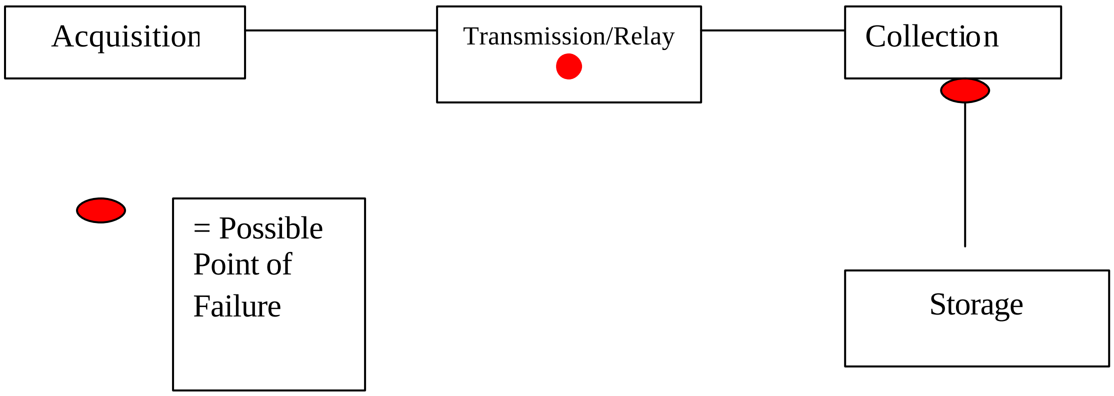

Figure 1 - Log Flow

Acquisition occurs the moment a network sniffer, a system agent or a daemon acquires the event and makes it available to a subsequent transmission process directed to a machine that is usually different from the one that is the source of the event. Once the log has reached the destination machine (called the Log Machine) it may be temporarily memorized in a pre-assigned slot or input to a database for later consultation. Once the policy-determined disc capacity has been reached, the data are stored in a predetermined location. The original logs are deleted to make room for new files from the source object. This method is known as log rotation.

Log file integrity can be violated in several ways. An attacker might take advantage of a non-encrypted transmission channel between the acquisition and destination points to intercept and modify the transiting log. He might also spoof the IP sending the logs, making the log machine think it is receiving log entries and files that actually come from a different source. The basic configuration of Syslog makes this a real possibility. The RFC 3164 states that Syslog transmissions are based on UDP, a connectionless protocol and thus one that is unreliable for network forensic purposes, unless separate LANs are used for the transmission and collection of log files. But even here there might be some cases that are difficult to interpret.

Another integrity problem regards the management of files once they have arrived on the log machine. If the log machine is compromised there is a very high probability of integrity violation. This usually happens to individual files, whose content is modified or even wiped. The integrity issue also regards how the paternity of log files is handled; in many juridical contexts, you have to be certain as to which machine generated the log files and who did the investigation.

There are several methods for resolving the problem. The first is specified in RFC 3195, which identifies a possible method for reliable transmission of syslog messages, useful especially in the case of a high number of relays (intermediate record retransmission points between the source and the log repository). The main problem in this case is that RFC 3195 has not been incorporated into enough systems to be considered an established protocol.

Hence, practically speaking, most system administrators and security analysts view SCP (Secure Copy) as a good workaround. The most evident contraindication is the unsuitability of such a workaround for intrusion detection purposes, since there is no real time assessment of the existence of an intrusion via log file reading. And the problem remains of security in transmission between the acquisition and the collection points. In response to the problem, in UNIX-based architectures the practice of using cryptcat to establish a relatively robust tunnel between the various machines is gaining wider acceptance.

The procedure is as follows:

On log-generating host:

1. you must edit /etc/syslog.conf in this mode:

```
 *.*                    @localhost
```

2. then run command:

```
# nc  -l -u -p 514 | cryptcat 10.2.1.1 9999
```

On log-collecting host:

1. run syslog with remote reception (-r) flag (for Linux)
2. run command:

```
# cryptcat -l -p 9999 | nc -u localhost 514
```

The above configuration will establish an encrypted connection among the various transmission nodes. An alternative would be to use a Syslog replacement such as Syslog – ng, which performs relay operations automatically and with greater security potentials.

From the practical standpoint, the methods described above offer a good compromise between operational needs and the theory that a hash must be generated for each log entry (something which is impossible in a distributed environment). The objective still remains of achieving transaction atomicity (transactions are done or undone completely) and log file reliability. The latter concept means being sure that the log file does not get altered once it has been closed, for example via interception during the log rotation phase. The most important aspect of this phase is the final-record message, indicating the last record written in the log, which is then closed and hashed. This sequence of processes may turn out to be critical when, after correlation, a whole and trustworthy log has to be provided to the judicial authorities.

1.2 LOG TIME STAMP MANAGEMENT: PROBLEMS AND POSSIBLE SOLUTIONS

Another problem of a certain importance is managing log file time stamping. Each report has to be 100% reliable, not only in terms of its integrity in the strict sense (IP, ports, payloads, etc.), but also in terms of the date and time of the event reported. Time stamping is essential for two reasons: atomicity of the report, and correlation. The most common problems here are the lack of synchronization and the lack of uniformity of the time zones.

The lack of synchronization occurs when the acquisition points (network sensors and Syslog devices) are not synchronized with an atomic clock but only within small groups. Reliance is usually placed on NTP in these cases, but this may open up a series of noted vulnerabilities, especially in distributed architectures connected to the public network. Furthermore, the use of NTP does not guarantee uniformity unless a series of measures recommended by certain RFCs is adopted for certain types of logs as we will describe below. Some technology manufacturers have come out with appliances equipped with highly reliable processors that do time stamping for every entry, synchronizing everything with atomic clocks distributed around the world. This sort of solution, albeit offering a certain degree of reliability, increases design costs and obviously makes management more complex. In a distributed architecture, a time stamping scheme administered by an appliance is set up as follows:

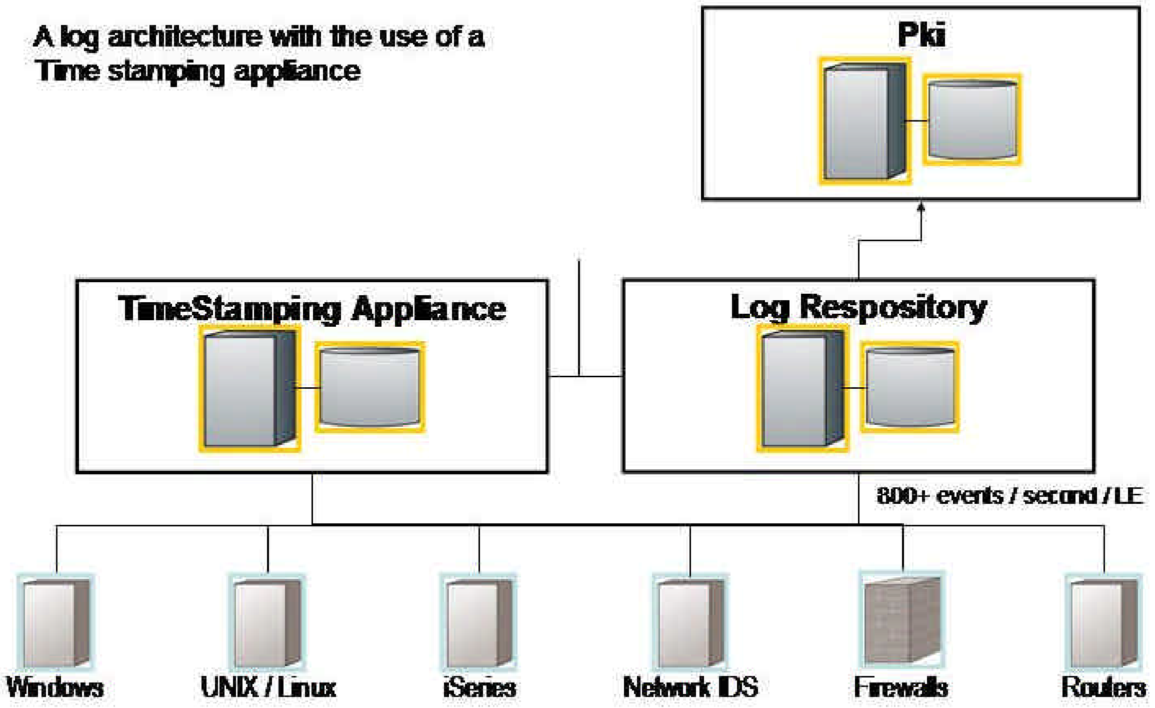

Figure 2: Log architecture with time stamping machine

The appliance interacts with a PKI that authenticates the transaction nodes to prevent the problem of report repudiation.

While this type of architecture may be "easily" implemented in an environment with a healthy budget, there are applications for less extensive architectures that may be helpful in guaranteeing a minimum of compliance with best practices.

Granted that one of the most commonly used log format is Libpcap-compatible (used by TcpDump, Ethereal) over TCP connections (hence 3-way), it is possible to attribute a further level of timestamping, as per RFCs 1072 and 2018, by enabling the SackOK option (Selective AcknowledgementOK). This option can return even a 32 bit time stamp value in the first 4 bytes of each packet, so that reports among transaction nodes with the SackOK option enabled are synchronized and can be correlated. This approach may be effective provided that the entire system and network is set up for it.

Another factor that is not taken into consideration are Time Zones (TZ). In distributed architectures on the international scale, some information security managers believe it is wise to maintain the time zone of the physical location of the system or network object. This choice has the disadvantage of making correlation more complicated and less effective because of time zone fragmentation. We are currently witnessing an increase of times zones being simply based on GMT, which has the plus of simplifying management even though it still requires that the choice be incorporated into a policy.

1.3 NORMALIZATION AND DATA REDUCTION PROBLEMS AND POSSIBLE SOLUTIONS

Normalization is identified in certain cases with the term event unification. There is a physiological need for normalization in distributed architectures. Numerous commercial systems prefer the use of Xml for normalization operations. This language provides numerous opportunities for event unification and management of digital signatures and hashing. There are two basic types of logs: system logs and network logs. If the reports all had a single format there would be no need for normalization. In heterogeneous architectures it is obvious that that is not the case. Let us imagine, for example, an architecture in which we have to correlate events recorded by a website, by a network sniffer and by a proprietary application. The website will record the events in W3C format, the network sniffer in LibPcap format, while the proprietary application might record the events in a non-standard format. It is clear that unification is necessary here. The solution in this case consists of finding points in common among the various formats involved in the transaction and creating a level of abstraction according to the diagram below.

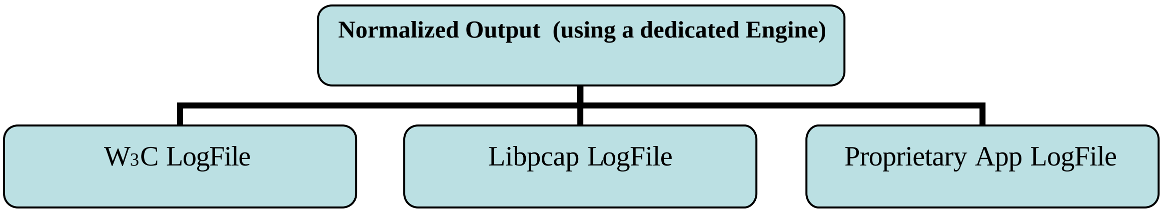

Figure 3: Normalization

It follows in this case that an attacker can once again seek to violate log integrity by zeroing in on the links between the various acquisition points and the point of normalization. We will discuss this below. Regarding the correlation, the point of normalization (normally an engine) and the point of correlation (an activity that may be carried out by the same module, for example, in an IDS) may be the same machine. It is clear that this becomes a potential point of failure from the perspective of network forensics and thus must be managed both to guarantee integrity and to limit possible losses of data during the process of normalization. For this purpose the state-of-the-art is to use MD5 and SHA-1 to ensure integrity and to perform an in-depth verification of the event unification engine to respond to the data reduction issue, keeping the "source" logs in the normalized format. In the following figure, where each source log is memorized on ad hoc supports, another layer is added to Figure 3.

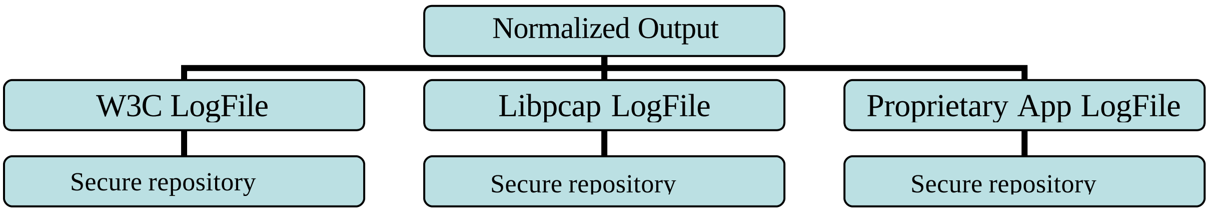

Figure 4: Multi-Layered Log Architecture

In order to manage the secure repository section and still use a series of "source log files" that guarantee a certain reliability, the machines in the second line of Figure 4 have to be trusted, i.e., hardened, and have cryptosystems that can handle authentication, hashing and reliable transmission as briefly discussed in Section 2.1.

2 CORRELATION AND FILTERING: NEEDS AND POSSIBLE SOLUTIONS

In performing log correlation and filtering, the Security Architect and the Manager have to deal with the problems described above. Here, the perspective on the problem shifts to the architecture.

2.1 CORRELATION AND FILTERING: DEFINITIONS

Correlation – "A causal, complementary, parallel, or reciprocal relationship, especially a structural, functional, or qualitative correspondence between two comparable entities". Source: dictionary.com

In this article we use Correlation to mean the activity carried out by one or more engines to reconstruct a given complex event, that may be symptomatic of a past or current violation.

By filtering we mean an activity that may be carried out by the same engines to extract certain kinds of data and arrange them, for example, by protocol type, time, IP, MAC Address and so on.

A fairly complex architecture may be set up as follows.

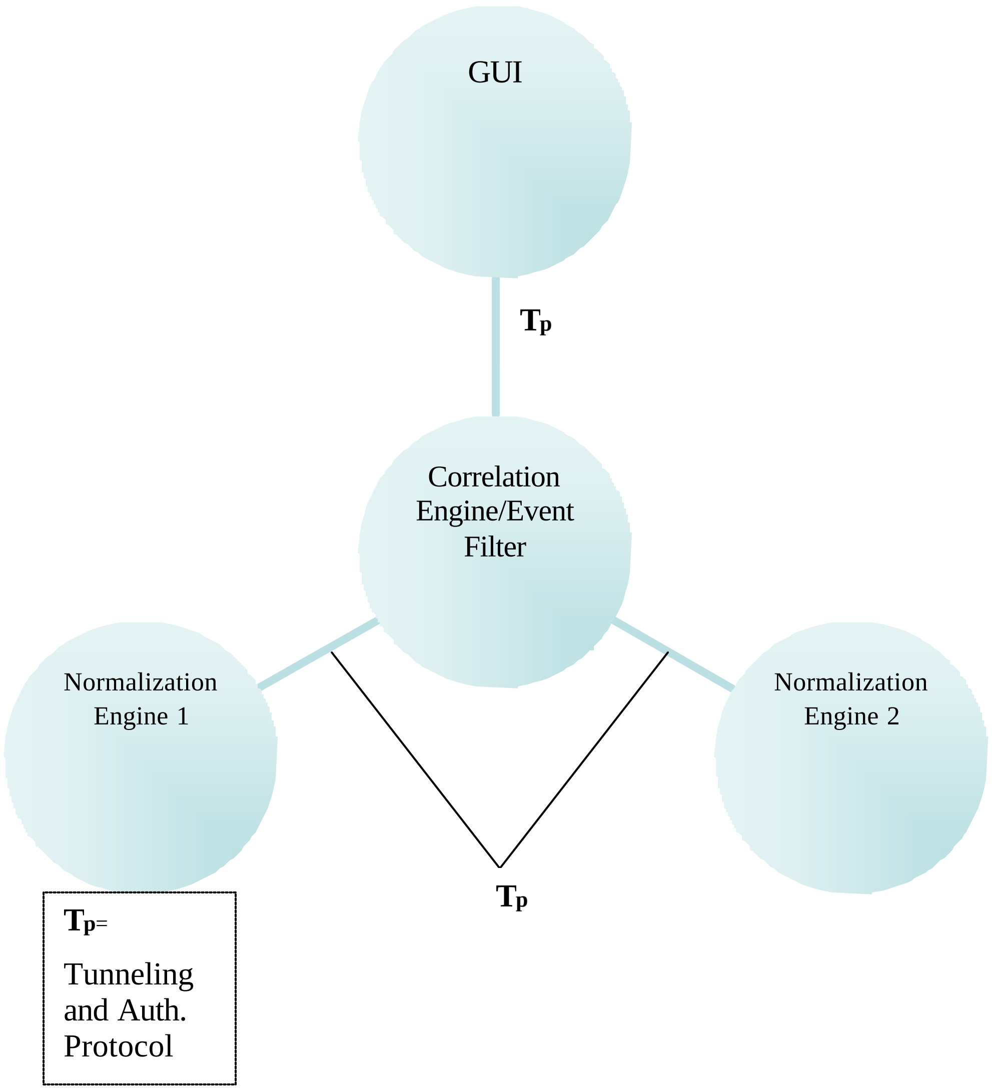

Figure 5: Correlating Normalized Events

As may be observed from Figure 5, and assuming the necessary precautions indicated in the above sections have been followed, if data is collected at the individual acquisition points (i.e., before the logs get to the normalization engines) by methods such as SCP, the very use of this method might slow down subsequent operations since these activities require a greater dynamism than the "simple" acquisition and generation of logs. Hence in this phase you have to use a Tunneling and Authentication (Tp) system based on a secure communication protocol that might be a level 3 ISO/OSI.

2.2 INTERPRETATION OF ONE OR MORE LOG FILES

In most cases the security administrator reads the result of a correlation done by a certain tool, but he only sees the tip of the iceberg. If you look at the figures in this paper, the set of processes upstream of the GUI display is much more complex. Whatever the case may be, the literature indicates two basic methods for analyzing logs, called *approaches*.

2.2.1 TOP-DOWN APPROACH

This is the approach most frequently used in network forensics when the examiner is working with an automated log and event correlation tool. While in intrusion detection a top-down approach means starting from an attack to trace back to the point of origin, in network forensics it means starting from a GUI display of the event to get back to the source log, with the dual purpose of:

1. Validating the correlation process used by the engine of the automatic log and event correlation tool and displayed to the Security Administrator;
2. Seeking out the source logs that will then be used as evidence in court or for subsequent analysis.

In reference to Figure 5, we have a top-down approach to get back to the source logs represented in the previous figures. Once retraced, the acquired logs are produced and recorded onto a CD-ROM or DVD, and the operator will append a digital signature.

2.2.2 BOTTOM-UP APPROACH

This approach is applied by the tool starting from the source log. It is a method used by the IDS to identify an ongoing attack through a real time analysis of events. In a distributed security environment the IDS engine may reside (as hypothesized in Section 2.3) in the same machine hosting the normalization engine. In this case the IDS engine will then use the network forensic tool to display the problem on the GUI. You start from an automatic low level analysis of the events generated by the points of acquisition to arrive at the "presentation" level of the investigative process. Such an approach, furthermore, is followed when log analysis (and the subsequent correlation) is performed manually, i.e., without the aid of automated tools. Here, a category of tools known as log parsers comes to your aid. The purpose of these tools is to analyze source logs for a bottom-up correlation. A parser is usually written in a script language like Perl or Python. There are however parsers written in Java to provide a cross-platform approach to network forensics examiners, perhaps on a bootable CD-ROM (see Section 5 for examples).

3 REQUISITES OF LOG FILE ACQUISITION TOOLS

Regardless of which vendor is chosen to represent the standard, the literature has identified a number of requisites that a logging infrastructure must have to achieve forensically compliant correlations:

- TCPdump support, both in import and in export;
- Use of MD5 or other state-of-the-art hashing algorithms;
- Data reduction capabilities as described in previous sections;
- Data Recovery. This feature comprises the ability to extract from the intercepted traffic not only the connections but also the payloads for the purpose of interpreting the formats of files exchanged during the transaction;
- Ability to recognize covert channels (not absolutely essential but still highly recommended);
- Read Only During Collection and Examination. This is an indispensable feature for this type of tool;

- Complete Collection. This is one of the most important requisites. It is important that all packets are captured or else that all losses are minimized and documented;
- Intrinsic Security, with special emphasis on connections between points of acquisition, collection repositories, administrative users, etc.

4 EXPERIMENTATION: USING GPL TOOLS FOR INVESTIGATION AND CORRELATION

So far we have introduced logs, correlation techniques and the associated security issues. Regarding the tools used for this type of analysis and investigation, there are GPL or opensource projects with the main goal of providing the necessary tools for a bottom-up investigation, which is a less costly and less complicated alternative to the top-down approach based on automated correlation and GUI display techniques. In this section we will introduce some projects and tools that may be used for the purpose at hand.

4.1 THE IRITALY PROJECT

IRItaly (Incident Response Italy) is a project that was developed at the Crema Teaching and Research Center of the Information Technology Department of the Università Statale di Milano. The main purpose of the project is to inform and sensitize the Italian scientific community, small and large businesses, and private and public players about Incident Response issues.

The Project, which includes more than 15 instructors and students (BSC and MSC), is divided into two parts. The first relates to documentation and provides broad-ranging and detailed instructions. The second comprises a bootable CD-ROM. The issues addressed regard information attacks and especially defensive systems, computer and network forensics on incident handling and data recovery methods.

Regarding response procedures to information incidents, best practices are presented for analyzing the victim machines in order to retrace the hacking episodes and understand how the attack was waged, with the final aim of providing a valid response to the intrusion. This response should be understood as a more effective and informed hardening of the system to reduce the possibility of future attacks. It does not mean the generation of a counterattack.

All the operations described so far are carried out with special attention to the method of identification, storage and possible use of evidence in a disciplinary hearing or in court. The unifying theme of the CD-ROM is the set of actions to undertake in response to an intrusion. It contains a number of sections offering a detailed analysis of each step:

- the intrusion response preparation phase;
- the analysis of available information on the intrusion;
- the collection and storage of associated information (evidence);
- the elimination (deletion) of tools used for gaining and maintaining illicit access to the machine (rootkits);
- the restoration of the systems to normal operating conditions.

Detailed information is provided on the following:

- management of different file systems;
- procedures for data backup;
- operations for creating images of hard and removable discs;
- management of secure electronic communication;
- cryptographic algorithms and their implementation;
- tools for the acquisition, analysis and safeguarding of log files.

The CD also proposes a number of standardized forms to improve organization and facilitate interactions between organizations that analyze the incident and the different targets involved in the attack. Specifically, an incident report form and a chain of custody form are provided. The latter is a valuable document for keeping track of all information regarding the evidence.

The CD-ROM may be used to do an initial examination of the configuration of the compromised computer.

The tools included offer the possibility to carry out analyses of the discs, generate an image of them and examine logs in order to carry out a preliminary analysis of the incident. The IRItaly CD-ROM (www.iritaly.org) is bootable and contains a series of disc and log analysis tools. All the programs are on the CD in the form of static binaries and are checked before the preparation of the magnetic support. After booting, the tool launches a terminal interface that the examiner can use to start certain applications such as TCPDump, Ethereal, Snort, Swatch and so on.

The CD can thus be used for a preliminary analysis of the logs present on the machine or for an analysis of the machine using the TASK/autopsy tool, which is more specific to the analysis of the hard disc. The correlation process, in this case, involves the comparison of logs present on the machine with others on other machines. In this case, the IRItaly CD essentially works in very small environments or even in one-to-one contexts, as illustrated below.

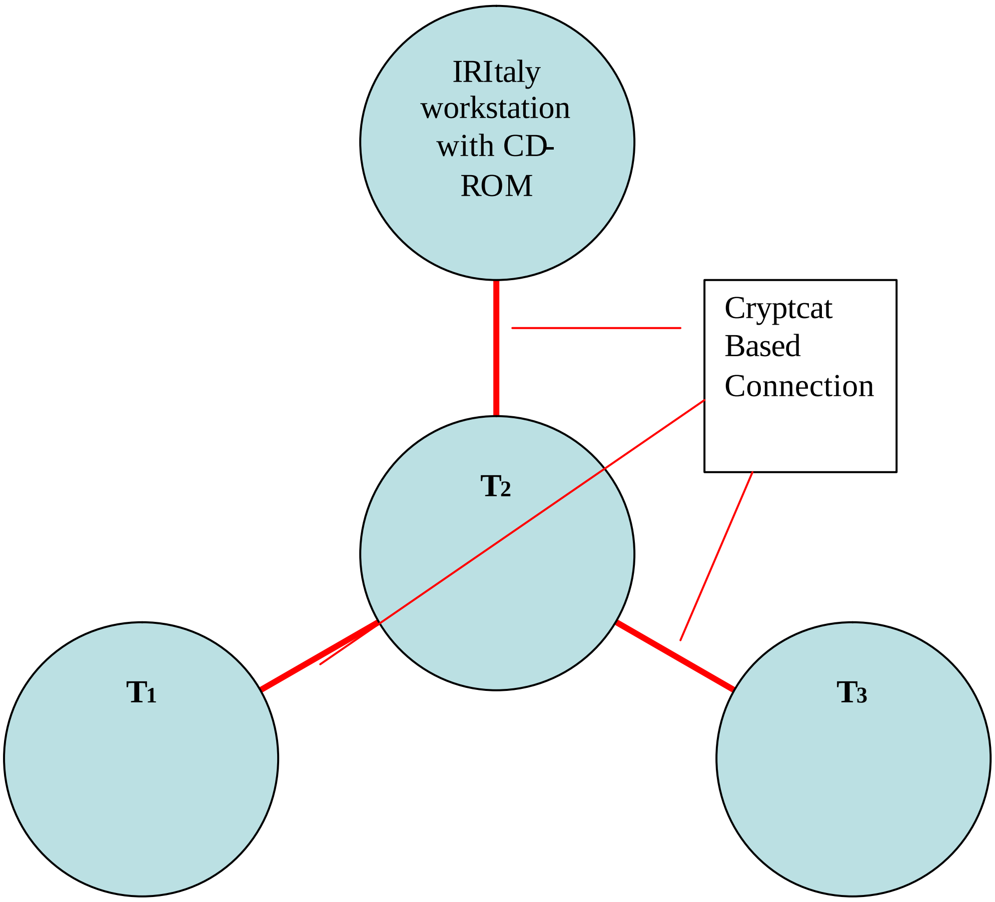

Figure 6: IRItaly CD-ROM Normal Use

Here, T1, T2 and T3 represent various targets that may be booted with the IRItaly CD and connected to the main forensic workstation with the aid of Netcat or Cryptcat. As stated above, the main limitation of the use of the completely functional CD is that it cannot be used in a distributed architecture due to obvious management difficulties. However, the IRItaly workgroup is carrying out a series of tests of a new version of the CD that should resolve some of the above problems with the aid of other tools. You may use IRItaly CD after validating it. The only possibile Point of Failure of this Project is that it is opensource/GPL Based and the validation process wasn’t made in court yet. Of course everybody can put in a test bed enviroment so the “validation problem” can be easily passed.

5 FURTHER DEVELOPMENTS

5.1 IRItaly CD-ROM VERSION 2

The IRItaly Project has already begun work on two fundamental tasks for the resolution of several of the issues illustrated in this paper. The first regards the release of a new version of the CD-ROM, which will contain a full implementation of Python FLAG.

According to the Project Documentation, FLAG was designed to simplify the process of log file analysis and forensic investigations. Often, when investigating a large case, a great deal of data needs to be analyzed and correlated. FLAG uses a database as a backend to assist in managing the large volumes of data. This allows FLAG to remain responsive and expedite data manipulation operations.

Since FLAG is web based, it is able to be deployed on a central server and shared with a number of users at the same time. Data is loaded into cases which keeps information separated. FLAG also has a system for reporting the findings of the analysis by extensively using bookmarks.

FLAG started off as a project in the Australian Department of Defence. It is now hosted on sourceforge. PyFlag is the Python implementation of FLAG - a complete rewrite of FLAG in the much more robust python programming language. Many additional improvements were made. Some of the most obvious features are:

- Disk Forensics
    - Supports NTFS, Ext2, FFS and FAT.
    - Supports many different image file formats, including sgzip (compressed image format), Encase's Expert Witness format, as well as the traditional dd files.
    - Advanced timelining which allows complex searching
    - NSRL hash support to quickly identify files
    - Windows Registry support, includes both win98 variant as well as the Window NT variant
    - Unstructure Forensics capability allows recovery of files from corrupted or otherwise unmountable images by using file magic
- Network Forensics
    - Stores tcpdump traffic within an SQL database
    - Performs complete TCP stream reconstruction
    - Has a "knowledge base" making deductions about network communications
    - Can construct an automatic network diagram based on TCPDump, or real time

- Log analysis
  - Allows arbitrary log file formats to be easily uploaded to database
  - GUI driven complex database searches using an advanced table GUI element

The ultimate objective is to integrate PyFlag into IRItaly's CD-ROM, in order to provide first responders with a tool that can guarantee a minimum of correlation that is significantly broader than that offered by the current version.

5.2 INTERNAL TOOL VALIDATION PROCESS

This remains one of the most pressing problems in digital forensics. The validation process that the IRItaly Project is seeking to complete offers as a deliverable a checklist of tools that comprise the daily toolset of a forensic investigator, according to master documents in the literature. The ultimate purpose of this deliverable is a checklist to ensure that the tools used are state-of-the-art. The priority is to guarantee, with the use of the tools described above, a minimum of compliance with best practices and a solution to the problems of integrity and security defined in Section 2. This is currently not possible since the issues expressed in Section 2 regard the acquisition phase and not the analysis phase, which is essentially done off-line with the tools cited above.

6 CONCLUSIONS

The objective of this paper is to act as a tutorial for log and event correlation. To ensure that the operations comply with the general principles of digital forensics, the tools used have to meet a series of requisites. The IRItaly Project is currently seeking to achieve precisely this objective. At the moment, the most important problems to resolve are the manageability of distributed architectures, with particular emphasis on top-down and real time approaches. We currently see a gap between the two approaches, which are pursued, respectively, by ISVs and by the GPL world. The latter is famously less well financed than the former, and for this reason cannot use the same methodology. In any case, the hope is to guarantee a minimum of autonomy to those operators who are not able to invest large sums in complex distributed systems.

7 REFERENCES

Abad, Taylor et al. Log *Correlation for Intrusion Detection: A Proof of Concept.* Proceedings of ACSAC 2003, United States

Chuvakin Anton: *Advanced Log Processing* , www.securityfocus.com

Casey, Eoghan, *Network Traffic as source of evidence: tool, strenghts, weaknesses and future needs,* Digital investigation Journal, Elsevier Science Group, Feb 2004

Boyd P and Fortsterm P: *Time and Date Issues in Forensic computing-a case history:* Digital investigation Journal, Elsevier Science Group, Feb 2004

Forte, Dario, *Analyzing the Difficulties in Backtracing Onion Router Traffic*
IjDE 2002 1:3

Girardin L. et al: A *visual Approach for monitoring logs. LISA 1998 conference* proceedings www.usenix.org/publications/library/proceedings/lisa98/full_papers/girardin/girardin.pdf –

The IRItaly Project and *The Italian Honeynet Project.* www.honeynet.it

Schneier B, and Kelsey J: *Secure Audit Logs to support computer forensic*: www.schneier.com

The Honeynet Project KYE series of papers: www.honeynet.org

8 ABOUT THE AUTHOR

Dario V. Forte, CFE, CISM, has been active in the information security field since 1992. He is 35 years old, with almost 15 years experience as a police investigator. He is a member of the Computer Security Institute of San Francisco/Usenix and Sage. His technical articles have been published in a host of international journals and he has spoken at numerous international conferences on information warfare. As an infosecurity analyst, Dario has worked in the public, government and corporate sectors, and is also involved in the Information Security Project on the international level under a non disclosure agreement. He teaches classes and holds lectures on information security management at universities and other accredited institutions worldwide. He has over ten years' experience working with international governmental agencies such as NASA, and the US Army and Navy, providing support in incident response and forensic procedures and has resolved many important hacking-related investigations. He has lectured at the Computer Security Institute, the United States D.H.S. and D.o.D., the Blackhat Conference, the DFRWS (US Air Force Rome Labs), and POLICYB (Canada). Dario has given interviews with Voice of America, Newsweek, the Washington Times and CSO Magazine. He provides security/incident response and forensics consulting services to the government, law enforcement and corporate worlds.