A Method for Forensic Analysis of Control

Fred Cohen - CEO - Fred Cohen & Associates

President - California Sciences Institute

Abstract—*This paper examines technical underpinnings for the notion of control as identified in laws and regulations in order to provide a technical basis for performing forensic analysis of digital forensic evidence in cases where taking control over systems or mechanisms is the issue.*

Keywords- Turing capability, control, digital forensics; attribution

I. Introduction *and Background*

Laws such as [1] and [2] are used to assert that a criminal or civil violation has potentially taken place when a suspect acts so as to "knowingly and without permission" "access" a computer system and "intentionally cause damage", or similar language. Technical questions for the digital forensic evidence (DFE) examiner include, without limit, whether or not such a claim is (1) possible in a particular situation, (2) (in)consistent with the available traces, and (3) attributable to the suspect or others.

A. *Terminology*

Specific terminology varies in the statutes, but as general usage within this paper, we will use the terms; (1) violation to express the undertaking of a prohibited intent and/or act as defined by the applicable law, regulation, obligation, or duty; (2) trace to mean the bits produced as a result of the execution of finite state machines and found for examination; (3) event to mean a statement, record, or other non-trace that is used as part of examination, and other terms as defined in [16].

B. *Conditions on forensic examiner options*

1) Is it possible?

In cases where such claims are made, certain sorts of "access" may be required in order to "intentionally" "cause" certain types of "damage". The notion of causality combined with the notion of intent, appears to imply a sort of control, in the sense that there must be some way to express intent to cause the identified damage, and some way to have that expression of intent acted upon so as to result in such damage. If the basics of these premises cannot be shown, it cannot be reasonably asserted that such a criminal act is even possible. For example, if the consequence occurs before that expression of intent can be realized, the time ordering requirement of causality cannot be fulfilled, and if there is no manner in which to specify the intent in terms of the available language expressions granted for the level of access asserted to have been attained, the requirement of expressed intent cannot be realized.

The question of what can be ruled in or out of the possible in terms of damage, intent, and causality given access, is the subject of this paper.

2) Is it (in)consistent with the traces?

Given that the intentional causation of asserted damages is possible, a conviction/exoneration should be demonstrable if and only if the available traces are consistent/inconsistent with a path through an identified possible event sequence. If no such set of traces consistent with such a path are found, or if a cut to each such sequence is identified, then the suspect should be exonerated. The issue of trace consistency and inconsistency has been and continues to be addressed in many other papers [3][4][5][6] and will not be substantially covered in this paper.

3) Attributable to the suspect or others

Finally, it is necessary that the traces associated with an identified event sequence and consistent with the claims, be attributable to the suspect. To the extent that they cannot be so attributed, or to the extent that they can be attributed to third parties, again an exoneration is called for. The issues of attribution have been and continue to be substantially covered elsewhere [7][8][9][10][11][12] and will not be substantially covered in this paper.

C. *Basic computing concepts*

At the hardware level, digital computers are composed of electronic or other circuits that act based on the physics of devices to process signals in one form or another. These hardware mechanisms are designed and implemented so as to represent and operate on binary values (bits) only. The hardware that processes bits is often referred to as an automaton (plural automata); mechanisms that process sets of input bits into output bits, in parallel, and without storage. are called combinational logic; and the mechanisms that take input bits, combine them with a stored set of bits, and produce output while updating the stored set of bits according to a predefined method. In such finite state machines (FSMs), we may reasonably think of the predefined method as the "program" and the inputs and outputs as "data". The program interprets the data to update the stored state and produce the output.

1) Turing machines and Turing capability

There is a class of FSMs that implement a finite version of a Universal Turing Machine (UTM).[13] In a UTM, an unlimited and potentially infinite number of stored data states are used in conjunction with a predefined method. It has been mathematically demonstrated, that in such a machine, any FSM can be modeled using a single predefined method for all FSMs, with the details of the particular modeled FSM described by the stored data states at the time the UTM starts operating. Thus, except for limits on performance, any transformation of input sequence to output sequence that could be implemented in any FSM could be modeled accurately with any UTM. The notion that the stored states contain the description of the FSM is identified with the term "stored program computer" in that the memory states store the "program" that describes the FSM that the UTM is intended to model. The stored program is typically characterized by a set of "instructions" that take "input", act on "memory", and produce "output". The "instruction set" of the computer is the "set" of stored values that are interpreted by the mechanism to perform various "instructions".

Since the UTM itself is just a predefined method with an unlimited number of memory states, a UTM can contain a "program" that is, itself, a UTM. The term "Turing capability" is often used to describe this general purpose nature of a computer system. Notionally then, any computer with Turing capability can model any other computer with Turing capability, and this notion goes on recursively, except of course that in real devices, each recursion further restricts the actual available memory and performance.

2) *Special and general purpose mechanisms*

Not all FSMs and not all programs implemented in computer systems have Turing capability. When we discuss computer programs, we may differentiate between "general purpose" and "special purpose" in that general purpose mechanisms have a finite version of Turing capability, while special purpose systems do not. For the purposes of this discussion:

> **A general purpose** mechanism **CAN be programmed** so as **to perform any function** any that other computer can perform.
>
> A **special purpose** mechanism **CANNOT be programmed** so as **to perform any function** that any other computer can perform.

3) *Special and general purpose devices*

Many, but not all, digital systems, are general purpose computers in the sense that they have one or more central processing units (CPUs) that execute instructions from hardware and/or firmware-defined instruction sets in sequence, and where the sequence of instructions is dictated by stored digital values. Any function that any other digital system could do, albeit not as quickly or for select cases requiring more storage than they have available, may be done by these general purpose computers, because they have the finite memory and finite speed version of Turing capability. General purpose devices are programmable in the sense that the stored values can be changed, either by replacement of a hardware device or by setting of the values of those bits through software.

Special purpose digital devices also exist, and in large numbers. For example, many digital watches, input and output devices, and special purpose control mechanisms are special purpose in that they can never be programmed or altered so as to be able to perform general purpose computational functions. DFE can come from special purpose devices. In order to examine such traces, the examiner must know how the device works and where the traces were stored within the device. Based on that knowledge, the examiner may interpret the meaning of traces in terms of what the device does under what circumstances.

4) *Special and general purpose operating environments*

Many, but not all, general purpose computers use general purpose operating systems to control their execution. Typical examples of special purpose implementations are micro-controllers, such as the programmable logic arrays used for high-speed encryption and decryption, visual image processing, and so forth.

One of the most important examples of a special purpose operating environment is the sort of programmable logic controller used to limit the operation of devices in some infrastructure systems, mechanisms that control movements of physical devices, doses of radiation, and so forth. These devices are designed to allow complex controls over complex machinery, but are specifically limited in their programming and programmability so that they do not allow the physical devices to go outside of specific controlled boundaries, to move too quickly or too slowly, to hit themselves, or to create hazardous conditions for the rest of their environment. While such systems are increasingly being replaced by general purpose operating environments, they continue to be used and provide higher surety in exchange for less programmability.

Many devices, such as telephone switches, cellular telephones, mobile phones with base stations, copiers, printers, telephone answering machines, and so forth, have either special purpose processors or general purpose processors with special purpose operating environments. These operating environments and their associated hardware work together according to their implementation. The examiner has to know specific details of how they operate in order to meaningfully interpret the sequences of bits that form the traces associated with them, in the same way as special purpose devices require such knowledge. The same is true for many satellite control systems, systems within automobiles, aeronautical systems, and other similar embedded systems.

Some of these devices, and almost all personal computers, mainframe computers, minicomputers, and all other digital devices of the sorts in use today, use general purpose operating systems. They implement their functions through programs that execute from within the operating environment provided by those operating systems, system libraries, and the hardware devices in which the operating system functions.

5) *Special & general purpose programs and interfaces*

Many, but not all, programs provide special purpose capabilities at most or all interfaces. Typical examples of special purpose programs include Web servers, calendar applications, music and video recording and playback mechanisms, and most applications that users interact with on appliances, like telephones and ticketing systems. Typical examples of special purpose interfaces include menu systems and work flow systems, which provide interfaces that access underlying general purpose mechanisms but limit the interactions with those systems to limit user functionality. Most programs that provide general purpose interfaces are called "programmable" and typically have defined languages by which they are programmed. For example, the Lisp, Perl, C, Basic, and other similar languages are designed and intended to be programmed, and provide both general purpose functionality and the ability to implement special purpose interfaces.

II. The notion of Control

A. *Control theory as a structure*

The field of control theory is based on the notion that a signal sent into a control system causes an output to be affected in a predictable way. As an example, a transistor is a simple control system in which, within defined parameters and when used in a defined configuration, a signal at the base induces an amplified signal at the collector.

If there is feedback from the output to the input, so that the source of the input signal can adapt its input to compensate for changes in the output, the control system is called closed loop, while without such feedback, the system is called open loop.

Because of the effects of time, some control systems must react within limited time frames in order to prevent the overall system from ringing or going into other modes that cause out of control conditions. Such out of control conditions sometimes lead to physical damage to both the control systems and the systems and/or mechanisms under control.

These notions are typically applied to analog systems, and there is a substantial body of mathematics and engineering literature surrounding this area. As notions, they can be applied analogously to digital systems, but only by recognizing that the physics of digital systems differs from that of analog systems.[14]

In particular, and without limit, these differences include that digital systems have finite granularity in space and time, computationally-based equivalents of the speed of light in addition to physical distance limits on delays, discontinuities surrounding essentially every point in the discrete space, and far more complex mechanisms in terms of the numbers of "components" in a typical system.

Thus, while we may apply and use the notions and terms of open and closed loop control, the discussion of control in the context of this paper differs substantially from that of other areas of engineering.

B. *Mens rae and intent*

There is a distinction to be made between an actor who:

(1) through intentional acts over which they have control, knowingly and intentionally causes an event sequence to take place, or knows or should know that an outcome will take place,

(2) through intentional acts over which they have control, unknowingly or unintentionally causes an event sequence to take place, or

(3) through accident or events not within their reasonably anticipated control, unknowingly and unintentionally causes an event sequence to take place.

This is the concept of the difference between "mens rae", the guilty mind, and happenstance. In the technical world of digital systems, there are some specific issues that may shed light on these issues.

C. *Instructions versus intent*

Programmable digital systems are automated mechanisms that, when properly operating, execute the instructions given to them. While people may, at times, anthropomorphize computers to associate seemingly human properties to them, computers are not people, and cannot today be reasonably said to have intent in the sense that people do.

It is common parlance to use the word "command" when describing interactions with computers. The person types a command and the computer carries it out. Thus, it could be argued that any interaction between a computer and a human involves a human with intent issuing commands and a computer receiving and executing those commands. When people make mistakes in entering commands, or when the computer program interpreting the inputs does not properly interpret the intent expressed or intended by the human, the computer will nonetheless, act on the input it was given, without regard to intent.

FSMs interpret whatever inputs they get, and perform whatever functions they are intended to perform, based on the manner in which they are designed, implemented, configured, and operating.

> For **general purpose mechanisms**, the input **can express any intent** within the capability of any computer, **and** if properly expressed, **the mechanism will carry out that intent**.
>
> For **special purpose mechanisms**, no matter what the intent of the person ultimately responsible for the input, in normal operation, the mechanism **can only carry out the intent of the designers** as expressed by the implementation and configuration.

A special purpose mechanism can only carry out the intent expressed in the input to the extent that it is within the intended envelope of purposes of the designers and the constraints of the implementation and configuration. Thus a difference between general purpose and special purpose mechanisms is that:

> **General purpose mechanisms** are designed to **allow any intent** expressed in the input **to be carried out.**
>
> **Special purpose mechanisms** are designed to **allow only the intent of the designer to be carried out.**

D. *What it means to be in control*

As a fundamental assumption, this paper hypothesizes that in order for knowing acts to intentionally cause damage, the actor must have at least two things:

1. The ability to act so as to express intent.
2. The ability to have that expressed intent carried out.

The term "control" may then be reasonably seen in the notion that it requires that intent can be expressed and that expressed intent is carried out. That is to say:

- If a party cannot meaningfully express intent with regard to any specific act, they cannot reasonably be said to have been in control of that act.
- If a party's expression of intent with regard to any specific act is not or can not be acted upon, then they cannot reasonably be said to have been in control of that act.
- If a party can express intent with regard to a specific act, and if their expression of intent with regard to a specific act is or can be carried out, then they may be in control of that act.

Put another way, acts not expressible by the interface or not within the manifold of the expressed intent of the designer, cannot be reasonably said to be controlled by the user of that interface or mechanism. For example, and without limit:

- The owner of a computer who can configure, program, and operate it so as to perform acts of their choosing, may be in control over that computer and thus potentially attributed as the cause of its actions.
- A user of a Web server who uses the server in the normal way it operates and within bounds of normal and reasonable usage, cannot reasonably be asserted to be responsible for the space taken up by that server's logs of that user's use.
- An actor who exploits a vulnerability in a Web server to cause that server to act outside of the manifold of the expressed intent of the designer and operator, may indeed have taken control over that mechanism and thus be attributed as the cause for resulting actions.

In the first case, control was in the hands of the owner, and thus their acts could potentially be considered causal with respect to what the computer did with regard to the things they controlled.

In the second case, the user of a Web server using it in the normal manner, has no control over whether or to what extent that server logs or does not log their activities, because (1) there is no syntax by which such a user can normally express an intent to have their actions logged or not logged, and (2) even if there was a way to express such a thing, the normal logging of Web servers does not respond to user requests so as to enable or disable logging, unless they are somehow specially privileged to do so, and using an administrative interface.

In the third case, the normally special purpose interface to the Web server is bypassed by the actor, thus changing it from a special purpose interface and/or mechanism to a general purpose interface and/or mechanism. At that point, the actor gained control over the server, or some part of it, that was outside of the expressed intent of the designer and operator.

III. The Paths to Control

There are several readily distinguishable paths to control that are typically identifiable.

1. Authorization provides direct control.
2. Authorization provides indirect control.
3. Use exceeds authorization to gain indirect control.
4. Use exceeds authorization to gain direct control.

A. No control

The simplest case is one of "no control". No control may be demonstrated by (1) No ability to express the prohibited intent or act, or (2) No ability to have an expressed prohibited intent carried out. Such lack of control may be demonstrated, without limit, by showing that:

1. No implicit or explicit syntax is available at the interface or between the party and the mechanism to express intent or acts within the envelope of the violation.
2. Where such syntax is available to express intent or acts within the envelope of the violation, the mechanism does not carry out the expressed intent or acts.

In presenting results where theses cases have been identified with respect to the particular claimed violation(s), it is appropriate to state:

"[Party] had no control over [acts] claimed to cause [violation]. In particular, and without limit,

> (1) [Party] had no ability or mechanism to express any such intent to [mechanism].
>
> (2) [Party] had no ability or mechanism to have any such intent carried out by [mechanism].

It appears that [Party] did not cause [violation]."

In many cases, this limit on control can be shown, while in other cases, inadequate basis has been provided to demonstrate that this is not true. As an example, of each:

Suppose that the interface is the interface normally used for the use of a "get-only" Web server,[15] in which the only expressible syntax starts with "get " followed by the name of the thing to get and other elements of syntax, and further suppose that there is no claim or evidence found indicating that any function other than "get" was used or that any mechanism that caused the server to exceed its normal authorization or envelope of control is found. If the claimed violation is the knowing and intentional deletion of a file on the system operating the Web server, then the examiner might reasonably express:

> "It appears that [Party] had no ability or mechanism to express such intent to [Web server]".

Suppose that in a different case, the interface is a terminal interface that grants users the ability to execute arbitrary commands and write programs on a server, and further that the file asserted as deleted was accessible to a user logged in under a user identity not authorized for use by the [Party]. But in this case, suppose that it has not been shown that the Party ever in fact accessed this interface to this system. In such a case, the examiner can and should reasonably state that:

> "I have found no evidence to support the claim that [Party] had a way to express an intent to [delete a file from the server], because I have found no

> evidence that is consistent with [Party] ever having communicated to that server."

These examples show ways in which the "no control" conclusion may be arrived at and expressed, and of course there are other such paths to the "no control" conclusion.

B. Direct vs. indirect control

We will call "direct control" the use of explicit syntax that expresses intent to and carries out an action. For example, and without limit, a syntax allowing the user to "delete" something whose deletion is prohibited, and a mechanism in which such an expressed syntax is executed and in fact deletes the item whose deletion is prohibited, can be said to be under the direct control of the user expressing such intent at such an interface.

We will call "indirect control" a sequence of uses that individually do not cause an act to take place, but that, in the aggregate, cause the act to take place. An example of indirect control is the alteration of a number in a control file that specifies how often a log file should be removed so as to reduce the time between entries and removals to a small enough number that the file is deleted before it can be meaningfully used or backed up.

Direct control may be readily used to indicate an expression of intent. For example:

> "[Party] expressed the intent to delete the file by entering a command ordering the deletion of the file."

Indirect control is less obvious and presumably requires more information in order to tie the intent to the act. For example:

> "[Party] expressed the intent to delete the file by altering only the portion of the control file that controlled the retention time of the file and altered it so as to reduce the retention time to less than 1 second from it's original value of 30 days.".

Indirect control may be quite indirect, and as the extent of the indirection increases, establishing both the causal chain and the demonstration of intent typically become more and more difficult. For example:

> Suppose the traces indicate that in the process of deleting multiple files, all appropriately, that the control file also got deleted. As a result, at the next system startup, which occurred after the user issued a "restart" command, the default control file was used to replace the previously deleted control file, and the default control file had a retention time of only 30 seconds, as opposed to the 30 day retention time in the control file that was deleted.

In this case, it might be reasonably argued that the user made a mistake and that therefore, the act was not intentional, or perhaps even knowing. Unless it can be shown that the user had knowledge of the mechanisms to the point where they could reasonably be expected to anticipate that their acts would cause this result, and unless it can be shown that their act to delete the previous control file was not an accident associated with the other deletions, the cause for intent is hard to make. However, the case for indirect control is clearly available, if the cause and effect chain can be adequately demonstrated. Even in cases where the effect only happens with a limited probability, control may potentially be demonstrated if repetition of cause was used, or if different probabilistic methods to similar effect were expressed over time.

C. The limits of authorization

Authorization is particularly complex in that the intent of the party granting authorization is not typically accurately reflected in the technical ability to specify authority with available syntax. Many attempts have been made to define authorization schemes that more accurately reflect the sorts of things that organizations assert they wish to express, but the limits of expression involving computers and their inability to interpret human intent continues to be limiting. For these reasons, from a technical standpoint, the notion of intent is limited by what is expressed.

To the extent that the courts may deem that examiners should opine on intent not expressible in the syntax of the mechanisms in use, this represents a problem for the examiner, and with the exception of examiners having expertise outside of the areas of digital forensic evidence examination opining in other areas of their expertise, such opinions should not be rendered.

Technical authorization is typically expressed in terms of specific access control mechanisms. The mechanisms, whatever they may be, allow the operator of a mechanism to exert control over use. For example, and without limit, such mechanisms may grant access to devices, files, systems, or capabilities, based on identification and authentication, time of day, location, and any number of other parameters, including history and situation.

Such mechanisms are typically controlled by either local or network-based control mechanisms or by tokens, certificates, or similar informational mechanisms provided to users.

In determining authorization, the examiner typically examines all of the available traces to determine whether, and to what extent, activities performed by the Party were authorized, by determining whether the Party is accurately depicted by identities and whether the activities performed were consistent with the authorities granted to the relevant identities at the relevant times.

In cases where the examiner determines that use is in excess of authority, the questions of intent and mechanism become an issue.

1) Mechanism of unauthorized use

Because the technical mechanisms of authorization are or should be built so as to limit paths to bypassing them, from a technical standpoint, some protective mechanism must either be inadequately comprehensive and bypassed through an uncovered path (e.g., a sequence of authorized activities that lead to an unauthorized one, like a sequence of commands that enables access not otherwise granted), or inadequately implemented and bypassed by exploiting a weakness in implementation (e.g., an unanticipated input sequence like a very long input string that alters the stored memory of the program, causing execution of instructions not originally intended to be executed).

2) Use of uncovered paths

In the case of an uncovered path, intent may be difficult to demonstrate. For example, if in a menu system, the user uses a program that is in the menu, and from that program enters data that, because of similar syntax, executes a command that grants access outside of the menu system and results in a prohibited act, it may be difficult to show that the act was intentional. The examiner might reasonably express this result as:

> "The traces show that, using the menu system, a sequence of inputs similar to the normal inputs used in normal tasks was entered, except that [show the invoked sequence vs. the alternative]
>
> It appears that this act caused [violation (e.g., the deletion of the file)]"

But if they use the same mechanism repeatedly, or once they have command line access use that access to perform act after act, all outside of their normal scope of legitimate work, and repeatedly engaging in prohibited acts, that will tend to indicate violations. The examiner might reasonably express these results as:

> "The traces show that, in using the menu system, an uncovered path through the menu system was used, resulting in access to [command interpreter]. This path was used subsequently and repeatedly to issue commands in the syntax of [command interpreter] that [directly or indirectly] caused [violation(s)]."

To some extent, the difference between these acts may be characterized in terms of the "distance" between normal authorized use and the use detected. In cases where the distance is low, particularly cases where it represents only the distance associated with a single mistyped character, offset of hand position on a keyboard, or key press that executes a keyboard macro, the examiner might want to err on the side of prudence and limit their description.

3) Exploitation of weaknesses

When examination reveals apparent exploitation of weaknesses in the implementation of controls, the distance between normal use and exploitive use is typically very substantial. In effect, large portions of the input sequences differ between an exploitation and an uncovered path.

Exploitation of weaknesses also most commonly produces a situation in which the exploiting party gains Turing capability through an interface that is not designed to provide such capability, or in which access outside of the intended envelope of the design of the interface is granted.

In the case of exploitation of weaknesses, a particular mechanism should typically be identified and it should be demonstrated that the use of this mechanism produces the identified excess access. This typically involves one or more of a generic classes of mechanisms and specific detailed mechanisms that invoke those classes of mechanisms. For example, a generic mechanism of an input overflow[17] might be identified, and the specific instance identified from a published database of such exploitation methods.[18] If the particular exploit method has not previously been identified, the examiner might have to do a detailed write-up of the mechanisms involved.

To the extent the such mechanisms might remove or alter traces, the integrity and chain of custody of the traces and related indicators might also be subject to challenge.

The claim of exploitation of a weakness by an examiner should then be reported in terms something like:

> "It appears that [method e.g., an input overflow] identified in [specifics, e.g., Common Vulnerabilities and Exposures (CVE) entry 23.45.67] was used to exploit a weakness in [specific details of what the mechanism does to grant unauthorized access]. Using this access in excess of authority, it appears that intent was expressed to [acts] by [how expressed], and those intents were carried out by [mechanisms]. These acts resulted in [violation(s)]."

Such an explanation should include full details of the specific traces and events used to come to these conclusions and should address the provenance of these traces in order to properly couch these results in terms of the reliability of their basis. In a summary section of the report, the details are likely to be left out for clarity of presentation, but the details should be provided in a detailed section of the same report.

4) Intent

Intent is not generally demonstrated by traces of such activities, even though by the use of different expressions such as those indicated above, intent might be reasonably inferred by the trier of fact. In the latter case of the use of uncovered paths, the examiner might want to go a step further by stating something like:

> "It appears that this uncovered path was used to repeatedly express an intent to and in fact use access in excess of authority to [violation (e.g., delete files that otherwise would not have been deleted by his use)]"

Such a statement appears to be factually accurate given that there is apparently evidence of repetition and use of an interface "in excess of authority". However, to really lock down such a conclusion, typically involves additional information not found in these sorts of traces, and normally found in events.[16] Examples include, without limit:

- Expressions of intent in the form of written works,
- Documents showing that Party was notified of what to do in such circumstances and failed to do so.
- Documents showing that Party sought out the particular information required in order to perpetrate the particular violation.
- Admissions.

To the extent that other events can be tied to the traces, this may justify increasing confidence in the use of this indicator of intent, and to the extent that such a basis exists for the use of such language, it should be well and explicitly documented in the Examiner's report.

In the case of the use of an exploitation, there is normally little leeway for the examiner to ignore the issue of intent. While demonstrating intent to perform the specific acts associated with the violation (other than the exploitation itself) may not be entirely obvious, the use of the exploitation is hard to associate with anything but intent.

5) Whose intent?

Attribution is beyond the scope of this paper, but clearly this issue must be addressed in such claims. This is potentially complicated by exploitation of parties to the legal action or others not involved in the legal action but who may have acted. For example, and without limit, the accused party may have had their systems exploited or paths uncovered, and thus may not in fact have knowingly done anything violative, or intervening infrastructure may have been compromised resulting in violations that appear to come from a place they did not come from. Again, traces and events may be central to the demonstration of intent, and the process of examination may have to be pursued beyond the direct target of the violative act.

IV. The Envelope of Control

Notionally, when a mechanism is implemented, it is designed to perform a set of activities either within a designed envelope (special purpose) or within the envelope that encloses the mechanism (general purpose). In order to determine whether or to what extent an envelope of control may have been exceeded, the examiner must understand what that envelope is.

A. Determining what type of envelope is present

In general, it is infeasible to determine whether a given envelope of control is special purpose or general purpose. In the unlimited case, it involves showing that the envelope has or does not have Turing capability. Since this depends, ultimately, on whether or not the machine halts, it is undecidable for unlimited memory and time. While this is not strictly true for finite systems, the computational complexity of examining all possible execution paths for all possible input sequences and initial states is, in general, too complex for practical examination in substantial systems. This situation is greatly complicated by the fact that systems may interact, and the composition of components without Turing capability may have Turing capability.

While the general problem of differentiating special purpose from general purpose is infeasible, in fact, it is very often easy to determine that specific mechanisms are special purpose within specific contexts. For example, many of the Internet protocols are defined in request for comments (RFC) documents and have their syntax described by "Backus-Naur Form" (BNF) specifications.[19] Many of these protocols have explicit FSM descriptions with defined semantics. As such, they are readily analyzable and, in many cases, form special purpose automata.

When it cannot be determined or reasonably asserted that the environment is special purpose, it is reasonable and prudent to assume that the mechanism is general purpose for the envelope of control analysis.

B. Special purpose mechanism envelopes of control

For special purpose mechanisms, the examiner may determine the envelope control by identifying the set of all sequences of inputs and initial states and the set of all consequences of operating on the input sequences in the states in terms of the effects on next state and potentially violative effects.

While this may seem complicated, for most interfaces in widespread use, this is not as hard as it may seem. For example, the simple mail transfer protocol (SMTP) used to transfer email has only a few major states and well-defined transitions between them. As such, it is pretty apparently special purpose. Furthermore, the envelope of control granted to the user is pretty well defined. The user can request information of a few types, request the delivery of a message to one or more recipients, and provide the message itself. This can be repeated indefinitely. The set of side effects are fairly limited as well. The process takes limited time and memory to execute, and the resulting messages, if delivered, consume system resources to store. Log files may also be produced at a reasonably well-defined rate. Some such servers may also forward messages on to third parties, in which case this process is recursive, and the envelope of control extended to the interaction with other similar servers.

The next step in examination is to determine whether the violative act is within the envelope of control of the special purpose mechanism. To the extent that the issue at hand is, for example, the extraction of content from a different and unrelated database on the machine containing the SMTP server, this is not within the syntax specifiable by the SMTP protocol or the envelope of control of a typical SMTP server from the user (in this case external server) interface.

In cases where a violative act is within the envelope of control of the mechanism, traces are found that are consistent with the claim of violation, and no traces are found that are inconsistent with that claim, the examiner should report something to the effect that:

> "I found traces consistent with [violation]. In particular, within the envelope of control of the [mechanism], there is a capability to [perform acts] that are violative in that [reason they are violative], such acts typically produce traces of the form [form of the traces], and [traces found] were consistent with [Party] having undertaken [violative acts]"

To the extent that no such traces are found, that the envelope of control does not include acts that can result in violations, or traces are inconsistent with violations, the examiner should so state and rule out the potential for violation using these mechanisms.

In addition, the examiner should consider the normal expectation of operation. For example, for a Web server, the normal expectation of use can reasonably be said to include the retrieval of Web page content by making "get" requests at a rate commensurate with typical usage patterns. If the usage exceeds this pattern in a substantial way, such as by making large numbers of requests over short periods of time, it may be asserted to exceed authority, but not the envelope of control. An examiner might express something like this:

> "I found traces indicating that [number] requests for access were made within a period of [time frame]. Based on [basis and calculations or reconstruction], this volume of these requests and their associated responses could cause [effects] that would constitute [violation(s)]."

Caution should be used in making such claims, because assertions of a difference in quantity being a difference in kind without adequate factual basis and events supporting intent are likely to be problematic.

C. General purpose mechanism envelopes of control

For general purpose systems, the computations of the mechanism are limited only by the time and space they consume and the amount of time and space available to them within the larger environment they operate in. However, it is not only their computations that are at issue in legal matters. Regardless of the computations they may perform, their operating environments may be limited as to what they can ultimately reach, at least in a direct sense.

As an example, a virtual machine environment provides, in essence, all of the capabilities of its host machine with the exception of access to some portion of its performance, memory, storage, and input and output capabilities. Thus the envelope in which it operates is the envelope of the entire machine less the portions of the machine not made available to it.

In examining such an instance, the examiner should examine the outer envelope (that of the encapsulating machine) to determine the control and access granted to the inner Turing capable machine and determine whether and to what extent that envelope allows the violation to take place.

Of course, in such a situation, the envelope may be subject to unauthorized use or exploitation, and there may indirect as well as direct acts capable of causing violations. In such cases, the same analysis must be performed for the enclosing machine, and so forth, recursively.

In cases where a general purpose mechanism becomes available to the accused party, through whatever means, and if that mechanism allows that party to carry out acts that produce the asserted violations, and if the available traces have been found to support the claim that access to those mechanisms in a manner so as to carry out those acts is identified, and if none of the traces are inconsistent with that finding, then a reasonable conclusion might be expressed as:

> "I found that [Party] had access to [mechanism], that [mechanism] provides the means for [Party] to cause [violation(s)], that available traces are consistent with [Party] having undertaken those acts and those acts producing [violation(s)]."

To the extent that some of these things might not be true, this potentially weakens the claim. In particular, if any of the necessary elements are not found, they cannot be included, and if inconsistent traces are found, this conclusion would seem to be refuted. The opponent in such a case might, assuming these things are all true, reasonably say:

> "[Other party] identified no traces showing that [Party] actually accessed [mechanism] at the time in question, that [mechanism] actually caused any such [violation(s)], and has failed to provide traces that would normally be present if [Party] in fact did everything [Other party] claimed. Furthermore, other parties also had similar access in similar time frames, and they have not been shown to have not caused [violation(s)]."

Both of these statements can be true at the same time, and examiners on each side should, presumably, make such statements as are appropriate and back them up with the detailed basis.

D. Envelopes of control analysis objectives

The overall objective of determining the envelope of control is to determine whether or not the acts claimed are within the envelope of control of the party accused. To the extent that the envelope of control is not found to include either direct or indirect mechanisms with which the accused can cause the violation(s), the examiner should state something to the effect:

> "I found no method for [mechanism] to be used by [Party] so as to cause [violation(s)]."

If there is such a mechanism, the examiner should then seek to identify traces consistent or inconsistent with the claims that the accused party used that method in that mechanism to do so and state something to the effect:

"I found that [method(s)] could be used by [Party] within [mechanism(s)] to cause [violation(s)]."

followed by the appropriate selection(s) from:

- "I found traces inconsistent with [Party] actually using such [method(s)] in such [mechanisms(s)] so as to cause [violation(s)]."
- "I found no traces consistent with [Party] actually using such [method(s)] in such [mechanisms(s)] so as to cause [violation(s)]."
- "I found traces of other parties using such [method(s)] in such [mechanisms(s)] so as to cause [violation(s)]."
- "I found no traces of other parties using such [method(s)] in such [mechanisms(s)] so as to cause [violation(s)]."
- "I found traces inconsistent with other parties using such [method(s)] in such [mechanisms(s)] so as to cause [violation(s)]."
- "I found traces consistent with [Party] using such [method(s)] in such [mechanisms(s)] so as to cause [violation(s)]."

In cases where findings support [Party] having undertaken such methods in such mechanisms so as to cause such violations, the examiner should then proceed to try to further attribute such specific acts to that party.

V. Analytical Process and Presentation

The analytical process provides a systematic means for an examiner to determine whether or not an actor appears to have had adequate control to be said to have knowingly and intentionally committed a prohibited act. It ignores attribution and trace consistency issues, in favor of linking to results of examination in those areas. The analytical process is as follows:

For each identified potential causal path $\mathbb{P}$ from mechanisms in the control of suspect to a violation:

- identify $\mathbb{P}=(m_1, ..., m_n)\in M$, (a sequence of cause ($\mathbb{C}$)→ effect ($\mathbb{E}$) mechanisms that constitute $\mathbb{P}$).
- $\forall m\in\mathbb{P}$,
  - Determine whether m is general purpose or special purpose.
  - If m is special purpose, examine the syntax and semantics of m to identify direct or indirect means to affect asserted effects. If no such means exists, rule out $\mathbb{P}$, otherwise identify and document $\mathbb{C}\rightarrow\mathbb{E}\in m$.
  - If m is general purpose, examine the envelope of m (recursively), identify direct or indirect means to affect asserted effects. If no means exists, rule out $\mathbb{P}$, otherwise identify and document $\mathbb{C}\rightarrow\mathbb{E}\in m$.
- $\forall$ remaining $\mathbb{P}$, $\forall m\in\mathbb{P}$, identify traces probative with respect to $\mathbb{C}\rightarrow\mathbb{E}$ and search for such traces. If $\exists t\in T$ that are inconsistent with $\mathbb{C}\rightarrow\mathbb{E}$, rule out $\mathbb{P}$. Otherwise, if $\exists t\in T$ consistent with $\mathbb{C}\rightarrow\mathbb{E}$, confirm m. Otherwise, indicate that traces do not confirm or refute m.
- $\forall$ remaining $\mathbb{P}$, $\forall m\in\mathbb{P}$, identify traces probative with respect to exploitation or bypass of m and $\mathbb{C}\rightarrow\mathbb{E}$ and search for such traces. If $\exists t\in T$ that are consistent with exploitation or bypass leading to $\mathbb{C}\rightarrow\mathbb{E}$, identify possible alternative explanations of m. Otherwise, indicate that traces do not confirm identified alternative explanations of m.
- $\forall$ remaining $\mathbb{P}$, $\forall m\in\mathbb{P}$, attribute acts to Party.

Identifying and exhausting $\mathbb{P}$ is infeasible in most cases, so examiners identify feasible $\mathbb{P}$ by using their education, training, experience, skills, and knowledge.

A. The case for the accuser

For the accuser, to the extent that this process was undertaken, full details should be provided of each step in the process so as to adequately support the claims. To the extent that there are redundant paths by which to the claims may be shown, this is potentially problematic in that, presumably, only one thing actually occurred. Several conditions arise other than a perfect and fully completed analytical process. For example, the total set of procedures that can be performed with feasible time and resources is far less than the total number of possible procedures that could be performed for any nontrivial case.[6] Traces may not be available $\forall m\in\mathbb{P}$, and thus many remaining paths may exist, all of which may be feasible. Thus the case generally consists of statements of the form:

"It appears that [$\mathbb{C}\rightarrow\mathbb{E}$, ..., $\mathbb{C}\rightarrow\mathbb{E}$]."

with the basis for this appearance provided in the level of detail available.

The granularity of $\mathbb{P}$ is also potentially an issue. While $\mathbb{P}$ could potentially be explored at the level of each hardware component, it is normally examined at a far higher level. Typically, $\mathbb{P}$ is explored at the level of actions by the suspect with technical details supporting the claims of these actions at whatever level is required in order to identify relevant traces or events. Again, there are almost certainly details not explored, and the $\mathbb{C}\rightarrow\mathbb{E}$ sequence forming $\mathbb{P}$ is incomplete, even if it is reasonably convincing.

B. The case for the accused

Given $\mathbb{P}$, the examiner for the accused is responsible for identifying the limitations on, and identifying flaws in, the claimed $\mathbb{P}$. This typically comes in two forms. One form is identifying the limitations to the claimed $\mathbb{P}$, and the other is identifying alternative $\mathbb{P}'$ that demonstrate that the claimed $\mathbb{P}$ is not unique. To the extent that these are both done, it benefits the accused.

1) Identifying refutations of $\mathbb{P}$

To the extent that claims in $\mathbb{P}$ can be refuted, this is the strongest argument against those claims. While science might, in some cases, assert that a single refutation destroys such a claim, this is not always true. Because $m\in\mathbb{P}$ are not all purely mathematical in nature, and because all $\mathbb{C}\rightarrow\mathbb{E}$ are not precise, there may be cases when a refutation is imperfect. In addition, since such information is generally shown to triers of fact, a single refutation may not be adequately convincing, even if, as a scientific claim, it is compelling. A refutation might be stated something like this:

> "Based on [basis], the claim that [$\mathbb{C}\rightarrow\mathbb{E}$] for [m] is refuted and thus cannot be true. Based on the fact that [$\mathbb{P}$] depends on [m], [Other party]'s claims are inconsistent with the [traces and events] and, thus these claims cannot be and are not true."

Of course this is not always the case, and care should be exercised in going too far in such a refutation when there are alternatives available to the other side. Leaving it at the level of the inconsistency may be adequate in many cases.

2) Demonstrating alternative $\mathbb{P}$

A second, and less compelling course to countering claims is to identify alternatives that might just as well be true. For example, even though traces are consistent with $\mathbb{C}\rightarrow\mathbb{E}$, they may also be consistent with an alternative $\mathbb{C}'\rightarrow\mathbb{E}'$, in which the accused is innocent. A demonstration of alternatives might be called out something like this:

"[Other party] claims [$\mathbb{C}\rightarrow\mathbb{E}$], but many other possibilities exist and are consistent with [the relevant traces and events]. For example, and without limit:

- [$\mathbb{C}'\rightarrow\mathbb{E}'$] (e.g., Joe's brother was present in the room and had access to the same stuff)
- [$\mathbb{C}''\rightarrow\mathbb{E}''$] (e.g., Joe's wife, who is suing for divorce, and who previously ...)
- ..."

Identifying such alternative possibilities, even though this is not as strong as refutation of $\mathbb{P}$, it may be strong enough to cause a judge or jury to become unconvinced that the standard of proof has been met for the matter at hand. The strength of such alternatives, presumably, increases as there are more and more convincing alternative paths shown. Such paths tend to be strengthened when they are supported by traces and events, and when they sound reasonable to the trier of fact. For example, when such alternatives include motives for 3rd parties, traces indicative of 3rd parties, traces showing a lack of consistency with the other party's claims, and similar supporting details, they become far stronger and, ultimately, may be as compelling as refutations in the minds of the triers of fact.

3) *Identifying limitations in ℙ*

The least compelling, but still viable approach to countering claims is to emphasize the limitations in ℙ as presented by the other party. This includes identifying the limits on granularity and thoroughness, cases in which traces or events were not found and thus consistency could not be demonstrated, and situations in which traces found did not demonstrate consistency, even though they also did not show inconsistency. These may be enumerated in a statement something like this:

"[Other party] claims ℙ including {[ℂ→𝔼], ...}, however, and without limit (and as/if appropriate):

- [Other party] fails to provide a basis for the claim that [ℂ'→𝔼']
- [Other party] fails to consider low-level mechanisms that might refute the claim that [ℂ'→𝔼']
- [Other party] has not shown traces consistent with the claim that [ℂ'→𝔼'] (despite having traces that (might/would) so indicate if [ℂ'→𝔼'] were in fact the case).
- [Other party] fails to identify and examine additional mechanisms such as [list some of them] necessary for [path ℙ] to actually take place.
- [Other party] only performed a limited set of procedure(s) [P] and in failing to perform procedure(s) [P'] did not account for [traces and events] that might have demonstrated [Party's] innocence."

There may be many such limitations identified, and to the extent that these limitations are considered substantive by and meaningful to the trier of fact, they may carry enough weight to sway the trier of fact below the threshold required to find the accused guilty.

VI. Summary and limitations

While this paper has discussed standards of proof associated with particular systems of justice, there are other systems of justice with different proof standards. As a result, the particulars of the standard of proof in the particular legal system with jurisdiction must be met by each side as appropriate to the situation. But regardless of the specific standard of proof and legal system, the technical issues of determining control with regard to attribution remains the same.

Today, tools don't address these sorts of approaches, and no tools on the horizon appears to provide a path to automating the approach as a whole. However, there are many components of this sort of analysis that could be automated. These include, without limit, databases of analysis already performed on specific mechanisms so that repeated analysis need not be done, tools for managing the process, depicting results (e.g., depictions of ℙ, and the underlying components ℂ→𝔼, ℂ'→𝔼', etc.), and tools to assist with analysis. However, there are limitations on such tools due to the potentially unlimited number of paths, the problem of convergence of the digital space with time[14], which leads to the inability do uniquely identify causes associated with traces in almost all cases, the question of granularity, which appears to be a psychological rather than technical issue, the practically unattainable set of all possible procedures, [6] and other similar complexity and decidability issues with the general class of Turing capable automata.

The notion of control seems fundamental to the claim of "knowingly and without permission" "access" a computer system and "intentionally cause damage". In this paper we have codified this notion into cases differentiated based on Turing capability and syntax and semantics of mechanisms, provided a methodology for examination, and language for stating results of such examination.

References

[1] USC 18, PI, Ch47, § 1030, *Fraud and related activity in connection with computers*. http://www.law.cornell.edu/uscode/18/1030.html [Whoever (5) (A) knowingly causes the transmission of a program, information, code, or command, and as a result of such conduct, intentionally causes damage without authorization, to a protected computer; (B) intentionally accesses a protected computer without authorization, and as a result of such conduct, recklessly causes damage; or (C) intentionally accesses a protected computer without authorization, and as a result of such conduct, causes damage and loss.]

[2] CA Penal Code Section 502, *Unauthorized Access To Computers, Computer Systems and Computer Data* [any person who commits any of the following acts is guilty of a public offense:(1) Knowingly accesses and without permission alters, damages, deletes, destroys, or otherwise uses any data, computer, computer system, or computer network in order to either (A) devise or execute any scheme or artifice to defraud, deceive, or extort, or (B) wrongfully control or obtain money, property, or data. (2) Knowingly accesses and without permission takes, copies, or makes use of any data from a computer, computer system, or computer network, or takes or copies any supporting documentation, whether existing or residing internal or external to a computer, computer system, or computer network. (3) Knowingly and without permission uses or causes to be used computer services. (4) Knowingly accesses and without permission adds, alters, damages, deletes, or destroys any data, computer software, or computer programs which reside or exist internal or external to a computer, computer system, or computer network. (5) Knowingly and without permission disrupts or causes the disruption of computer services or denies or causes the denial of computer services to an authorized user of a computer, computer system, or computer network. (7) Knowingly and without permission accesses or causes to be accessed any computer, computer system, or computer network. (8) Knowingly introduces any computer contaminant into any computer, computer system, or computer network.]

[3] T. Stallard and K. Levitt, "*Automated Analysis for Digital Forensic Science: Semantic Integrity Checking*", ACSAC-2003, p. 160.

[4] B. Carrier, "*A Hypothesis Based Approach to Digital Forensic Investigation.*" PhD Dissertation; Purdue University; May, 2006.

[5] P. Gladyshev, "*Formalising event reconstruction in digital investigations.*" PhD Dissertation; University College Dublin; 2004

[6] F. Cohen, "A*nalysis of redundant traces for consistency*", IEEE International Workshop on Computer Forensics in Software Engineering (CFSE 09), p. 42, Seattle, Washington, USA, July 20-24, 2009

[7] C. Chaski, "*Who's At The Keyboard? Authorship Attribution in Digital Evidence Investigations*", International Journal of Digital Evidence, V4#1, p 2005.

[8] M. Corney, "*Analysing E-mail Text Authorship for Forensic Purposes*", Masters Thesis, Queensland University of Technology, March, 2003 [This thesis examines using a variety of classifiers with output fed into a Support Vector Machine (SVM). The approach is to compare a specific email to an SVM model built from a corpus of emails with known provenance e.g. given 20 emails from each of A, B and C, compare a new email to those models to see which author it is most likely to belong to.]

[9] K. Narayanaswamy, "Survey/Analysis of Levels I, II, and III Attack Attribution Techniques", Cs3, Inc., April 27, 2004.

[10] D. Cohen and K. Narayanaswamy, "Techniques for Level 1 Attack Attribution", 2004/04/08, CS3, Inc., available at: http://isis.cs3-inc.com/level1-x.html

[11] D. Cohen and K. Narayanaswamy, "Techniques for Level 2 Attack Attribution", 2004/03/22, CS3, Inc., available at: http://isis.cs3-inc.com/level2.html

[12] F. Cohen, "Attribution of messages to sources in digital forensics cases", HICSS-43, Jan 7, 2010.

[13] A. Turing, "*On Computable Numbers, with an Application to the Entscheidungsproblem*", London Math Soc. Ser 2. Vol 42,Nov 12,1936,230-265.

[14] F. Cohen, "Digital Forensic Evidence Examination", ASP Press, 2009 [Chapter 3 also appearing at http://infophys.com/InfoPhys.pdf covers The Physics of Digital Information].

[15] F. Cohen, "*A Secure World-Wide-Web daemon*", IFIP-TC11, Computers and Security, V15#8, 1996, pp. 707-724(18). [This paper describes a provably secure get-only Web server that was subsequently proven to meet its security requirements.]

[16] F. Cohen, "*Two models of digital forensics examination with application examples in bulk message analysis*", IEEE SADFE (in conjunction with the IEEE Oakland Conference), 2009-05-21, Oakland, CA

[17] F. Cohen, Cynthia Phillips, Laura Painton Swiler, Timothy Gaylor, Patricia Leary, Fran Rupley, Richard Isler, and Eli Dart "*A Preliminary Classification Scheme for Information System Threats, Attacks, and Defenses; A Cause and Effect Model; and Some Analysis Based on That Model*", Encyclopedia of Computer Science, 1999.

[18] *Common Vulnerabilities and Exposures* (CVE), http://cve.mitre.org/

[19] Crocker, D. and P. Overell, "*Augmented BNF for Syntax Specifications: ABNF*", RFC 4234, October 2005. This also references RFC 733 and 822 as source documents. Available at http://www.ietf.org/rfc/rfc4234.txt