The loop fallacy and deterministic serialisation in tracing intrusion connections through stepping stones

Xinyuan Wang

Department of Information and Software Engineering,
George Mason University, Fairfax, VA, USA
E-mail: xwangc@gmu.edu

Abstract: In order to conceal their identity and origin, network based intruders seldom attack directly from their own hosts, but rather stage their attacks through intermediate 'stepping stones'. To identify attackers behind stepping stones, it is necessary to be able to trace and correlate attack traffic through the stepping stones and construct the correct intrusion connection chain. A complete solution to the stepping stones tracing problem consists of two complementary parts. Firstly, the set of correlated connections that belongs to the same intrusion connection chain has to be identified; secondly, those correlated connections need to be serialised in order to construct the accurate and complete intrusion connection chain. Existing approaches to the tracing problem of intrusion connections through stepping stones have focused on identifying the set of correlated connections that belong to the same connection chain and have overlooked the serialisation of those correlated connections. In this paper, we use set theoretic approach to analyse the theoretical limits of the correlation-only approach, demonstrate the gap between the perfect stepping stone correlation solution and the perfect solution to the stepping stones tracing problem, and we show what it takes to fill the gap. Firstly, we identify the serialisation problem and the loop fallacy in tracing connections through stepping stones. We formally demonstrate that even the perfect correlation solution, which gives us all and only those connections that belong to the same connection chain, does not guarantee to be able to serialise the correlated connections deterministically. Secondly, we show that the complete set of correlated connections, even with loops, could be serialised deterministically without synchronised clock. We present an efficient intrusion path construction method based on adjacent correlated connection pairs. Finally, we show that the incomplete set of correlated connections due to limited observing area of stepping stones only provides enough information to construct a partial-order of subsequences of the connection chain in general, and we present an efficient way to determine when the incomplete set of correlated connections could be serialised deterministically.

Keywords: stepping stones; intrusion tracing; serialisation; correlation.

Reference to this paper should be made as follows: Wang, X. (2006) 'The loop fallacy and deterministic serialisation in tracing intrusion connections through stepping stones', *Int. J. Security and Networks*, Vol. 1, Nos. 3/4, pp.184–197.

Biographical notes: Xinyuan Wang is currently an Assistant Professor in the Department of Information and Software Engineering at George Mason University, Fairfax, VA, USA. He received his PhD in Computer Science from North Carolina State University in 2004. His research interests have been around network security in general and intrusion tracing in particular. He has substantial research experiences in developing highly effective attack attribution systems, and he has invented the first active watermark tracing approach.

1 Introduction

Network Based Attackers seldom attack directly from their own hosts, but rather stage their attacks through intermediate 'stepping Stones' to hide their identity and origin (Stoll, 2000). For example, an attacker at host A may telnet or ssh into host B, and from there launch an attack against host C. The victim at host C can use IP traceback techniques (Goodrich, 2002; Li et al., 2004; Savage et al., 2000; Snoeren et al., 2001) to find out that the attack comes from host B, but IP traceback cannot determine that the attack actually originate from host A behind host B. By laundering the attack through a number of intermediate stepping stones, the attacker makes it much more difficult to determine the real source of the attack. To identify intruders behind stepping stones, it is critically important to be able to trace the intrusion connections through the stepping stones and construct the correct intrusion connection chain.

A complete solution to the stepping stones tracing problem includes:

1 the identification of the set of correlated connections that belongs to the same intrusion connection chain and
2 the serialisation of the set of correlated connections in order to construct the accurate and complete intrusion connection chain.

However, existing approaches to the stepping stone tracing problem have focused on identifying the set of correlated

connections that belong to the same intrusion connection chain and have left the serialisation of correlated connections an afterthought. While finding the right set of correlated connections forms the foundation of solving the tracing problem of intrusion connection chain, it does not, however, completely solve the tracing problem.

In this paper, we leave the problem of how to correlate intrusion connections across stepping stones aside and instead focus on analysing the theoretical limits of the correlation-only approach in the context of solving the stepping stone tracing problem. To be specific, we use set theoretic approach to demonstrate the gap between the perfect stepping stone correlation solution and the perfect solution to the stepping stones tracing problem, and what it takes to fill the gap. Our contributions are a number of fundamental results that apply to any stepping stone tracing and correlation solutions. Firstly, we identify the serialisation problem and the loop fallacy in tracing connections through stepping stones. We formally demonstrate that even the perfect correlation solution, which gives us all and only those connections that belong to the same connection chain, does not guarantee to be able to serialise the correlated connections deterministically. Secondly, we show that the complete set of correlated connections, even with loops, could be serialised deterministically without synchronised clock. We present an efficient intrusion path construction method based on adjacent correlated connection pairs. Finally, we show that the incomplete set of correlated connections due to limited observing area of stepping stones only provides enough information to construct a partial-order of subsequences of the connection chain in general, and we present an efficient way to determine when the incomplete set of correlated connections could be serialised deterministically.

The rest of this paper is organised as follows. Section 2 reviews related works on tracing intrusion connections through stepping stones. Section 3 formally formulates the overall problem of tracing intrusion connections through stepping stones and identifies the serialisation problem. Section 3.2 illustrates the loop fallacy in deterministic serialisation of correlated connections. Section 4.1 analyses the serialisation problem and presents a solution to the deterministic serialisation of the complete set of the correlated connections without synchronised clock. Section 5.4 analyses the serialisation of the incomplete set of the correlated connections and identifies a way to determine if and when the incomplete set of the correlated connections could be serialised deterministically. Section 6.3 concludes this paper.

2 Related works

Existing works on tracing intrusion connections have been based on three different characteristics of the intrusion connections:

1 host login activity
2 connection content (i.e. packet payload) and
3 connection packet timing.

The earliest works (DIDS by Snapp et al. (2001), CIS by Jung et al. (1993)) on tracing intrusion connections through stepping stones were based on tracking users' login activities at different hosts. The fundamental problem of host login activity based approaches is that the information of user's login activity collected from stepping stones is not trustworthy. Because the attacker who has root control of the stepping stone could easily disguise, delete or forge user login activities at the stepping stone, tracing approaches based on tacking users' login activities at stepping stone could be easily defeated. To overcome this shortcoming, Tracing and correlation approaches based on comparing packet contents have been developed.

Thumbprinting by Staniford-Chen and Heberlein (1995) is the first published network content based correlation technique. It utilises a small quantity of information (called thumbprint) to summarise a certain section of a connection. The thumbprint is built, through principle component analysis technique in statistics, upon the frequencies that each character occurs within a period of time. Ideally it can distinguish a connection from unrelated connections and correlate a connection with those related connections in the same connection chain. Because it correlates based on connection content, thumbprinting works even when all stepping stones are compromised and under attacker's total control, and it can be useful when only part of the internet implements thumbprinting.

SWT by Wang et al. (2001a,b) is a network content based correlation and tracing scheme that applies the principles of steganography and active networking. It exploits two properties of the connections across stepping stones:

1 the application level content of unencrypted connections is invariant across stepping stones and

2 interactive intrusion connections across stepping stones are bidirectional and symmetric at the granularity of connection.

SWT is 'sleepy' in that it does not introduce overhead when no intrusion is detected, yet it is 'active' in that when an intrusion is detected, the intrusion target 'injects' carefully designed watermark into the backward response traffic of the intrusion connection. SWT traces and correlates intrusion connections based on the injected watermarks in their application content and is able to trace through the intrusion connection chain across all stepping stones within a single keystroke of the intruder. With its unique active tracing, SWT is able to trace through all the stepping stones even when the intruder is silent.

Network content based approaches require that the packet payload content be invariant across stepping stones. Since the packet payload content could be changed by encryption (i.e. IPSEC, SSH), network content based approaches are limited to correlating and tracing unencrypted connections. To be able to correlate and trace encrypted attack traffic, new generation of network based correlation approaches has been developed, based on the inter-packet timing characteristics.

The ON/OFF based correlation by Zhang and Paxson (2001) is the first network-based correlation scheme that utilises the inter-packet timing characteristics to correlate interactive connections across stepping-stones. Depending on whether there is any traffic for a (adjustable) period of time, the duration of a flow can be divided into either

ON or OFF periods. The correlation of two flows is based on mapping the ends of OFF periods (or equivalently the beginnings of ON periods). Because it correlates based on inter-packet timing characteristics rather than packet content, ON/OFF based correlation is able to correlate both encrypted and unencrypted connections, and it is robust against packet payload padding. However, ON/OFF based correlation requires that the packets of connections have precise, synchronised timestamps in order to be able to correlate them.

The deviation-based approach by Yoda and Etoh (2000) is another network-based correlation scheme. It defines the minimum average delay gap between the packet streams of two TCP connections as deviation. The deviation based approach considers both the packet timing characteristics and the TCP sequence numbers. It does not require clock synchronisation and is able to correlate connections observed at different points of network. However, it can only correlate TCP connections that have one-to-one correspondences in their TCP sequence numbers, and thus is not able to correlate connections where padding is added to the packet payload (e.g. when certain types of encryptions are used).

Unlike the ON/OFF based approach, IPD-based approach (Wang et al., 2002) does not require synchronised timestamps, and it defines its correlation metrics over the inter-packet timing characteristics. It has shown that

1 (after some filtering) the Inter-Packet Delays (IPDs) of both encrypted and unencrypted, interactive connections are preserved across many router hops and stepping stones

2 the timing characteristics of normal interactive connections such as telnet and SSH are almost always distinct enough to prove correct correlation across stepping stones and

3 both encrypted and unencrypted interactive connections can be effectively correlated based on IPDs.

Yung (2002) proposed a method for detecting the existence of connection chain based on the time gap between the client request and the server reply echo. However, its method can only tell whether an interactive flow belongs to some connection chain and it is unable to correlate a flow to any other flows.

While the inter-packet timing based correlations are currently the most capable and promising approaches, they are vulnerable to the active timing perturbation by adversary. The adversary could perturb the timing characteristics of a connection by selectively or randomly introducing extra delays when forwarding packets at the stepping stone. The timing perturbation could either make related flows have very different timing characteristics or make unrelated flows exhibit similar timing characteristics, which would either decrease the correlation true positive rate or increase the correlation false positive rate.

To address the new challenge of active timing perturbation by adversary, Donoho et al. (2002) have recently studied the theoretical limits of the adverse effects of the active timing perturbation. By using a multiscale analysis technique, they are able to separate the long-term behaviour of the connection from the short-term behaviour of the connection, and they show that correlation from the long-term behaviour (of sufficiently long flows) is still possible despite timing perturbation by the attacker. However, they do not present any tradeoffs between the magnitude of the timing perturbation, the desired correlation effectiveness and the number of packets needed. Another important issue that is not addressed by Donoho et al. (2002) is the correlation false positive rate. What left open are the question whether correlation is achievable for arbitrarily distributed (rather than Pareto distribution conserving) random timing perturbation, and an analysis of the achievable tradeoff of the false positive and true positive rates.

Wang and Reeves (2003) have developed a watermark based correlation framework. Unlike any previous timing based approaches, their IPD watermark based correlation is active in it actively embeds some unique watermark into the flow by slightly adjusting the timing of selected packets and utilises redundancy techniques to make the embedded watermark robust. If the embedded watermark is unique enough and robust enough against the timing perturbation by adversary, the watermarked flow could be uniquely identified and thus effectively correlated. By utilising redundancy techniques, the IPD watermark based correlation reveals a rather surprising result on the inherent limits of random timing perturbations over sufficiently long flows. Work by Wang and Reeves (2003) is the first that identifies

1 the accurate quantitative tradeoffs between the achievable correlation effectiveness, the defining characteristics of the timing perturbation and

2 a provable upper bound on the number of packets needed to achieve any desired correlation effectiveness, given a bound on the magnitude of timing perturbation.

Compared with previous passive timing based correlation approaches, the active IPD watermark based correlation is significantly more robust against the random timing perturbation by adversary and requires lesser packets at the same time.

Blum et al. (2004) proposed another passive, timing based correlation method that considers both correlation true positive and false positive at the same time. However, their work did not describe any experimental results, nor did it address such practical issues as how to derive model parameters in real-time.

As we have shown, almost all previous network-based tracing approaches have focused on correlation only. While the correlation of encrypted attack traffic is still a challenging task due to various active countermeasures used by adversary, there is a limit on the theoretically achievable effectiveness of even the perfect correlation solution. It is very important to understand the inherent limit of the correlation only approach in the context of tracing attack traffic across stepping stones.

In the rest of this paper, we investigate the gap between the perfect stepping stone tracing solution and the perfect stepping stone correlation solution, and we show what it takes to fill the gap. We first consider the ideal case where we could monitor all flows and find all the correlated flows. We later

relax our assumption and consider the case of correlation with incomplete set of correlated connections due to limited observing capability.

3 The problem of tracing intrusion connections through stepping stones

In this section, we use set theoretic approach to formulate the overall problem of tracing intrusion connections through stepping stones. We first review the basic concepts of Set Theory (Machover, 1996) we used.

3.1 Ordinals of basic set theory

For binary relation R on set S, we use Field(R) to denote the set of elements of each ordered pair in R. That is $Field(R) = \{x :< x, y >\in R \vee < y, x >\in R\}$. We also use the notations $< x, y > \in R$ and $x\ R\ y$ interchangeably.

Binary relation R is called

Reflexive:	if $\forall x \in Field(R)[x\ R\ x]$
Irreflexive:	if $\forall x \in Field(R)[\neg(x\ R\ x)]$
Symmetric:	if $\forall x, y \in Field(R)[x\ R\ y \Leftrightarrow y\ R\ x]$
Anti-symmetric:	if $\forall x, y \in Field(R)[(x\ R\ y \wedge y\ R\ x) \Rightarrow x = y]$
Transitive:	if $\forall x, y, z \in Field(R)[(x\ R\ y \wedge y\ R\ z) \Rightarrow x\ R\ z]$
Linear (*connected*):	if $\forall x, y \in Field(R)[x\ R\ y \vee y\ R\ x]$

Binary relation R on S is a *partial-order* if it is both antisymmetric and transitive. Partial-order R on S is a *total-order* if it is linear (connected).

Given partial-order R on S and $A \subseteq S$, if there exists $a \in A$ such that $\forall x \in A[a\ R\ x]$, we say a is the R-least (or R-minimal) in A. A total-order R on S is a *well-order* on S if every non-empty subset of S has a R-minimal.

3.2 Overall tracing problem model

Given a series of computer hosts H_1, H_2, … H_{n+1} ($n > 1$), when a person (or a program) sequentially connects from H_i to H_{i+1} ($i = 1, 2, \ldots n$), we refer to the sequence of connections $< c_1, c_2, \ldots c_n >$, where $c_i = < H_i, H_{i+1} >$ ($i = 1, \ldots n$), as a *connection chain* on $< H_1, H_2, \ldots H_{n+1} >$. Here all c_i's are always distinct, but not all H_i's are always distinct. In case some host appears more than once in sequence $< H_1, H_2, \ldots H_{n+1} >$, there exists a loop in the connection chain $< c_1, c_2, \ldots c_n >$.

The *tracing problem* of a connection chain (or stepping stone) is, given c_1 of some unknown connection chain $< c_1, c_2, \ldots c_n >$ ($n > 1$), to identify $< c_1, c_2, \ldots c_n >$.

Any particular connection chain $< c_1, c_2, \ldots c_n >$ is *sequence* of connections. We refer those connections within same connection chain as *correlated* to each other and corresponding set $\{c_1, c_2, \ldots c_n\}$ as *set of correlated connections* or *correlation set*. This can be formally modelled by a binary relation on the overall connection set. We define binary relation CORR on the overall connection set $\widehat{C}$ such that

$$\forall c, c' \in \widehat{C}[c\text{CORR}c'\text{iff} \quad (c \in \{c_1, c_2, \ldots c_n\} \Rightarrow c' \in \{c_1, c_2, \ldots c_n\})] \tag{1}$$

It is obvious that CORR is specific to the correlation set and it is

1 self-reflexive

2 symmetric and

3 transitive.

Therefore binary relation CORR is an equivalence relation on C and it partitions the overall set of connections into a particular set of correlated connections and rest of the connections.

Because connection chain $< c_1, c_2, \ldots c_n >$ is an ordered set, each c_i has an order number Ord(c_i) associated with it. The overall ordering information of $< c_1, c_2, \ldots c_n >$ can be formally modelled by the binary relation $\angle$ on $\{c_1, c_2, \ldots c_n\}$ such that

$$\forall c, c' \in \{c_1, c_2, \ldots c_n\}[c \angle c'\text{iff Ord}(c) < \text{Ord}(c')] \tag{2}$$

It is obvious that $\angle$ well-orders set $\{c_1, c_2, \ldots c_n\}$ and it uniquely determines $< c_1, c_2, \ldots c_n >$ from $\{c_1, c_2, \ldots c_n\}$.

For any particular connection chain $< c_1, c_2, \ldots c_n >$, there exists unique binary relations CORR and $\angle$, which in turn uniquely determine $< c_1, c_2, \ldots c_n >$. Therefore, the overall tracing problem of connection chain can be divided into the following subproblems:

1 *Correlation problem*: given c_1 of some unknown connection chain $< c_1, c_2, \ldots c_n >$, identify set $\{c_1, c_2, \ldots c_n\}$; Or equivalently, given any two connections c and c', determine if c CORR c'.

2 *Serialisation problem*: given unordered set of correlated connections $C = \{c_1, c_2, \ldots c_n\}$, serialise $\{c_1, c_2, \ldots c_n\}$ into an ordered set $< c'_1, c'_2, \ldots c'_n >$ ($c'_i \in C, i = 1, \ldots n$) such that $c'_i \angle c'_{i+1} (i = 1, \ldots n-1)$; Or equivalently, given any two connections c and c', determine if $c \angle c'$ or $c' \angle c$.

Two observations can be made about the overall tracing problem:

1 the result of the serialisation problem is based upon the result of the correlation problem and

2 the perfect result of the overall tracing problem consists of the perfect result of the correlation problem and the perfect result of the serialisation problem based upon the perfect correlation result.

Observation 1 shows the inter-dependency between the correlation problem and the serialisation problem, and it explains why existing works on the overall tracing problem have focused on the correlation problem. Observation 2 reveals that while the solution to the correlation problem is the very foundation of the solution to the overall tracing problem, it is not adequate to construct the complete solution to the overall tracing problem. What's missing from the correlation-only approach is the serialisation of the correlation result.

In the rest of this paper, we identify, analyse this gap and we present an efficient solution to the serialisation problem.

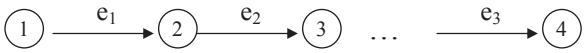
4 The loop fallacy in deterministic serialisation of correlated connections

The complete solution to the stepping stone tracing problem should give not only the complete set of correlated connections but also the relative order between those correlated connections. Such order information about the observed correlated connections is extremely important in the security forensic analysis and it helps to identify the initial penetration point of the system and network being attacked.

In real world scenario, it is likely that any network attack tracing system has limited observing area. While tracing systems with limited observing area could miss some stepping stones and attack traffic, such tracing systems should deterministically point out the right direction from which the intrusion comes.

Unfortunately, even with perfect correlation solution, which gives all and only those correlated connections within the observing scope that belong to the same connection chain, it is still not adequate to deterministically construct the complete intrusion path or even find the right direction from which intrusion comes in. In case the intrusion connection passes each stepping stone only once, each stepping stone has only one incoming and outgoing connection, and there is only one way to serialize those correlated connections to construct the intrusion path as shown in Figure 1.

Figure 1 Loopless linear connection chain



As an effort to complicate serialisation of correlated connections, attackers could easily stage their attacks through some stepping stone more than once, which would create loops or cycle in the intrusion connection chain. Such a loop in the intrusion connection chain would introduce dilemma in serialising correlated connections. Figure 2 shows an example of intrusion connection chain with multiple stepping stones, where node 1 is the intrusion target and e_1, e_2, e_3, e_4, e_5, e_6, e_7 are the backward connections from the intrusion target toward the source of the intrusion. A perfect correlation solution would report that e_1, e_2, e_3, e_4, e_5, e_6, e_7 are correlated and belong to the same intrusion connection chain. Given the knowledge that node 1 is the intrusion target, we know that the intrusion to node 1 comes from node 2 as there is only one correlated connection e_1 between node 1 and node 2. However, node 2 has two outgoing connections e_2 and e_5 that are part of same connection chain, and there are multiple ways to serialise those correlated connections. Furthermore, when some stepping stones are outside of the observing area of the tracing system, loops in the intrusion connection chain could introduce dilemma in determining the right direction from which the intrusion comes in. Figure 3 shows two such examples. When nodes 3,4,5 are outside the observing area of the tracing system, node 2 sees two correlated outgoing connections e_2 and e_5. Without additional information, there is no way for node 2 to determine which connection points to the host that is closer to the intrusion source. When node 3 is out of the observing scope, there are multiple ways to serialising the correlated connections, which point to different directions to the intrusion source. For example, both serialisation $< e_1, e_2, \ldots e_3, e_4, e_5, e_6, e_7 >$ and $< e_1, e_5, e_6, e_7, \ldots e_3, e_4, e_2 >$ are possible, which imply e_7 and e_2, respectively as the connections pointing to the intrusion source.

Figure 2 Loop fallacy in serialising correlated connections

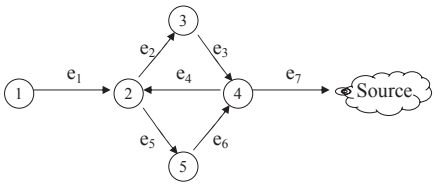
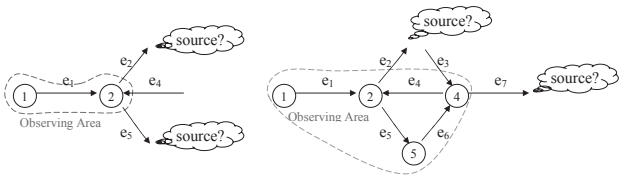


Figure 3 Tracing dilemma with limited observing area



These examples indicate that correlation only approach is a partial-solution to the problem of tracing intrusion connections through stepping stones. What is missing from the correlation only solution is the serialisation of those correlated connections. It is this phenomenon – that people in general do not take the potential loops or cycles of intrusion connection chain into account when intuitively solving the tracing problem with correlation only approaches – that is named 'the loop fallacy' in tracing intrusion connections through stepping stones.

4.1 Deterministic serialisation of correlated connections

We have shown that the set of correlated connections itself is not adequate to serialise those correlated connections deterministically. In order to deterministically serialise correlated connections, some additional information on the correlated connection is needed.

One intuitive way to serialise correlated connection is use globally synchronised timestamp to determine the relative order of correlated connections. However, a packet across a connection chain could traverse one stepping stone and reach another stepping stone in less than 1 msec. In order to use timestamps of correlated connections to determine their relative order, each host on the internet needs to have synchronised clock of precision better than 1 msec. Without special device, normal computer host is not able to achieve clock synchronisation of such precision. Furthermore, even if every host has somehow achieved clock synchronisation of

such precision, microscopic deviations in device hardware in each computer could easily lead to clock skew of tens of milliseconds (Kohno et al., 2005). Therefore, it is impractical to simply use timestamps of correlated connections to determine their relative order.

Another way to serialise correlated connections is based on adjacency or causal relationship of those correlated connections. Compared with timestamp based approach, adjacency based approach does not require any global clock synchronisation at all and is robust against network delay jitters.

In this paper, we focus on solving the problem of deterministic serialisation of correlated connections without global clock synchronisation. We initially assume that the tracing system has global observing capability and we are able to get all the correlated connections. We will relax our assumption and discuss the serialisation with incomplete set of correlated connections in Section 5.4.

5 Deterministic serialisation with the complete set of correlated connections

In this section, we use set theory approach to formally establish that while the complete set of correlated connection itself is not adequate to serialise those correlated connections, the complete set of adjacent correlated connection pairs of each stepping stone is sufficient to serialise those correlated connection deterministically even if there is loops with the connection chain.

Given a set of correlated connections C, it can be thought as a set of edges of a directed graph DG such that DG=$< V, E >$, $V = \{x : \exists < x, y > \in C \ \vee \ \exists < y, x > \in C\}$ and $E = C$. We assume that there is no *self-loop edge* in DG, that is $\forall < u, v > \in E \ [u \neq v]$. Therefore, the serialisation of elements of C can be represented by the ordering of elements of either V or E.

We use $u \rightarrow v$ to represent that there is directed path from u to v. and we define *DG* to be *one-way connected* if: $\forall u, v \in V \ [\exists u \rightarrow v \ \vee \ \exists v \rightarrow u]$, and DG to be *edge one-way connected* if: $\forall < u_1, v_1 >, < u_2, v_2 > \in E \ [v_1 \rightarrow u_2 \ \vee \ v_2 \rightarrow u_1]$. For example, in Figure 2, nodes 1 – 4 is one-way connected (through intermediate nodes 2 and 3) but nodes 4 to node 1 is not one-way connected as there is no directed path from node 4 to node 1. Similarly edges $e_1 - e_4$ is edge one-way connected (through intermediate edges e_2 and e_3) and $e_4 - e_1$ is not edge one-way connected.

Because the intrusion path is necessarily one-way connected and edge one-way connected, the correct serialisation of the complete set of the correlated connections has to maintain the one-way connectivity of the edges and end-points of correlated connections.

5.1 Point connectivity and serialisation based on point adjacency

Here we consider the serialisation of correlated connections based on point adjacency property of those correlated connections, and we use binary relation to formally define the point adjacency and point connectivity and reason about the serialisation based point adjacency.

We define *Point-Adjacency* (*P-Adj*) on V as the binary relation $\{ < u, v > : < u, v > \in E\}$. Since there is no self-loop edge in E, P-Adj is irreflexive and it models the adjacency relation among the elements of V.

We define it Point Connectivity (*PC*) as the binary relation on V, such that

1 $\forall < u, v > \in E \ [u \text{ PC } v]$

2 $\forall u, v, w \in V \ [(u \text{ PC } v \ \wedge \ v \text{ PC } w) \ \Rightarrow \ u \text{ PC } w]$

Therefore binary relation PC is the *transitive-closure* of *P*-Adj. Here we use $<_{\text{PC}}$, to represent PC. If there exists some $v \in V$, such that $\forall u \in V [u \neq v \ \Rightarrow \ v \ <_{\text{PC}} \ u]$, we define such an element v as PC-minimal on V.

From the definitions, it is easy to see that given a DG, there is only one *P*-Adj and $<_{\text{PC}}$ defined on V. Here binary relation < formally models the directed connectivity among the vertices in V and $u \ <_{\text{PC}} \ v$ *iff* there exists a path from u to v.

The following theorem, whose proof can be found in the Appendix, describes the necessary and sufficient conditions for the serialisation based point adjacency to be deterministic.

Theorem 1: *The necessary and sufficient conditions for <pc to be well-order on V are:*

1 DG=$< V, E >$ is one-way connected

2 DG has no directed cycles.

As shown in Figure 2, an intrusion connection chain may pass a particular stepping stone more than once, which would introduce directed cycles in the connection chain. Therefore, the serialisation of end points of correlated connections based on point adjacency is not deterministic.

5.2 Edge connectivity and serialisation based on edge adjacency

We now consider serialisation of correlated connections based on edge adjacency relation among those correlated connections. For any connection c between two hosts, we use $Start(c)$ to denote the origination host of c and we use $End(c)$ to denote the termination host of c.

We define *Edge-Adjacency* (*E-Adj*) on E as the binary relation: $\{<< u, v >, < v, w >> : < u, v >, < v, w > \in E\}$. It is easy to see that E-Adj is irreflexive and it models the adjacency relation among the elements of E.

We define *Edge Connectivity* (*EC*) as the binary relation on E, such that

1 $\forall e_i, e_j \ \in E \ [(\text{End}(e_i) = \text{Start}(e_j)) \Rightarrow e_i \text{ EC } e_j]$

2 $\forall e_i, e_j, e_k \ \in E [(e_i \text{ EC } e_j \ \wedge \ e_j \text{ EC } e_k) \ \Rightarrow e_i \text{ EC } e_k]$

Therefore binary relation EC is the *transitive-closure* of E-Adj. Here we use $<_{\text{EC}}$ to represent EC. If there exists some $e \in E$, such that $\forall e_i \in E \ [e_i \neq e \Rightarrow e <\text{EC} e_i]$, we define e as *EC-minimal* on E.

From the definitions, it is easy to see that given a DG, there is only one E-Adj and $<_{\text{EC}}$ defined on E. Binary relation $<_{\text{EC}}$ also models the directed connectivity among vertices of V and $< u_1, v_1 > \ <_{\text{EC}} \ < u_2, v_2 >$ iff there exists a path from v_1 to u_2.

The following theorem describes the necessary and sufficient conditions for the serialisation based on edge adjacency to be deterministic.

Theorem 2: *The necessary and sufficient conditions for* $<_{EC}$ *to be a well-order on E are:*

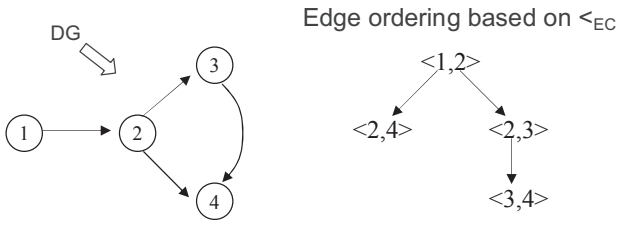
1 DG=$< V, E >$ is one-way connected

2 DG has no directed cycles and

3 DG has no out-branch: $\forall v \in V$(v has at most single successor).

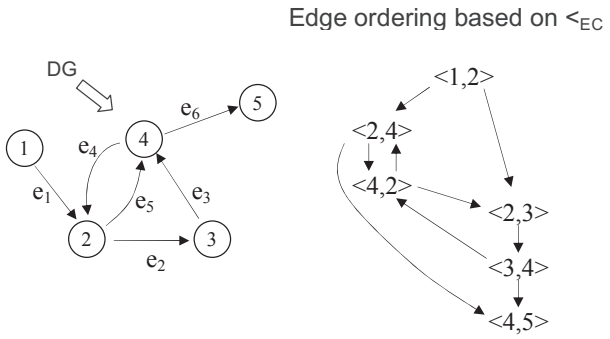
Please be noted that given DG=$< V, E >$, in order for $<_{EC}$ to well-orders E, DG must have no out-branch, which is not required for $<_{PC}$ to well-order V. Figure 4 shows such an example, where $<_{PC}$ well-orders $\{1, 2, 3, 4\}$ and $<_{EC}$ is not even a total-order on E as $< 2, 3 >$ and $< 2, 4 >$ have no relative order.

Figure 4 Point connectivity $<_{PC}$ and Edge connectivity $<_{EC}$



Because no directed cycles is a necessary condition for $<_{EC}$ to be well-order on E, the serialisation of correlated connections based on edge adjacency is not deterministic either. Figure 5 shows an example of serialisation of connections based on edge adjacency, both edge serialisations: $<< 1, 2 >, < 2, 3 >, < 3, 4 >, < 4, 2 >, < 2, 4 >, < 4, 5 >>$ and $<< 1, 2 >, < 2, 4 >, < 4, 2 >, < 2, 3 >, < 3, 4 >, < 4, 5 >>$ satisfy $<_{EC}$.

Figure 5 Edge serialisation based on edge connectivity



5.3 Serialisation based on adjacent connection pairs

We have demonstrated that the serialisation of the complete set of correlated connections based on either point or edge adjacency is not always deterministic and unique. When the intrusion connection chain has loops or cycles, there are multiple ways to serialise those correlated connections while keeping the connectivity. This dilemma is due to the fact that there could be more than two connections adjacent to each other through one vertex and the set of correlated connections gives no clue about how to pair match those incoming connections with outgoing connection.

A stepping stone may have multiple incoming connections and outgoing connections correlated. To serialise multiple correlated incoming and outgoing connections deterministically, we need information about how the incoming connections and outgoing connections to and from a stepping stone are pair matched. This is modelled by the concept of adjacent connection pair.

Given a connection chain $< c_1, c_2, \ldots c_n >$ on host list $< H_1, H_2, \ldots H_{n+1} >$, where connection c_i is from H_i to H_{i+1}, we define $< c_i, c_{i+1} >$ $(i = 1, 2, \ldots n-1)$ as the *adjacent connection pair* on host H_{i+1}. It may be noted that all H_i's $(1 \leq i \leq n+1)$ are not necessarily distinct, but all c_i's $(1 \leq i \leq n)$ are always distinct. Even if both c_i and $c_j (1 \leq i, j \leq n$ and $i \neq j)$ could start from the same host $H_i = H_j$ to the same host $H_{i+1} = H_{j+1}$, connections c_i and c_j are still different based on their setup time. Therefore, the adjacent connection pair carries the relative order information about two adjacent connections on a particular vertex and $< c_i, c_{i+1} >$ means connection c_i *happens right before* connection c_{i+1}. We use *PE-Adj* to represent the set of adjacent connection pairs. By definition, PE-Adj is anti-symmetric and irreflexive.

Given a set of adjacent connection pairs PE-Adj, we can construct the set of connection

$$E_{\text{PE-Adj}} = \{e : \exists < e, e_i > \in \text{PE-Adj} \vee \exists < e_j, e > \in \text{PE-Adj}\}$$

and the set of vertices

$$V_{\text{PE-Adj}} = \{v : \exists < e_i, e_j > \in \text{PE-Adj} [v = \text{Start}(e_i) \vee v = \text{End}(e_i) \vee v = \text{End}(e_j)]\}$$

and the directed graph DG=$< V_{\text{PE-Adj}}, E_{\text{PE-Adj}} >$. Therefore PE-Adj is binary relation on $E_{\text{PE-Adj}}$ and PE-Adj $\subseteq E$-Adj on $E_{\text{PE-Adj}}$.

We define binary relation Paired Edge Connectivity (PEC) on $E_{\text{PE-Adj}}$, such that

1 $\forall < e_i, e_j > \in \text{PE-Adj}[e_i \text{PEC} e_j]$

2 $\forall e_i, e_j, e_k \in E_{\text{PE-Adj}}[(e_i \text{PEC} e_j \wedge e_j \; PEC \; e_k) \Rightarrow e_i \text{PEC} e_k]$

By definition of PEC, $e_i \text{PEC} e_j$ means e_i happens before e_j. Therefore binary relation PEC is anti-symmetric. Because of PEC is also transitive, PEC is a partial-order. Here we use $<_{\text{PEC}}$ to represent PEC. If there exists $< u_1, v_1 > \in E$, such that $\forall < u_2, v_2 > \in E[< u_1, v_1 > \neq < u_2, v_2 > \Rightarrow < u_1, v_1 > <_{PEC} < u_2, v_2 >]$, we define $< u_1, v_1 >$ as *PEC-minimal* on E.

To utilise the result of Theorem 1, we transform the directed graph DG into another directed graph. In particular, element of PE-Adj $< e_i, e_j >$, can also be thought as a directed edge whose endpoints (tail and head) are e_i and e_j. By mapping edges in DG into vertices and mapping element of PE-Adj, $< e_i, e_j >$ into edges, another directed graph can be deterministically constructed.

We define the *paired line graph* of DG, written as PL(DG), as the directed graph whose vertices are the edges of DG, and whose edges are $< e_i, e_j > \in$ PE-Adj. That is, the edges in DG correspond to vertices in PL(DG), and adjacent connection pairs in DG corresponds to edges in PL(DG).

Therefore $V(\text{PL(DG)}) \equiv E(\text{DG}) \equiv E_{\text{PE-Adj}}$, PE-Adj on DG corresponds to P-Adj on PL(DG) and $<_{\text{PEC}}$ on DG corresponds to $<_{\text{PC}}$ on PL(DG).

We further define *reachable set* of a particular edge $e \in E_{\text{PE-Adj}}$ as $\text{RS}_{\text{PE-Adj}}(e) = \{e_i : e <_{\text{PEC}} e_i\}$. PE-Adj is *edge one-way connected iff* $\forall e_i, e_j \in E_{\text{PE-Adj}}[e_i <_{\text{PEC}} e_j \vee e_j <_{\text{PEC}} e_i]$. PE-Adj is *loopless iff* $\forall e \in E_{\text{PE-Adj}}[e \notin \text{RS}_{\text{PE-Adj}}(e)]$.

We find PE-Adj is loopless if any connection within the set of adjacent connection pair will not reach itself through the adjacent connection pairs. Because $<_{\text{PEC}}$ is known antisymmetric, PE-Adj is loopless.

For any $e_i \neq e_i \in E_{\text{PE-Adj}}$, e_i and e_i are part of some connection chain[1]. Assume without loosing generality that e_i happens before e_i, and the segment between e_i and e_j in the connection chain is: $e_i\ e_{i+1}\ \ldots\ e_{i+k}\ e_i$. If PE-Adj contains all the adjacent connection pairs of the connection chain, $< e_{i+j}, e_{i+j+1} > \in$ PE-Adj $(0 \leq j \leq k-1)$ and $< e_{i+k}, e_j > \in$ PE-Adj, that is $e_i <_{\text{PEC}} e_j$. Similarly, if e_j happens before e_i, $e_j <_{\text{PEC}} e_i$. Therefore, if PE-Adj contains all the adjacent connection pairs from every stepping stone along the connection chain, PE-Adj is edge one-way connected.

The following theorem describes the sufficient condition for the serialisation based on adjacent connection pairs to be deterministic.

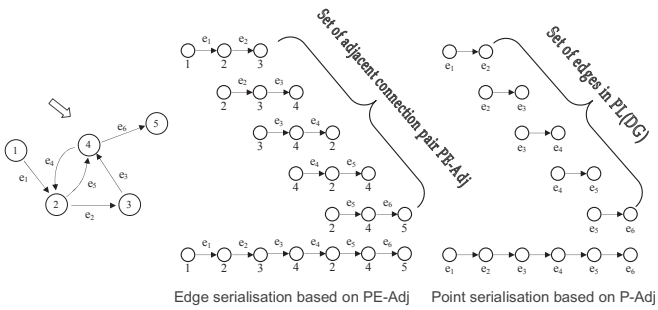
Theorem 3: *If PE-Adj is edge one-way connected and loopless,* $<_{PEC}$ *well-orders* $E_{PE\text{-}Adj}$.

Therefore, $<_{\text{PEC}}$ well-orders $E_{\text{PE-Adj}}$. In other words, the complete and accurate intrusion connection chain can be constructed deterministically from the complete set of adjacent correlated connection pairs, even if there are loops within the connection chain. Figure 6 illustrates an example of the deterministic serialisation of correlated connections from the complete set of adjacent correlated connection pairs. In particular, the left graph in Figure 6 shows the complete correlated connection chain across all stepping stones; the middle graph shows complete set of adjacent connection pairs PE-Adj and the edge serialisation based on PE-Adj; the right graph shows the corresponding point serialisation on PL(DG) based on P-Adj.

5.4 Finding adjacent correlated connection pairs

We have established that the complete set of correlated connections can be serialised deterministically based on the complete set of adjacent correlated connections pairs. Now we consider how to find the adjacent correlated connection pairs.

Figure 6 Edge serialisation based on adjacent connection pair PE-Adj



We say that the set of adjacent correlated connection pairs is with regard to (wrt) connection c if c is correlated with all connections that form the adjacent correlated connection pairs. The complete set of adjacent correlated connection pairs (with regard to connection c) is the union of all subset collected at each stepping stone.

The subset of adjacent correlated connection pairs at each stepping stone can be constructed based on the connected initial arrival or departure time by the following algorithm:

1 For each new incoming (or outgoing) connection I_i (or O_i) that is not self-loop, record I_i (or O_i) into queue $Q : x_1, x_2, \ldots\ x_{i-1}$, where $x_j (1 \leq j \leq i-1)$ could be either incoming or outgoing connection.

2 Using correlation approach to find those, if any, connections that are correlated with c, from all the connections recorded in Q.

3 Extract those correlated connections, in sequence, from Q into correlation queue Q_c.

4 Assume Q_c has $c_1, c_2, \ldots\ c_m$, if c_1 is incoming connection, the subset of correlated connection pairs is $\{ < c_1, c_2 >, < c_3, c_4 >, \ldots < c_{2\times\lfloor m/2\rfloor-1}, c_{2\times\lfloor m/2\rfloor} > \}$; if c_1 is outgoing connection, the subset of adjacent correlated connection pairs is $\{ < c_2, c_3 >, < c_4, c_5 > , \ldots < c_{2\times\lfloor (m-1)/2\rfloor}, c_{2\times\lfloor (m-1)/2\rfloor+1} > \}$.

The correctness of the algorithm is guaranteed by the following property of $Q_c = c_1, c_2, \ldots\ c_m$: if c_i is incoming connection, then c_{i+1} is outgoing connection; if c_i is outgoing connection, then c_{i+1} is incoming connection.

Therefore, in order to construct the set of adjacent correlated connection pairs, we just need to record the start of all the incoming and outgoing correlated connection at each stepping stone in sequence, from which we can construct the subset of adjacent correlated connection pairs of that stepping stone. Then we can construct the whole set of adjacent correlated connection pairs by union of all the subsets collected at each stepping stone regarding the same correlation.

For example, assume the sequence of the backward traffic from the attack target to the attack source showed in Figure 6 is $< e_1, e_2, e_3, e_4, e_5, e_6 >$. By applying the first three steps of the algorithm described above, node 2 will have its $Q_c = e_1, e_2, e_4, e_5$, node 3 will have its $Q_c = e_2, e_3$ and node 4 will have its $Q_c = e_3, e_4, e_5, e_6$. After step 4, node 2

will have a set of correlated connection pairs: $\{ < e_1, e_2 > , < e_4, e_5 > \}$, node 3 will have a set of correlated connection pairs: $\{ < e_2, e_3 > \}$, and node 4 will have a set of correlated connection pairs: $\{ < e_3, e_4 >, < e_5, e_6 > \}$. Therefore, the complete set of the adjacent correlated connection pairs is $\{ < e_1, e_2 >, < e_2, e_3 >, < e_3, e_4 >, < e_4, e_5 >, < e_5, e_6 > \}$.

6 Serialisation with an incomplete set of adjacent correlated connection pairs

In the previous section, we have assumed that the tracing system has a global observing area and it can detect all the correlated connections and all the stepping stones given a perfect correlation solution. However, any tracing system in real world is likely to have limited observing area. With limited observing area, even a perfect correlation solution may only see an incomplete set of correlated connections and stepping stones. In this section, we consider the problem of serialisation with an incomplete set of adjacent correlated connection pairs.

Unlike serialisation with the complete set of adjacent correlated connection pairs, the serialisation of correlated connections with an incomplete set of adjacent correlated connections pairs is not guaranteed to be deterministic. In the rest of this section, we start with serialization based on local *happen-before* relations and we show some examples of both deterministic and non-deterministic serialisation with incomplete set of adjacent correlated connection pairs. We then demonstrate that the serialisation with incomplete set of adjacent correlated connection pairs is equivalent to a partial-order of one or more subsequences[2] of the connection chain. By constructing the subsequences and its partial-order, we give an efficient way to determine whether and when any set of detected correlated connections from an incomplete set of stepping stones could be serialised deterministically.

6.1 Serialisation based on local happening before relation

Given any particular stepping stone x, let $S(x)$ be the set of correlated connections that either originate from or terminate at stepping stone x, and let PE-Adj(x) be the set of adjacent correlated connection pairs that is collected around stepping stone x. Apparently the adjacent correlated connection pairs in PE-Adj(x) consist of correlated connections in $S(x)$. Because each connection in $S(x)$ has an unique time of initial arrival or departure, those connections in $S(x)$ can be serialised by the local timestamp at stepping stone x. Let $\angle(x)$ be the binary relation on $S(x)$ that represents the *happen-before* relation between any two connections in $S(x)$. That is, $< e_i, e_j > \in \angle(\mathrm{x})$ iff e_i happens before e_j at stepping stone x. Apparently $\angle(x)$ is a well-order on $S(x)$.

Given an incomplete set of stepping stones $\{x_1, \ldots, x_m\}$ of a connection chain, let $\angle(\{x_1, \ldots, x_m\})$ be the transitive closure of $\bigcup_{1 \le i \le m} \angle(x_i)$. Here $\angle(\{x_1, \ldots, x_m\})$ represents all the order information obtained from the incomplete set of adjacent correlated connection pairs, and apparently it is antisymmetric. Therefore $\angle(\{x_1, \ldots, x_m\})$ is a partial-order on $\bigcup_{1 \le i \le m} S(x_i)$.

Figures 7 and 8 show two examples of serialisation with incomplete set of observed correlated connections with limited observing area. In specific, the observing area of Figure 7(a) includes host 1, 2 and 4, and its corresponding set of observed correlated connections is $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$. The sets of adjacent correlated connection pairs at host 2 and 4 are $\{ < e_1, e_2 > , < e_4, e_5 >, < e_9, e_{10} > \}$ and $\{ < e_3, e_4 >, < e_6, e_7 > , < e_8, e_9 > \}$, respectively. Based on the initial arrival or departure time at host 2, we know $< e_1, e_2 >$ happens before $< e_4, e_5 >$ and $< e_4, e_5 >$ happens before $< e_9, e_{10} >$. Similarly, we know $< e_3, e_4 >$ happens before $<e_6, e_7 >$ and $<e_6, e_7 >$ happens before $< e_8, e_9 >$. Figure 7(b) shows the corresponding $\angle(\{x_1, \ldots, x_m\})$, and it clearly indicates that $\angle(\{x_1, \ldots, x_m\})$ is indeed a partial order. To be specific, $\angle(\{x_1, \ldots, x_m\})$ obtained from the incomplete set of adjacent correlated connection pairs lacks order information between e_2 and e_3, e_5 and e_6. However, since e_2 happens before e_4 and e_3 happens right before e_4, e_2 must happen before e_3. Similarly we know e_5 happens before e_6 as e_5 happens right after e_4. Therefore, the set of correlated connection observed from host 1, 2 and 4 in figure 7(b) can be serialised deterministically into sequence: $e_1e_2 \ldots e_3e_4e_5 \ldots e_6e_7 \ldots e_8e_9e_{10}$ based on the happen right before (or after) information.

Figure 7 Deterministic serialisation with incomplete set of observed correlated connections

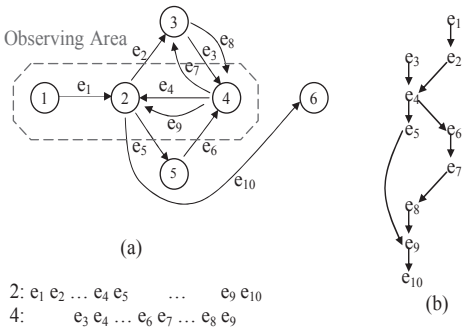
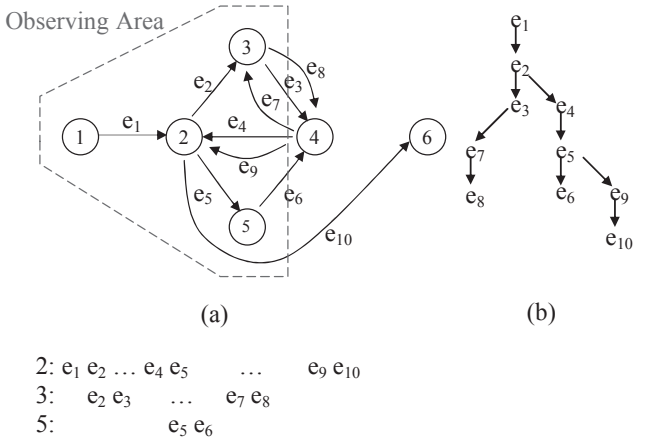


Figure 8 shows another example, and the observing area of Figure 8(a) includes host 1, 2, 3 and 5. The corresponding set of observed correlated connections is $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$ as well. The sets of adjacent correlated connection pairs at host 2, 3 and 5 are $\{ < e_1, e_2 >, < e_4, e_5 >, < e_9, e_{10} > \}, \{ < e_2, e_3 >, < e_7, e_8 > \}$ and $\{ < e_5, e_6 > \}$, respectively. Based on the initial arrival or departure time at host 2, we know $< e_1, e_2 >$ happens before $< e_4, e_5 >$ and $< e_4, e_5 >$ happens before $< e_9, e_{10} >$. Similarly, we know $< e_2, e_3 >$ happens before $< e_7, e_8 >$ at host 3. Figure 8(b) shows the partial order $\angle(\{x_1, \ldots, x_m\})$, and it indicates that the relative orders between e_3 and e_4, e_6 and e_7, e_8 and e_9 are missing.

By merging those adjacent correlated connections (based on the happen right before relation obtained from the set of adjacent correlated connection pairs), we get the following subsequences: $e_1e_2e_3$, $e_4e_5e_6$, e_7e_8 and e_9e_{10}. Based on the initial arrival or departure time at host 2,

3 and 5, we know $e_1e_2e_3$ happens before $e_4e_5e_6$, $e_4e_5e_6$ happens before e_9e_{10} and $e_1e_2e_3$ happens before e_7e_8, but we do not have enough information to determine the relative order between $e_4e_5e_6$ and e_7e_8, e_7e_8 and e_9e_{10}. In fact, each of the serialisations $e_1e_2e_3 \ldots e_7e_8 \ldots e_4e_5e_6 \ldots e_9e_{10}$, $e_1e_2e_3 \ldots e_4e_5e_6 \ldots e_7e_8 \ldots e_9e_{10}$ and $e_1e_2e_3 \ldots e_4e_5\ e_6\ \ldots\ e_9\ e_{10}\ \ldots\ e_7\ e_8$ satisfies all the local orders and adjacencies of correlated connections observed at host 1, 2, 3 and 5. Therefore, the serialisation of the incomplete set of observed correlated connections at host 1, 2, 3 and 5 is non-deterministic.

Figure 8 Non-deterministic serialisation with incomplete set of observed correlated connections



The above two examples demonstrate that while the happen right before (or after) information obtained from the incomplete set of adjacent correlated connection pairs helps to serialise the correlated connections, it does not guarantee the deterministic serialisation. It is important to understand under what condition, the order information obtained from the incomplete set of adjacent correlated connection pairs is enough to serialise those observed correlated connections deterministically. To determine such a condition, we model the happen right before (or after) information through the subsequence of connection chain.

6.2 Subsequences of connection chain

Given an incomplete set of adjacent correlated connections pairs PE-Adj of a connection chain, PE-Adj is no longer guaranteed to be edge one-way connected. That is, there exists $e_i, e_j \in E_{\text{PE-Adj}}$ such that there is no paired edge connectivity between e_i and e_j. For example, PE-Adj of Figure 7(a) is $\{ < e_1,\ e_2 >, < e_3,\ e_4 >, < e_4,\ e_5 >, < e_6,\ e_7 >, < e_8,\ e_9 >, < e_9,\ e_{10} > \}$, there is no paired edge connectivity between e_2, e_3 while there is paired edge connectivity between e_3, e_5. Therefore, the incomplete set of adjacent correlated connection pairs could be, based on binary relation $<_{\text{PEC}}$, partitioned into one or more subsets of adjacent correlated connection pairs, where each subset is pair edge one-way connected. For example, the set of all observed correlated connection pairs of Figure 7(a) is partitioned into 4 subsets: $\{< e_1,\ e_2 >\}$, $\{< e_3,\ e_4 >, < e_4,\ e_5 >\}$, $\{< e_6,\ e_7 >\}$ and $\{< e_8,\ e_9 >, < e_9,\ e_{10} >\}$. According to Theorem 3, each subset of adjacent correlated connection pairs could be serialised deterministically into a sequence of connections based on binary relation $<_{\text{PEC}}$. We call the sequence of each subset of adjacent correlated connection pairs a *subsequence* of the connection chain. Therefore, an incomplete set of adjacent correlated connection pairs could be uniquely transformed into a set of subsequences of connections, where each subsequence represents a fragment of the connection chain. For example, the sets of subsequences for the incomplete sets of adjacent correlated connection pairs PE-Adj of Figures 7(a) and 8(a) are $\{e_1e_2,\ e_3e_4e_5,\ e_6e_7,\ e_8e_9e_{10}\}$ and $\{e_1e_2e_3,\ e_4e_5e_6,\ e_7e_8,\ e_9e_{10}\}$, respectively.

The following theorem describe the 'tightness' property of the subsequences of the correlated connections.

Theorem 4: *Given any two subsequences* $< e_{i,1}, \ldots, e_{i,l_i} >$ *and* $< e_{j,1}, \ldots, e_{j,l_j} >$ *derived from an incomplete set of adjacent correlated connection pairs of a connection chain, if any connection* $e_{i,k_1} \in < e_{i,1}, \ldots, e_{i,l_i} >$ *happens before any connection* $e_{j,k_2} \in < e_{j,1}, \ldots, e_{j,l_j} >$*, then every connection in* $< e_{i,1}, \ldots, e_{i,l_i} >$ *happens before every connection in* $< e_{j,1}, \ldots, e_{j,l_j} >$.

For example, in Figure 7(a), we know e_1 happens before e_4, based on the subset of adjacent correlated connection pairs collected around stepping stone 2 and 4, we know e_1e_2 and $e_3e_4e_5$ are two subsequences. Because e_2 happens right after e_1, and e_3 happens right before e_4, we know e_2 happens before e_3.

In summary, correlated connection set $E_{\text{PE-Adj}}$ could, based on the incomplete set of adjacent correlated connection pairs PE-Adj, be partitioned into one or more subsequences of connections where all connections in each subsequences are well ordered. In order to serialise all the connections in $E_{\text{PE-Adj}}$ deterministically, we just need to serialise the subsequences deterministically.

6.3 Relative order of subsequences

Theorem 4 states that if any connection in subsequence 1 happens before any connection in another subsequence 2, then every connection in subsequence 1 happens before every connection in subsequence 2. This property gives us a way to define the relative order of subsequences.

Assume the incomplete set of adjacent correlated connection pairs $\bigcup_{1 \le i \le m}$ PE-Adj (x_i) is transformed into $n > 0$ subsequences based on PEC, where each subsequence $< e_{i,1}, \ldots, e_{i,l_i} >$ consists of l_i connections $(1 \le i \le n)$. We define *subsequence-order* (denoted as $\angle_{\text{seq}}$) on all subsequences as binary relation such that subsequence $< e_{i,1}, \ldots, e_{i,l_i} > \angle_{\text{seq}} < e_{j,1}, \ldots, e_{j,l_j} >$ if and only if any connection in $< e_{i,1}, \ldots, e_{i,l_i} >$ happens before any connection in $< e_{j,1}, \ldots, e_{j,l_j} >$.

Because $\angle_{\text{seq}}$ is based on happen before relation, it is inherently antisymmetric. Therefore, binary relation $\angle_{\text{seq}}$ is a partial-order on set of all subsequences. Because all the connections within each subsequence are well ordered, the necessary and sufficient condition for the deterministic serialisation of those observed correlated connections is that $\angle_{\text{seq}}$ is a well order.

Therefore, in order to determine if the set of correlated connections observed from an incomplete set of stepping stones could be serialised deterministically, we just need to do

1 collect the subset of adjacent correlated connection pairs around each observed stepping stone

2 collect the local order of those correlated connections observed from each individual stepping stone

3 derive the set of subsequences based binary relation PEC

4 derive the partial-order $\angle_{\text{seq}}$ and

5 check if $\angle_{\text{seq}}$ is a well-order.

Depending on the relative order of connections within $\bigcup_{1\le i\le m} S(x_i)$ and incomplete set of adjacent correlated connection pairs $\bigcup_{1\le i\le m}$ PE-Adj (x_i), partial-order $\angle(\{x_1, \ldots, x_m\})$ may or may not be a well- order. For example, in Figure 7(a), the subsequences are e_1e_2, $e_3e_4e_5$, e_6e_7, $e_8e_9e_{10}$ and the corresponding $\angle_{\text{seq}} = \{ < e_1e_2,\ e_3e_4e_5 >, < e_1e_2,\ e_6e_7 >,\ < e_1e_2,\ e_8e_9e_{10} >,\ < e_3e_4e_5, e_6e_7 >,\ < e_3\ e_4\ e_5,\ e_8\ e_9\ e_{10} >,\ < e_6e_7,\ e_8e_9e_{10} > \}$ which happens to be a well-order that serialises all the subsequences into sequence $e_1e_2 \ldots e_3e_4e_5 \ldots e_6e_7 \ldots e_8e_9e_{10}$. In Figure 8(a), the subsequences are $e_1e_2e_3, e_4e_5e_6,\ e_7e_8,\ e_9e_{10}$, and the corresponding $\angle_{\text{seq}} = \{ < e_1e_2e_3,\ e_4e_5e_6 >,\ < e_1e_2e_3,\ e_7e_8 >,\ < e_1e_2e_3,\ e_9e_{10} >,\ < e_4e_5e_6,\ e_9e_{10} > \}$ is not a well-order as there is no relative order between subsequences e_7e_8 and $e_4e_5e_6$, e_7e_8 and e_9e_{10}.

In summary, the incomplete set of adjacent correlated connection pairs does not have enough information to guarantee the deterministic serialisation of those observed correlated connections. In general, the incomplete set of adjacent correlated connection pairs could partition the set of correlated connections into one or more subsequences within which all connections are well ordered. The relative order between those subsequences that is derived from the incomplete set of adjacent correlated connection pairs is guaranteed to be a partial-order. The set of subsequences and the partial-order of those subsequences contain all possible and only those serialisations of the observed correlation connections that satisfy all the local relative orders observed at each observed stepping stone.

7 Conclusions

Tracing the source and identifying the attack path of the network based attacks launched behind stepping stones are challenging problems, especially when the intrusion connection chain passes some stepping stone multiple times in an attempt to further disguise its intrusion path and source.

In this paper, we used set theory approach to analyse the theoretical limit of the correlation solution in solving the stepping stone tracing problem, and we obtained a number of fundamental results that apply to any stepping stone tracing and correlation solutions. We first identified the largely overlooked serialisation problem and the loop fallacy in tracing intrusion connections though stepping stones. Existing approaches to the tracing problem of stepping stones have focused on correlation only and have left the serialisation of correlated connections as an afterthought. We demonstrated that even the perfect correlation solution, which gives all and only those correlated connections, is not sufficient to construct the complete intrusion path deterministically, when there is loop in the intrusion connection chain. We further showed that the complete intrusion path can be constructed deterministically from the complete set of adjacent correlated connection pairs, no matter whether there is any loop in the connection chain or not. We presented an efficient algorithm to construct the set of correlated connection pairs and effective method to serialise correlated connections without global clock synchronisation.

We further considered the case when the tracing system only sees an incomplete set of correlated connections and stepping stone due to its limited observing area. We demonstrate that

1 the incomplete set of adjacent correlated connection pairs plus local orders of correlated connection at each (but not all) stepping stone are, in general, not adequate to deterministically serialise those incomplete set of observed correlated connections

2 they could partition the set of observed correlated connections into one or more subsequences and derive a partial-order of those subsequences and

3 The set of subsequences and the partial-order of those subsequences contain all possible and only those serialisations of the observed correlation connections that satisfy all the local relative orders of connections at each observed stepping stone.

In case the partial-order of those subsequences is a well order, those incomplete set of observed correlated connections could indeed be serialised deterministically based on the incomplete set of correlated connection pairs.

Our analytical results have revealed new insights into the difficulty of the stepping stone tracing problem, and have illustrated how some simple countermeasures by the network based attackers could make the intrusion source tracing genuinely difficult.

References

Blum, A., Song, D. and Venkataraman, S. (2004) 'Detection of interactive stepping stones: algorithms and confidence bounds', *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004) LNCS-3224*, Springer, October pp.258–277.

Donoho, D., Flesia, A.G., Shanka, U., Paxson, V., Coit, J. and Staniford, S. (2002) 'Multiscale stepping stone detection: detecting pairs of jittered interactive streams by exploiting maximum tolerable delay', *Proceedings of the Fifth International Symposium on Recent Advances in Intrusion Detection (RAID 2002) LNCS-2516*, Springer, October pp.17–35.

Goodrich, M.T. (2002) 'Efficient packet marking for large-scale IP traceback', *Proceedings of the Ninth ACM Conference on Computer and Communications Security (CCS 2002)*, ACM, October pp.117–126.

Jung, H., et al. (1993) 'Caller identification system in the internet environment', *Proceedings of the Fourth USENIX Security Symposium*, pp.69–78.

Kohno, T., Broido, A. and Claffy, K. (2005) 'Remote physical device fingerprinting', *Proceedings of IEEE Symposium on Security and Privacy*, IEEE, pp.211–225.

Li, J., Sung, J., Xu, J. and Li, L. (2004) 'Large scale IP traceback in high-speed internet: practical techniques and theoretical foundation', *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, IEEE, pp.115–129.

Machover, M. (1996) *Set Theory, Logic and Their Limitations*, Cambridge University Press.

Savage, S., Wetherall, D., Karlin, A. and Anderson, T. (2000) 'Practical network support for IP traceback', *Proceedings of the ACM SIGCOMM 2000*, ACM, September pp.295–306.

Snapp, S., et al. (2001) 'DIDS (Distributed intrusion detection system) – motivation, architecture and early prototype', *Proceedings of the 14th National Computer Security Conference*, pp.167–176.

Snoeren, A., et al. (2001) 'Hash-based IP traceback', *Proceedings of ACM SIGCOMM 2001*, ACM, September, pp.3–14.

Staniford-Chen, S. and Heberlein, L.T. (1995) 'Holding intruders accountable on the internet', *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pp.39–49.

Stoll, C. (2000) *The Cuckoo's Egg: Tracking Spy through the Maze of Computer Espionage*, Pocket Books, October.

Wang, X. (2004) 'The loop fallacy and serialization in tracing intrusion connections through stepping stones', *Proceedings of the 19th ACM Symposium on Applied Computing (SAC 2004 Track on Computer Security)*, March, pp.404–411.

Wang, X. and Reeves, D.S. (2003) 'Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays', *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS 2003)*, October, pp.20–29.

Wang, X., Reeves, D.S. and Wu, S.F. (2002) 'Inter-packet delay-based correlation for tracing encrypted connections through stepping stones', D. Gollmann, G. Karjoth and M. Waidner, (Eds). *Seventh European Symposium on Research in Computer Security (ESORICS'2002) LNCS-2502*, October, pp.244–263.

Wang, X., Reeves, D.S., Wu, S.F. and Yuill, J. (2001a) 'Sleepy watermark tracing: an active network-based intrusion response framework', *Proceedings of 16th International Conference on Information Security (IFIP/Sec'01)*, June, pp.369–384.

Yoda, K. and Etoh, H. (2000) 'Finding a connection chain for tracing intruders', F. Guppens, Y. Deswarte, D. Gollmann and M. Waidner, (Eds). *Sixth European Symposium on Research in Computer Security (ESORICS'2000) LNCS-1895*, October, pp.191–205.

Yung, K.H. (2002) 'Detecting long connection chains of interactive terminal sessions', *Proceedings of the Fifth International Symposium on Recent Advances in Intrusion Detection (RAID'2002) LNCS-2516*, October, pp.1–16.

Zhang, Y. and Paxson, V. (2001) 'Detecting stepping stones', *Proceedings of the Ninth USENIX Security Symposium*, pp.171–184.

Notes

[1]Since we assume we have a perfect correlation solution that gives us all and only those connections in a connection chain.

[2]We will define subsequence later in the section.

Appendix

Proof of Theorem 1:

Sufficiency:– Given that DG has no directed cycles, $<_{PC}$ is antisymmetric: $\forall u, v \in V[u <_{PC} v \Rightarrow \neg(v <_{PC} u)]$. Because DG is one-way connected, $<_{PC}$ is transitive. Therefore $<_{PC}$ is a partial- order on V.

Given DG is one-way connected, $\forall u, v \in V(u \neq v)$, there exists a directed path either $u \rightarrow v$ or $v \rightarrow u$. We have either $u <_{PC} v$ or $v <_{PC} u$. Therefore, $<_{PC}$ is a total-order on V.

Assume $<_{PC}$ is not a well-order on V, then there exists a non-empty set of vertices $V' \subseteq V$ such that V' does not have PC-minimal. That is $\forall v \in V', \exists u \in V'$ such that $u <_{PC} v$. We list elements of V', starting from $\forall v_1 \in V'$, and adding $v_{i+1} \in V'$ to the left of $v_i \in V'$ if $v_{i+1} <_{PC} v_i$ and $v_{i+1} \notin \{v_i, v_{i-1} \ldots v_1\}$ as following:

$$v_n \ldots v_{i+1} v_i \ldots v_2 v_1$$

Because V' is finite, the above list is also finite. Assume the left-most element of above list is v_n, we have $v_i (1 \leq i < n)$ such that $v_i <_{PC} v_n$, therefore $< v_i, v_n, \ldots v_i >$ forms a directed cycle in G. This contradicts condition 2). Therefore $<_{PC}$ well-orders V.

Necessity:

1 Because $<_{PC}$ is well-order on V, it is total-order on V. $\forall u, v \in V(u \neq v)$, we have either $u <_{PC} v$ or $v <_{PC} u$. Then there exists a path either u to v or $v \rightarrow u$. Therefore DG is one-way connected.

2 Assume DG has directed cycle of $n > 1$ vertices: $v_n \ldots v_2 v_1$, consider non-empty subset of $V\{v_n \ldots v_2 v_1\}$, there is no PC-minimal in that set. This contradicts the prerequisite that $<_{PC}$ well-orders V. Therefore DG has no directed cycle.

Proof of Theorem 2:

Sufficiency: – Given $\forall < u_1, v_1 >, < u_2, v_2 > \in E$ and $< u_1, v_1 > \neq < u_2, v_2 >$, we have $u_1 \neq u_2$ because of 3.

Assume $v_1 = v_2$. Consider $u_1, u_2 \in V$, because of 1, there exists path: $u_1 \rightarrow u_2$. Because of 3 we have $v_1 \rightarrow u_2$, that is $v_2 \rightarrow u_2$. Then we have a cycle $< v_2, u_2, v_2 >$, and it contradicts condition 2. Therefore $v_1 \neq v_2$.

Assume $v_1 \rightarrow u_2$, because of condition 2, there is no path from u_2 to v_1 (otherwise we have a loop). Because of condition 3, no path from u_2 to v_1 means no path from v_2 to u_1. That is $\forall < u_1, v_1 >, < u_2, v_2 > \in E[< u_1, v_1 > <_{EC} < u_2, v_2 > \Rightarrow \neg(< u_2, v_2 > <_{EC} < u_1, v_1 >)]$. Therefore, $<_{EC}$ is a partial-order on E.

Assume there is neither path from v_1 to u_2 nor path from v_2 to u_1. Because of 1), we have $u_2 \rightarrow v_1$ and $u_1 \rightarrow v_2$. Because of 3), we have $v_2 \rightarrow v_1$ and $v_1 \rightarrow v_2$. That forms a cycle, which contradicts condition 2). Therefore there is either $v_1 \rightarrow u_2$ or $v_2 \rightarrow u_1$. That is equivalent to either $< u_1, v_1 > <_{EC} < u_2, v_2 >$ or $< u_2, v_2 > <_{EC} < u_1, v_1 >$. Therefore $<_{EC}$ is a total-order on E.

Assume $<_{EC}$ is not well-order on E, then there exists a non-empty set $E' \subseteq E$ such that there is no *EC-minimal* on E'. That is $\forall < u_1, v_1 > \in E', \exists < u_2, v_2 > \in E'$ such that $< u_2, v_2 > <_{EC} < u_1, v_1 >$. We list elements E', starting from $\forall < u_1, v_1 > \in E'$, and adding $< u_{i+1}, v_{i+1} > \in E'$ to the left of $< u_i, v_i > \in E'$ if $< u_{i+1}, v_{i+1} > <_{EC} < u_i, v_i >$ and $< u_{i+1}, v_{i+1} > \notin \{ < u_i, v_i >, < u_{i-1}, v_{i-1} > \ldots < u_1, v_1 > \}$ as the following:

$$<u_n, v_n> \ldots <u_{i+1}, v_{i+1}><u_i, v_i> \ldots <u_1, v_1>$$

Because E' is finite, the above list is also finite. Assume the left-most element of above list is $< u_n, v_n >$, we have $< u_i, v_i > (1 \leq i < n)$ such that $< u_i, v_i > <_{EC} < u_n, v_n >$, therefore $<< u_i, v_i >, < u_n, v_n >, \ldots < u_i, v_i >>$ forms a directed cycle in DG. This contradicts condition 2). Therefore $<_{EC}$ well-orders E.

Necessity:

1 $\forall u, v \in V(u \neq v)$, there exists $e_1, e_2 \in E(e_1 \neq e_2)$ such that u is endpoint of e_1 and v is endpoint of e_2. Without losing generality, we assume that $e_1 =< u, x >$ and $e_2 =< v, y >$. Because $<_{EC}$ well-orders E, it total-orders E. Therefore either $e_1 <_{EC} e_2$ or $e_2 <_{EC} e_1$. There exists path either from u to v or from v to u in G.

2 Assume G has directed cycle of $n > 1$ edges: $< v_1, v_2 >, < v_2, v_3 >, \ldots < v_n, v_1 >$, consider non-empty subset of $E\{ < v_1, v_2 >, < v_2, v_3 >, \ldots < v_n, v_1 > \}$, there is no *EC-minimal* in that set. This contradicts with the prerequisite that $<_{EC}$ well-orders E. Therefore DG has no directed cycles.

3 Assume DG has out-branch: $\exists < u, x >, < u, y > \in E(x \neq y)$. Because $<_{EC}$ well-orders E, we have either $< u, x > <_{EC} < u, y >$ or $< u, y > <_{EC} < u, x >$. Without losing generality, we assume $< u, x > <_{EC} < u, y >$, then there exists a path $x \rightarrow u$. $\{x, \ldots u, x\}$ forms a cycle, which contradicts the necessary condition 2 just proved. Therefore DG has no out-branch: $\forall v \in V$ (v has single successor).

Proof of Theorem 3

Because PE-Adj is edge one-way connected, $<_{PEC}$ total-orders $E_{PE\text{-}Adj}$.

Assume DG=$< V_{PE\text{-}Adj}, E_{PE\text{-}Adj} >$, consider the paired line graph of DG: PL(DG)=$< V, E >$, where $V = E_{PE\text{-}Adj}$ and $E =$ PE-Adj. $<_{PEC}$ total-orders $E_{PE\text{-}Adj}$ corresponds to $<_{PC}$ on V total-orders V. Therefore PL(DG) is one-way connected.

Because PE-Adj is loopless, $\forall v \in V$, it would not reach v again in PL(DG). That is PL(DG) has no directed cycle.

Apply theorem 1, $<_{PC}$ well-orders V on PL(DG), which corresponds to $<_{PEC}$ well-orders $E_{PE\text{-}Adj}$.

Proof of Theorem 4:

By definition, all connections in a subsequence are edge one-way connected with each other and connections from different subsequences are not edge one-way connected.

$e_{i,k}$ is adjacent to $e_{i,k+1}$ $(1 \leq k < l_i - 1)$, which means $e_{i,k}$ happens right before $e_{i,k+1}$ or $e_{i,k+1}$ happens right after $e_{i,k}$. Since e_{i,k_1} happens before e_{j,k_2}, e_{i,k_1+1} must also happen before e_{j,k_2} as e_{i,k_1+1} happens right after e_{i,k_1}. By induction, e_{i,l_i} happens before e_{j,k_2} . Similarly, e_{i,l_i} happen before e_{j,k_2-1} as e_{j,k_2-1} happens right before e_{j,k_2}. By induction, e_{i,l_i} happens before $e_{j,1}$. By transitivity, every connection in subsequence $< e_{i,1}, \ldots, e_{i,l_i} >$ happens before every connection in subsequence $< e_{j,1}, \ldots, e_{j,l_j} >$.