# Improving evidence acquisition from live network sources

by Bruce J. Nikkel
nikkel@digitalforensics.ch

Originally published by Elsevier in Digital Investigation
The International Journal of Digital Forensics and Incident Response
Vol. 3, No. 2 (doi:10.1016/j.diin.2006.05.002)

June 29, 2006

**Abstract**

The pervasiveness of network technology is causing a shift in the location of digital evidence. What was once largely found on individual disks tied to single individuals is now becoming distributed across remote networked machines, under the control of multiple organizations, and scattered over multiple jurisdictions. The network interactions between these machines are also becoming recognized as a source of network evidence. These live network sources of evidence bring additional challenges which need to be addressed. This paper discusses these issues and suggests some improvements in the methods used for the collection of evidence from live network sources.

**Keywords:** Network forensics, Live network evidence, Live network acquisition, Live network forensics, NFAT

# Contents

# 1 Introduction

The pervasiveness of network computing is driving the need to formalize network forensics. Questions of evidence authenticity, reliability, preservation, admissibility, tool testing and verification, etc., all need to be addressed before live network forensic evidence can be accepted with a similar degree of confidence as storage media forensics. This paper concentrates on improving one piece of this greater research area, the acquisition of data from live network sources.

The area of network forensics encompasses many sources of evidence including captured network traffic, data collected from remote network services, intrusion detection system (IDS) logs, logs from network infrastructure devices (routers, switches, firewalls, etc.), and logs from servers providing network services. Of these sources of evidence, captured network traffic and data collected from remote network services both share many similarities and can be generalized into one category of *live network evidence sources*[1]. This paper uses this generalization and discusses improvements to various areas of the live network evidence acquisition process. It builds on previous work in the use of captured network traffic as a source of evidence[2], as well as previous work done in collecting evidence from remote network services[3][4].

Some of the difficulties of collecting and using live network evidence from untrusted networks such as the Internet, include issues of authenticity and reliability. Source Internet Protocol (IP) and Media Access Control (MAC) addresses can be spoofed, various protocol headers can be faked, and the content transmitted can be fabricated. Network traffic can be intercepted and modified during transit. The source of network packets, connections, or messages can also be hidden using various anonymizing techniques (mixmaster/cypherpunk remailers, onion routing, web anonymizers, proxy chains, etc.). Certain forms of anti-forensic activity may also hamper the collection of evidence from live network sources. These all create potential difficulties in trusting evidence collected. Several methods can be introduced which improve the reliability of evidence collection, as well as improving the detection of untrustworthy evidence.

It is important to emphasize that the concepts presented here are improvements to current methods of live network acquisition, and do not guarantee the collection of authentic and reliable network forensic evidence. They do, however, take steps to reduce the risk of collecting inaccurate data and help to evaluate aspects of evidence certainty. Also specified are some practical guidelines for the collection process, including the preservation and handling of evidential data during and after collection.

## 2 Collector location

There are a number of issues involving collector location which have an impact on the collection of evidence. In some cases, strategic positioning of a collector will allow for more complete or more accurate collection of evidential data, in other cases, the positioning of a collector could affect the authenticity and credibility of collected data.

### 2.1 Minimizing distance to source

Both physical and logical network distance can be viewed in multiple ways. Physical distance might refer to the classic straight line between two geographical locations, but it could also refer to a physical length of installed cable, such as a telco leased line or physical Ethernet segment[1]. Logical distance could be measured in network hops such as a TCP/IP network, a path of Border Gateway Protocol (BGP) Autonomous Systems, or a sequence of Message Transfer Agents (MTA) used during email delivery. It could also be measured in terms of logical network segments, as in a bridged Ethernet environment. A further, more abstracted, example of logical distance could be the number of software components in a complex IT infrastructure (browser ⇒ webserver ⇒ application ⇒ database). These representations of distance in computer networking all share a common theme regarding the collection of evidence. Minimizing the distance between the investigator and the evidence is advantageous[2].

The effect of distance on the collection of evidence can have an impact on the error rate. For example, increasing distance in a wireless network will cause a decrease in the signal quality. Increased distance may also affect overall network latency, possibly causing timeouts and errors at other layers (where a collection tool might be in use). In general, the greater the distance between the investigator and the evidence, the greater the potential for error during acquisition of that evidence. Seminal work on analyzing error in network forensics can be found in [5].

Distance can play a role in the completeness of data collected at various network layers. For example, a network traffic capturing device only needs to be one hop away from a segment before it is unable to collect that segment's non-routeable traffic. The ability to collect lower level information, such as a MAC address for example, is also be affected by the distance. If a collector is separated from an offender by a router, such link layer information is lost. If a capturing device is one or more hops away on a redundant network, some

[1] Measured using time domain reflectometry

[2] In some cases encryption can mitigate certain adverse effects of distance. This is discussed in a later section.

routed traffic may take a different network path, bypassing the collector altogether. This effect can also be seen with other definitions of 'distance'. For example, a more complete set of data might be extracted with direct SQL access to a database rather than through a restricted web interface. In this logical sense, the 'closer' an investigator is to the source of the evidence (the database in this case), the more complete the acquisition of data might be.

Distance can also affect the accuracy and authenticity of evidence collected. For example, as the number of network hops increases, the potential for man-in-the-middle activity also increases. Conversely, if a collection device is connected directly to another machine with a crossed Ethernet cable, malicious man-in-the-middle activity would require physical access to the cable.

### 2.2 Issues with firewalls, proxies and address translation

In discussing the location of collectors on the Internet, the connection itself plays an important role. Firewalls and proxy gateways have several issues and shortcomings when used in the collection of digital evidence.

Collection devices using reserved IP ranges such as 10.0.0.0/8, 192.168.0.0/16, or 172.16.0.0/12[6] are not routeable on the Internet, and do not exhibit the property of global uniqueness. The routeability and uniqueness of such addresses can only be guaranteed within the scope of the local subnet or private intranet. In some cases, such as the promiscuous collection of traffic, this may not be an issue. However, when collecting evidence from remote network sources such as distant websites, or ftp servers, etc., this could cause certain difficulties unless Network Address Translation (NAT) is used.

Proxies and port forwarders are another potential source of problems for certain types of live network evidence collection. This is especially the case when the collector itself is behind a proxy or firewall, and collecting evidence from remote systems such as websites or FTP servers. In some cases, such devices prevent certain pieces of network evidence from being collected. In other cases the evidence could be modified, stale (from caching) or even destroyed (consider malicious code which a filtering proxying might remove from a webpage before passing it on to a protected client). Some proxies (SOCKS for example) pass the upper layer content unmodified, but any interesting information below this layer is lost.

Firewalls are an expected component when connecting to the Internet today, but when configured too restrictively they may block certain infrastructure traffic which may be of interest. There may be certain blocked

Internet Control Message Protocol (ICMP) types which could provide more accurate information about the state of a remote service being collected, there might also be connection attempts by external machines which might be interesting, but blocked or dropped by a local packet filter.

In addition to issues surrounding collector placed behind firewalls or proxies, the reverse situation is also possible. A remote machine containing evidence may be situated behind a firewall or proxy as well, obscuring or masking relevant information. In cases where malicious activity is taking place, an offender may be covering their tracks using proxies or portforwarders (SOCKS or netcat for example) placed to hide their original IP address. In such cases, the investigation of the observed IP address could lead to a compromised PC and possible implication of the wrong individual.

### 2.3 Multiple corroborating collectors

Issues of evidence provenance and corroboration have been discussed in the past[7]. In an effort to improve the collection of evidence from untrusted network sources, particularly on the Internet, its is useful to have multiple collectors corroborate on collected evidence. This provides increased reliability, more complete data collection, and in some cases reduced error. It is here where the concept of the acquisition window[1] becomes useful. Consider two independent parties collecting evidence from the same live network evidence source, during the same acquisition window. The evidence collected may be considered independently verified, similar to third party verification of seized hard disks[3].

As an example, consider a directory on a remote ftp site containing illegal material during a certain period of time. Evidence of the material's existence must be shown. The acquisition window may be several days. During this acquisition window, two independent parties collect evidence, selectively limiting or filtering collection to the IP address and the directory containing the illegal material. After collection, both parties show identical sets of collected evidence. This corroboration shows independent verification of live network evidence collection.

A second example illustrates a scenario of evidence capture from an Ethernet segment. Consider an attacker compromising a local server on a network. Evidence of the attack taking place must be shown. The acquisition window of the attack as a whole may be several days, but the acquisition windows of individual network packets are only several milliseconds during network

[3]Note: this does not verify the authenticity of the data. Both parties may still have collected spoofed or otherwise fabricated data.

transfer. During these acquisition windows, two independent parties collect evidence through promiscuous packet capturing, selectively filtering for any traffic exchanged between the IP address of the server under attack, and the IP address(s) of the attacker. After collection, both parties show identical sets of captured network traffic. This corroboration shows independent verification of evidence collection.

Multiple corroborating collectors may also be used to acquire a more complete set of evidence, such as related activity taking place over different network paths, or from multiple origins. In addition, certain deceptive behavior may be revealed when several collectors see different views or different parts of the same network. For example, consider an attack which is orchestrated in successive steps from different IP addresses. An intruder may exploit a vulnerability from one IP address, and then connect using another IP address to continue the attack. If an IDS only detects the first IP address as malicious, information about the second IP address may be lost. Multiple collectors may help correlate a more complete picture of an incident.

# 3 Evidence integrity during and after acquisition

There are a number of important principles in computer forensics which serve to protect the integrity of collected evidence. Achieving similar principles within the realm of live network acquisition is less straight forward, but certain measures can still be defined.

## 3.1 Imposing layered and filtered write-blocking

The idea of a layered approach to network write blocking has been discussed in the past[1]. The basic principles deal with enforcing a write-blocking property at individual layers of the network, instead of the network connection as a whole. While much work has been done to ensure total read-only network access (one-way Ethernet cables, stealth mode operation, for example), in many situations, a certain amount of network 'writing' is necessary to collect evidence.

The concept of imposing layered write-blocking to the acquisition process involves collection tools which can demonstrate non-modification of evidence on the network layer at which they operate. An example of such a tool might be a read-only File Transfer Protocol (FTP) client which does not have the ability to upload, delete, or otherwise modify data. If a device or tool can demonstrate that no writes or alterations have been made to collected data at a specific network layer and within the confines of specific filter criteria, then the evidence collected can be considered unmodified by the collection process.

It is important to note, that such 'read-only' tools may not work in every situation involving evidence acquisition from live network sources. Consider the simple Hypertext Transfer Protocol (HTTP) GET command. Depending on the HTTP server configuration, requesting a Uniform Resource Identifier (URI) may invoke server-side scripts which could cause remote evidence to be modified.

### 3.2 Maintaining integrity of collection device

The subject of system hardening and integrity has been studied in depth both in IDS and computer security. There have also been calls for secure evidence collection systems in network forensics literature[2]. Of primary interest here is making a distinction between typical hardened/secured systems, and network forensic collectors. There are a number of special attributes of forensic collectors which may go beyond the requirements of a standard hardened system build.

The basic requirements of a typical hardened system include such items as minimal services/software running, defensive configuration, local packet filtering, integrity checking, logging and alerting, strong user/admin authentication, etc. These are all valuable requirements which can be reused for building network forensic collectors (especially collectors which will be located on untrusted networks). Additional qualities and requirements which may have an impact on the evidence collected and the forensic process are as follows:

- precise Network Time Protocol (NTP) synchronization and provisions for timestamping collected data
- safe collection tools, resistant to malicious or corrupt collected data
- strict control or knowledge of collector traffic written to the network
- provisions for signing and/or hashing collected data
- provisions for encrypting sensitive collected data
- showing validation or correct operation of collection tools
- network location stamping
- additional logging requirements
- legal/regulatory or organizational policy issues which may place certain technical restrictions on collection

These requirements provide additional functionality for preservation of evidence and integrity of the collection process.

### 3.3 Integrity and protection of collected data

Principles of evidence preservation and chain of custody can be applied to collected network forensic evidence in nearly the same way as traditional computer forensics However, there are several additional factors to consider, especially when promiscuously capturing data from a live network.

Typical forensic acquisitions include the cryptographic hashing of all acquired data. This is commonly achieved using standard algorithms such as MD5 or SHA-1[4]. Recording the cryptographic hash of collected data at (or during) the time of acquisition will indicate if the evidence has been preserved over time.

When collecting data from live network sources, especially within certain organizations, maintaining the confidentiality of collected data may be critical. The collected data may contain information which is protected or restricted by local or regulatory law. Within organizations, it may also be subject to additional information security policies. In such cases, measures need to be taken to ensure compliance. This could take the form of strict selective filtering, or encrypting files or filesystems containing collected evidence.

In certain cases, segregation of duty regarding collected network evidence may be desired. If the team collecting the data is different from the team analyzing it, it maybe unnecessary or undesirable for the collection team to have access to large amounts of previously collected data. In other instances, it may be necessary to begin data collection before a formal authorization process has been completed. Under such circumstances, it may be possible to asymmetrically encrypt the data as it is being saved. With separate acquisition and analysis teams, decryption keys could be made available only to the analysis team, preventing the collection team from accessing previously encrypted data. In the cases where prior approval is required for handling captured network traffic, decryption keys for previously collected data could be held by the body granting authorization. When needed, decryption keys could be handed over as part of a formal approval process. Such an asymmetric encryption process would also preserve the integrity of evidence from the moment of acquisition. Useful work done in securing logs from attack[8] and other similar ideas might be adapted to suit this purpose.

[4]MD5 and SHA-1 are commonly used cryptographic hashing algorithms

# 4 Acquiring a complete set of data

The difficulty in collecting a complete set of data from live network sources requires an approach different from traditional storage media forensics. Kenneally and Brown[9] argue that partial acquisition of a source can also be used as valid evidence. When acquiring data from live network sources, this selective collection is indeed the only feasible method.

## 4.1 Difficulties with live network acquisition

A collector capturing traffic on a high bandwidth network may have difficulty maintaining the required performance for acquiring a complete set of packets. A collector acquiring data from network sources such as remote websites or FTP servers may be restricted to acquiring only that which has been made publicly available. Storage capacity is another obvious factor in collecting large amounts of live network evidence. High-speed networks (Gigabit Ethernet, OC-24 speeds and higher, etc.) cannot realistically be captured in their entirety beyond a certain length of time, and large repositories of remotely accessible data may be impossible to store on a local forensic collection device.

## 4.2 Selecting data to be retrieved

To perform selective acquisition from a live network source, a set of well defined boundaries specify precisely what will be collected or ignored. These boundaries can be implemented as a set of technical filters which collection tools may use in performing a selective acquisition. When using selective collection methods, the term 'complete acquisition' then applies only to the scope of the defined boundaries, not to the entirety of available data.

In defining precisely what to collect and how to specify technical filters, it is useful to determine what evidence may be required for later presentation, or for furthering an investigation. Enough data should be collected in order to show:

- the existence of evidence during transit between one or more nodes
- the existence of evidence available from a remote network service
- self-consistency of underlying protocol data
- due diligent collection by the collector and technician
- any additional corroborating data to support authenticity or reliability

### 4.3 Retrieve multiple instances/sources of data

In a modern networked environment, issues of performance and availability typically result in multiple instances or sources of information. This can be useful in forensic collection, in particular, finding authoritative data, reconstructing past instances of data, and providing additional corroboration. Practical investigation of complex web architectures and Internet sites has been been previously documented[10][11].

When collecting data from remote network services, multiple instances of data typically take the form of multiple servers with replicated or mirrored data. To ensure completeness of acquisition, all known and accessible sources of data should be identified and acquired. These can be later analyzed and used to judge authenticity.

Traffic in transit across multiple network segments presents the opportunity for multiple or distributed collectors to acquire more than one instance of evidence. Corroborating evidence collected from multiple points on an internetwork can be useful in judging authenticity.

### 4.4 Including protocol and out-of-band activity

When collecting data from live network sources, there may be certain protocol and out-of-band information which may help provide a more comprehensive description of the evidence collected and the source(s) from which it came[5]

Protocol information such as IP, TCP, or HTTP headers, can all be collected and used to provide useful investigative information for a case, or to help verify consistency and authenticity. This information also makes the technical details of the acquisition process available, and can show the due diligence of the investigator or forensic technician in adhering to accepted methods.

The collection of low level protocol information and other out-of-band data can be accomplished in a number of ways. The most comprehensive method involves promiscuous packet capture[6] of all investigator activity during an evidence collection session. Individual collection tools may also have additional logging or increased verbosity which can be enabled to preserve more technical details of a particular activity.

[5] The expression 'out-of-band' is used here to describe related information found outside or surrounding the channel or protocol containing the evidence being collected.

[6] Using tools such as tcpdump, ssldump, tcpflow, etc.

Various network layers provide name-to-address mappings which should be recorded along with the raw protocol data captured. Some examples of name-to-address mappings include:

- Domain Name System (DNS) records or Hosts tables which map hostnames to IP addresses
- Address Resolution Protocol (ARP) tables which map nodenames and IP addresses to MAC addresses
- UID and GID[7] numbers which map to user and group names
- TCP and UDP port numbers which map to network services
- ICMP types or codes which map to particular messages
- Application specific name-to-number identifiers

Whenever such mappings resolve between two pieces of information, this could be significant in furthering an investigation, or in judging the authenticity of evidence. These mappings may also provide additional information when examining the reverse mapping. For example, the reverse lookup of a website's IP addresses, is not always the same website Fully Qualified Domain Name (FQDN), and could provide useful information about the organization hosting the server.

# 5 Documenting live network acquisition process

Documenting the various aspects of digital forensic acquisition helps create a qualitative description of the acquisition as a whole. It describes such things as:

- due diligence on the part of the forensic technician or investigator
- adherence to accepted methods and procedures
- precisely what was collected (or in some cases, not collected)
- start/end timestamps
- additional technical information (lower level protocol information or headers, for example)
- errors and lost/corrupted data
- other meta information such as the investigator, case ID, case/evidence descriptions, etc.

[7] Operating System User and Group Identifiers

When building digital forensic tools, much of the documentation process can be integrated and automated, allowing for copious recording of various technical details. This is especially useful when collecting data from live network sources. Having additional technical detail may help reveal inconsistencies and issues with both the data collected, and with the collection process itself.

There are a number of advantages in recording a detailed technical description of a network forensic acquisition. These might include:

- showing that the collector existed as a unique and valid node at a certain layer of a network during the acquisition process
- more accurately reconstructing a sequence of events
- precise recording of what data was acquired and what was ignored
- precise recording of network writes or modifications made at each network layer during collection
- determining any ill effects the collection method may have had on evidence collected
- providing additional technical information which might be corroborated with other sources (such as infrastructure logs, ISPs, other 3rd parties, etc.)
- provide enough technical detail to reveal problems or raise suspicion regarding the authenticity and reliability of evidence collected.

## 5.1 Documenting network writes and filtering

In many cases, a certain amount of network 'writing' must be done in order to collect data. This may take place in the form of a handshake or setup protocol, or a query to request data from a remote network service. Writing to the network must be documented and any effect the network 'write' had on the evidence being collected must be shown.

When acquiring data from remote network services, a significant amount of network writing takes place. Consider the minimal network writing needed to get a simple webpage:

- Link layer: ARP requests sent to determine the MAC address of the router, plus all other network layer traffic sent out in encapsulated in Ethernet frames
- Network layer: all transport layer traffic sent out inside IP packets

- Transport layer: DNS query packets sent to nameservers, TCP packets sent in setting up, maintaining, and terminating the session with the webserver.
- Application layer: HTTP request headers sent

Modern operating systems may inject other network packets onto the network for other purposes. These could be for dynamic configuration and network service detection, auto checking of software updates, or software components which do network polling. This traffic should be controlled if possible, or at least documented.

A useful method of documenting network writes when collecting evidence from remote network services is a local promiscuous packet sniffer. Capturing all the traffic generated by the investigator activity during the acquisition procedure ensures complete recording of all writes to the network. Individual tools for collecting data from remote network services may also have comprehensive built in logging capabilities which record collection activity.

Recording this level of detail while collecting remote evidence allows for closer scrutiny of the evidence collected. It shows the precise behavior of the tools used to collect the data. An accurate picture of the network writing as well as the filtering can be determined.

### 5.2 Time stamping

The recording of accurate timestamps is important when collecting evidence[12]. Live network environments are in a constant state of change. Establishing precise timestamps allow more accurate sequential ordering of events and provide more accurate correlation with other independent evidence sources.

The granularity of timestamping evidence during collection can vary with the method of collection and the with the type of data collected. In some cases, timestamping individual captured IP packets may be required (individual ICMP or UDP packets containing a malicious payload, for example). In other cases, timestamping the establishment and termination of a TCP session may be sufficient (determining usage of various TCP services, for example). At higher application layers, a single timestamp signifying the completion of a certain transaction may be acceptable. The granularity of attaching timestamps to 'pieces of data' depends largely on the type and purpose of the evidence and the live network source from which it is retrieved.

The timestamps attached to data collected may be absolute or relative to a specific starting point. Evidence collection software relies on accurate system time to create these timestamps. The standard protocol for keeping correct system time is the Network Time Protocol (NTP) which provides time services from from various sources (Local system clock, GPS, DCF77 receivers, or higher stratum NTP sources). NTP is a relatively complex protocol which is able take into account various effects of network latency, system processing time, and hardware clock drift. Synchronizing to a number of time sources will ensure accurate timestamps on collected data. Failing to keep a collector's time synchronzed will make it difficult or impossible to correlate events and network traffic accross large numbers of machines over precise intervals of time.

### 5.3 Location stamping

When collecting evidence from a live network source it may be important to demonstrate that the collection device existed as a valid node at some layer of the network during the time of acquisition. This can be shown by recording the network identifying details at the appropriate network layers. For example, on the Internet the identifying location information may consist of the following sets of information.

Physical location details:

- geographic address where acquisition is taking place (City, Street address, etc)
- location of collector (floor, data center/server room, office or cubicle no.)
- point of attachment details (patch panel details, rack no., hub/switch no., hub/switch/port number, etc.)

Link layer details:

- OUI[8] unique MAC address
- hub/switch logical details (logical port number, segment name/number , VLAN info, etc.)
- other lower layer access/attachment information (PSTN dial-up numbers, ISDN PRI/BRI details, etc.)

Network layer details:

- Valid, routeable IP address

[8]Organizationally Unique Identifier

- Subnet details (netmask, broadcast)
- Routing details (default router or routing protocol used)
- DNS info (forward and reverse)
- Whois info, Autonomous System (AS) number

If evidence is being acquired from a particular remote node on a network, the location information for that node should also be collected (where possible). In addition, the network path to the remote node should be documented (traceroute). A systematic approach to collecting this information is documented in [11].

If evidence is being collected from application layer 'networks' such as peer to peer services, chat services, etc., any abstract location or identification meta data should also be recorded.

### 5.4 Reporting errors

Previous work in measuring error in network forensics is found in [5]. While error correction is built in to many network protocols, errors may still occur at other network layers. These errors may reveal themselves through ICMP messages, SNMP traps, or application specific errors such as timeouts, or input validation errors, etc.

It is important to distinguish between errors generated by the collection tool, and those errors generated by the live network source being acquired. A corroborating collector should notice the same errors generated by the live network source. However, errors generated by the collection device or tool itself will not necessarily match those of a corroborating collector. When recording these errors, it is important log the source of the error along with a timestamp. This information may be useful for various purposes at later stages of the forensic process. Implementing such error reporting on a collector may done through increased system logging, verbose logging of collection tool usage, or monitoring local Simple Network Managemnt Protocol (SNMP) and OS kernel statistics.

## 6 Analyzing authenticity and reliability of acquired data

The authenticity of data collected from the Internet has been denoted as inherently untrustworthy. This trust issue can be dealt with to a certain extent by analyzing certain details of the data collected. In particular, the

authority of a live network source, self-consistency of data, certain cryptographic properties of data, and the use of corroborating collectors, can all play a role in judging the quality of collected evidence.

## 6.1 Identifying authoritative sources

When multiple instances of a live network source exist, typically one instance contains the best or most updated set of data. Such a source is referred to here as a *source of authority*. Depending on the network service, these typically use such terms as master, primary, authoritative, main, etc. Other instances are typically described with terms such as secondary, slave, cache, replica, backup, standby, non-authoritative, etc.

When acquiring network data for forensic purposes, it is desirable to acquire data from authoritative sources. Secondary sources may also be acquired, and can act as a corroborating source or as evidence of previously existing content. However, not all non-authoritative sources may be useful for evidential purposes. For example, a web browser cache[9], corporate web proxy cache, or poorly updated server may contain stale or outdated evidence.

Some examples of network technologies which have an authoritative source are: DNS and Whois servers, Webserver farms, and Mirrored FTP repositories. Determining the authoritative source can be quite simple, for example, mirrored FTP sites typically publish the name of the master site. In other cases, such as DNS or Whois servers, finding the authoritative source can be found using simple tools[11]. When examining server farms, there may not even be a machine that is considered more authoritative than the others, as content might be pushed out to all from an internal (non-accessible) master. Useful work on examining load-balanced and mirrored sites can be found in [10].

## 6.2 Self-consistency/self-contradiction

The complexity of technology increases the amount of work and technical understanding needed to perform forensic acquisition and analysis of collected data. The consequences of technological complexity also make it more difficult for certain evidence to be plausibly faked or fabricated by malicious parties. When looking closely at the lower level technical details of collected evidence, self-consistency can be used to evaluate certain aspects of evidence authenticity and reliability.

[9] In some cases, if a site is already down, locally cached data may be the only evidence left.

The advantage of complexity is especially prevalent when dealing with live network evidence. Evidence collected from a network typically involves multiple network nodes, with interaction at multiple network layers, ordered over a certain duration of time. Any inconsistencies within this myriad of activity can be used in evaluating the authenticity or reliability of collected data. Self-contradiction of collected evidence may be indicative of tampering, falsification, or corruption[10].

For example, spoofed/faked emails may have certain inconsistencies in the Simple Network Mail Protocol (SMTP) header: Hostnames and IP addresses may not exist, timestamp ordering may be improbable, timezones may not match the server's geographical location, etc. Other examples could include an improbable Time To Live (TTL) field in an IP packet, or certificate details signed by a Certificate Authority (CA) which are inconsistent with the owner of a website. There may be a number of technical details which, if inconsistent with each other, could place the reliability of collected evidence in question.

The self-consistency of collected data may also show due diligence on the part of the investigator and the collection tools used to acquire the data. It may help show that critical data has not been altered during the collection process.

### 6.3 Independent corroborating sources

The authenticity of evidence collected from live networks may be supported using independent existing sources such as log files from various servers or other infrastructure devices. Some examples where corroboratory data may be found are as follows:

- Webserver, email, application logs, etc.
- Firewall and IDS logs
- Operating system and device logs (syslog)
- SNMP statistics on various infrastructure devices
- Network Management System (NMS) data such as SNMP traps or other alerts
- Traffic analysis logs such as netflow

[10] Note that in some cases the inconsistencies themselves may be the very evidence in need of collection

Evidence collected by independent third party organizations may also be useful in corroborating evidence collected from live network sources. This could be used as independent verification that the evidence was collected in a proper manner, and not modified by one of the investigative parties. Such third parties could be completely external to an organization (consultants or auditors), or independent units within an organization.

### 6.4 Cryptographic authentication

Traditionally, cryptography has worked against forensic practitioners, making it difficult to access evidential data. However, when determining evidence authenticity, cryptographic authentication can be advantageous.

The use of cryptography on networks typically serves to both encrypt and to authenticate data. Encrypted network traffic creates the obvious problems for investigators and has been well documented in the past [13][14]. However, cryptographically signed data offers a high degree of authentication which helps overcome many of the problems with live network acquisition. For example:

- shows non-modification of data during network transit
- authenticates the network origin
- provides a degree of non-repudiation
- shows completeness in collecting a set of data
- provides useful binding meta information (CA/Certificate details for example)

In the case of traffic promiscuously captured from networks, even if the data is protected, simply having a collection of signed packets may be enough evidence to prove certain activities with a high degree of certainty. When collecting evidence from remote network sources, possibly outside the control or jurisdiction of the investigator, authenticated information might be collected from SSL/TLS[11] encrypted websites (especially when the remote site uses a trusted CA certificate), authenticated DNS queries (DNSSEC), or any other remote network services which provide cryptographic authenticity. Using cryptography in this manner when collecting evidence on untrusted networks provides a significantly higher degree of authenticity. The risk of spoofing or man-in-the-middle attacks is greatly reduced. Evaluating the authenticity of the remote site and data is improved.

[11] Secure Socket Layer, Transport Layer Security

# 7 Conclusion

There are several conclusions that can be drawn from this work. Acquiring evidence from live network sources is complex, but can be simplified and accomplished in an organized and systematic way. Acquisition can be considered forensically complete within the boundaries of the selective filtering used. The authenticity and reliability of acquired data can be better judged or shown plausible. Detection of tampering is possible in many cases. If evidence has been cryptographically signed by a suspect, suggestions of tampering during network transmission can be proved false.

Methods for collecting evidence from live network sources are still incomplete and require further research. The concepts outlined here do not completely validate or invalidate the usability of such evidence. They do however suggest certain improvements in the collection process and add a certain level of due diligence to maintain. This will make evidence collected from live network sources easier to evaluate when using models such as Casey's scale of evidence certainty[5]. This may also provide additional guidance when developing network forensic collection tools.

# References


[1] Bruce Nikkel, Generalizing sources of live network evidence, Digital Investigation Vol. 2 No 3, 2005

[2] Eoghan Casey, Network traffic as a source of evidence: tool strengths, weaknesses, and future needs, Digital Investigation Vol 1 No 1, 2004

[3] Peter Sommer, Downloads, Logs, and Captures: Evidence from Cyberspace, Journal of Financial Crime, October 1997

[4] Peter Sommer, Directors and Corporate Advisors' Guide to Digital Investigations and Evidence, Information Assurance Advisory Council (IAAC), September 2005

[5] Eoghan Casey, Error, Uncertainty and Loss in Digital Evidence, International Journal of Digital Evidence, Summer 2002

[6] Network Working Group, Request for comments (RFC1597), Y. Rekhter, B. Moskowitz, D. Karrenberg, G. de Groot, March 1994

[7] Philip Turner, Digital provenance - interpretation, verification and corroboration, Digital Investigation Vol. 2 No 2, 2005

[8] Bruce Schneier, J. Kelsey, Secure Audit Logs to Support Computer Forensics, 7th Usenix Security Symposium Proceedings, 1999

[9] Erin E. Kenneally, Christopher L.T. Brown, Risk sensitive digital evidence collection, Digital Investigation Vol 2 No 2, 2005

[10] Angus M. Marshall, An Improved Protocol for the Examination of Rogue WWW Sites, Center for Internet Computing, University of Hull, 2003

[11] Bruce Nikkel, Domain name forensics: a systematic approach to investigating an Internet presence, Digital Investigation Vol. 1 No 4, 2004 No 3, 2004

[12] Chet Hosmer, Proving the Integrity of Digital Evidence with Time, International Journal of Digital Evidence, Spring 2002

[13] Eoghan Casey, Practical Approaches to Recovering Encrypted Digital Evidence, International Journal of Digital Evidence, Fall, 2002

[14] Jason Siegfried, Christine Siedsma, Bobbie-Jo Countryman, Chester D. Hosmer, Examining the Encryption Threat, International Journal of Digital Evidence, Winter 2004

## Document History

**Sept 2004 - Dec 2005:** Created original article
**Jan 2006** Submitted to Elsevier for peer review
**May 2006** Accepted for publication