Inter-Packet Delay Based Correlation for Tracing Encrypted Connections through Stepping Stones

Xinyuan Wang[1], Douglas S. Reeves[1,2], and S. Felix Wu[3] *

[1] Department of Computer Science
[2] Department of Electrical and Computer Engineering
North Carolina State University
{xwang5,reeves}@eos.ncsu.edu
[3] Department of Computer Science, University of California at Davis
wu@cs.ucdavis.edu

Abstract. Network based intrusions have become a serious threat to the users of the Internet. Intruders who wish to attack computers attached to the Internet frequently conceal their identity by staging their attacks through intermediate "stepping stones". This makes tracing the source of the attack substantially more difficult, particularly if the attack traffic is encrypted. In this paper, we address the problem of tracing encrypted connections through stepping stones. The incoming and outgoing connections through a stepping stone must be correlated to accomplish this. We propose a novel correlation scheme based on inter-packet timing characteristics of both encrypted and unencrypted connections. We show that (after some filtering) inter-packet delays (IPDs) of both encrypted and unencrypted, interactive connections are preserved across many router hops and stepping stones. The effectiveness of this method for correlation purposes also requires that timing characteristics be distinctive enough to identify connections. We have found that normal interactive connections such as telnet, SSH and rlogin are almost always distinctive enough to provide correct correlation across stepping stones. The number of packets needed to correctly correlate two connections is also an important metric, and is shown to be quite modest for this method.

1 Introduction

Network-based intrusions have become a serious threat to users of the Internet. Today, perpetrators can attack networked information systems from virtually anywhere in the world.

One major problem in apprehending and stopping network-based intrusion is that the intruders can easily hide their identity and point of origin through

* This work was supported by AFOSR contract F49620-99-1-0264 and by DARPA contract F30602-99-1-0540. The views and conclusions contained herein are those of the authors.

D. Gollmann et al. (Eds.): ESORICS 2002, LNCS 2502, pp. 244–263, 2002.

readily available means. One of the techniques most commonly used by intruders is to hide their origin by connecting across multiple stepping stones [4, 9, 13] before attacking the final targets. For example, an attacker logged into host A may telnet into host B, and from there launch an attack on host C. An analysis of the traffic at C will only reveal it is being attacked from B, but will not identify the actual source of the attack. A careful inspection of the contents of the traffic coming into and going out of B may reveal that A is the source of the attack. However, if the traffic arriving at B is encrypted (using SSH [6, 11] or IPSEC [3]) before being transmitted to C, it will not be possible to use the traffic contents for correlation purposes. Network-based intruders thus have an easy way to launch attacks without revealing their identity. Without a means of effectively and quickly tracing the source of an attack back to its source, it will not be possible to stop further attacks or punish those who are responsible.

In this paper, we address the problem of correlating the incoming and outgoing connections of a stepping stone. The goal is to identify which connections are part of an attack path, so that the attack can be traced back to its source. We assume that attack traffic may be encrypted at any stepping stone in an attempt to interfere with correlation. We propose a novel scheme based on the inter-packet timing characteristics of both encrypted and unencrypted connections. While, as with most intrusion tracing and detection systems, out correlation scheme could be evaded by highly sophisticated intruders, it is our goal to make it difficult to do so and thus deter network-based intrusions.

The remainder of the paper is organized as follows. In section 2, we give a summary of related works. In section 3, we formulate the correlation problem of our focus and give our correlation problem solution model. In section 4, we discuss IPD (Inter-Packet Delay) based correlation in detail. In section 5, we evaluate correlation effectiveness of our proposed correlation metrics through experiments. In section 6, we conclude with summary of our findings.

2 Related Work

Most of the existing work on correlating connections across stepping stones assumes the traffic is unencrypted. In general, attack tracing approaches can be categorized as either being host-based or network-based. In a host-based approach, the stepping stone itself participates in the tracing, while in the network-based approaches, the stepping stones are not used for tracing purposes. Based on how the traffic is traced, tracing approaches can further be classified as either active or passive. Passive approaches monitor and compare all traffic, allowing any traffic to be traced at any time. On the other hand, active approaches dynamically control when, where, what and how the traffic is to be correlated, through customized packet processing. They only trace the traffic of interest when needed. Table 1 provides a classification of existing tracing approaches, as well as our proposed tracing mechanism.

The Distributed Intrusion Detection System (DIDS) [8] developed at UC Davis is a host-based tracing mechanism that attempts to keep track of all the

Table 1. Classification of Correlation and Tracing Approaches

	Passive	Active
Host-Based	DIDS [8] CIS [2]	
Network-Based	Thumb printing [9] ON/OFF-Based [13] Deviation-Based [12]	IDIP [7] SWT [11] IPD-Based (proposed method)

users in the network and report all activities to network-wide intrusion detection systems. The Caller Identification System (CIS) [2] is another host-based tracing mechanism. It eliminates centralized control by utilizing a truly distributed model. Each host along the connection chain keeps a record about its view of the connection chain so far.

A fundamental problem with the host-based tracing approach is its trust model. Host-based tracing places its trust upon the monitored hosts themselves. In specific, it depends on the correlation of connections at every host in the connection chain. If one host is compromised and is providing misleading correlation information, the whole tracing system is fooled. Because host-based tracing requires the participation and trust of every host involved in the network-based intrusion, it is very difficult to be applied in the context of the public Internet.

Network-based tracing is the other category of tracing approaches. It does not require the participation of monitored hosts, nor does it place its trust on the monitored hosts. Rather, it is based on the assumption that one or more properties of a connection chain is maintained throughout the chain. In particular, the thumbprint [9] is a pioneering correlation technique that utilizes a small quantity of information to summarize connections. Ideally it can uniquely distinguish a connection from unrelated connections and correlate successfully those connections in the same connection chain.

Sleepy Watermark Tracing (SWT) [11] applies principles of steganography and active networking in tracing and correlating unencrypted connections through stepping stones. By injecting watermarks in the traffic echoed back to the attacker, SWT is able to trace and correlate even a single keystroke by the intruder. By actively generating tracing traffic, it can trace and correlate even when an intrusion connection is idle.

IDIP (the Intrusion Identification and Isolation Protocol) [7] is a part of Boeing's Dynamic Cooperating Boundary Controllers Program that uses an active approach to trace the incoming path and source of the intrusion. In this method, boundary controllers collaboratively locate and block the intruder by exchanging intrusion detection information, namely, attack descriptions. The ON/OFF-based scheme [13] by Zhang and Paxson is the first correlation intended to correlate traffic across stepping stones even if the traffic is encrypted

by the stepping stone. The method is based on correlation of the ends of OFF periods (or equivalently the beginnings of ON periods) of interactive traffic, rather than the connection contents. While it is robust against payload padding, ON/OFF-based correlation requires that the packets of connections have precise, synchronized timestamps in order to be able to correlate them. This makes correlations of measurements taken at different points in the network difficult or impractical.

The deviation-based approach [12] by Yoda and Etoh is another network-based correlation scheme. It defines the minimum average delay gap between the packet streams of two TCP connections as the deviation. The deviation-based approach considers both the packet timing characteristics and their TCP sequence numbers. It does not require clock synchronization 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 payload, e.g., when certain types of encryption are used.

Both ON/OFF-based and deviation-based approaches define their correlation metrics over the entire duration of the connections to be correlated. This makes the correlation applicable to post-attack traces only.

The IPD-based approach we discuss in the remainder of the paper defines its correlation metric over the sliding window of packets of the connections to be correlated. This enables it to correlate both live traffic (at real-time) and collected traces (post-attack). It supports distributed correlation of traffic measure at different points of network and is robust against payload padding.

3 Problem Formulation

A *connection* c_i (also called a flow) is a single connection from computer host H_i (the source) to host H_{i+1} (the destination). A user may log into a sequence of hosts H_1, H_2, ... H_{n+1} through a connection chain c_1, c_2, ... c_n, where connection c_i is a remote login from host H_i to host H_{i+1}.[1] The *tracing problem* is, given connection c_n, to determine the other connections c_1, c_2, ... c_{n-1} in the chain, and only those connections. From these connections the identities of all the hosts in the chain, including H_1, may be directly determined.[2]

3.1 Correlation Problem Solution Model

Let $\hat{C}$ represent the set of all connections being examined. We can define an *ideal correlation function* $\mathrm{CF} : \hat{C} \times \hat{C} \rightarrow \{0, 1\}$ such that $\mathrm{CF}(c_i, c_j) = 1$ iff c_i

[1] The same host may appear in a connection chain more than once, in which case the chain contains a loop. Due to space limitations we do not consider this case here.

[2] If IP spoofing is used, of course, the packets of a connection will incorrectly identify the source of the connection. We consider this problem to be orthogonal to our problem and do not address it here. It is, in any event, unlikely that IP spoofing will be used for interactive traffic, where the response to the interactive traffic must be correctly echoed back to the source.

and c_j are in the same connection chain, otherwise $\mathrm{CF}(c_i, c_j) = 0$. To solve the tracing problem we must find such a function CF.

In practice, connection correlation is based on the characteristics of the connections, which may include packet contents, header information (such as packet size) and packet arrival and/or departure times. The connection characteristics can be modeled by a metric function of the connection

$$M : \hat{C} \times P \rightarrow Z \tag{1}$$

where $\hat{C}$ is the set of connections to be correlated, P is some domain of parameters and Z is the correlation metric domain. Based on the connection metric, a *correlation value function* (CVF) can be defined as

$$\mathrm{CVF} : Z \times Z \rightarrow [0, 1] \tag{2}$$

where the result of CVF is a real number between 0 and 1. To approximate CF through CVF, we introduce a threshold $0 \leq \delta \leq 1$ such that c_i and c_j are considered correlated iff

$$\mathrm{CVF}(M(c_i, p), M(c_j, p)) \geq \delta \tag{3}$$

Therefore the tracing problem is now replaced by the following: *find or construct M, p, CVF and δ such that*

$$\begin{gathered} \forall c_i, c_j \in \hat{C}, \mathrm{CVF}(M(c_i, p), M(c_j, p)) \geq \delta \\ \text{iff } c_i \text{ and } c_j \text{ are in the same connection chain} \end{gathered} \tag{4}$$

In finding M, p, CVF and δ, the key is to identify those unique characteristics of connections that are invariant across routers and stepping-stones. If those identified invariant characteristics of connections are distinctive enough to exclude other unrelated connections, reliable correlation of connections can be constructed from these metrics.

4 IPD Based Correlation of Encrypted Connections

In principle, correlation of connections is based on inherent characteristics of connections. To correlate potentially encrypted connections, the key is to identify a correlation metric from the connection characteristics that is: 1) invariant across routers and stepping stones; 2) not affected by encryption and decryption; 3) unique to each connection chain. Potential candidates for the correlation metric of a flow of packets include header information, packet size, inter-packet timing etc. In particular, inter-packet timing should not be affected by encryption and decryption. We now present an original correlation method based on inter-packet delays or IPDs.

4.1 General IPD Based Correlation Model

The overall IPD correlation of two connections is a two-step process. First, the two connections to be correlated are processed to generate a number of *correlation points* between the two connections. Second, these generated correlation points are evaluated to obtain the *correlation value* of the two connections.

The rationale behind this two-step process is to support the true real-time correlation, which is the capability to correlate "live" traffic when they come and go. This means that the approach must be able to correlate connections before their ends are reached. Therefore, the correlation metric for true real-time correlation cannot be defined over the entire duration of a connection; we choose instead to compute it over a window of packets in the connection. A correlation point generated from IPDs within the window reflects some localized similarity between the two flows; the correlation value obtained from all the correlation points will indicate the overall similarity of the two flows.

Basic IPD Correlation Concepts and Definitions Given a bi-directional connection, we can split it into two unidirectional flows. We define our correlation metric over the unidirectional flow of connections.

Given a unidirectional flow of $n > 1$ packets, we use t_i to represent the timestamp of the i^{th} packet observed at some point of the network. We assume all the t_i's of a flow are measured at the same observation point with the same clock. We define the i^{th} *inter-packet delay* (IPD) as

$$d_i = t_{i+1} - t_i \tag{5}$$

Therefore, for any flow consisting of $n > 1$ packets, we can measure the inter-packet delay (IPD) vector $\langle d_1, \ldots, d_{n-1} \rangle$. Ideally, the IPD vector would uniquely identify each flow and we could construct our correlation metric from the IPD vectors. To support real-time correlation based on the IPD vector, we define the *IPD correlation window* $W_{j,s}$ on $\langle d_1, \ldots, d_n \rangle$ as

$$W_{j,s}(\langle d_1, \ldots, d_n \rangle) = \langle d_j, \ldots, d_{j+s-1} \rangle \tag{6}$$

where $1 \leq j \leq n$ represents the starting point of the window, and $1 \leq s \leq n-j+1$ is the size of the window.

Given any two flows X and Y, whose IPD vectors are $\langle x_1, \ldots, x_m \rangle$ and $\langle y_1, \ldots, y_n \rangle$ respectively, we define a *Correlation Point Function* CPF over IPD correlation windows of $X : W_{j,s}(X)$ and of $Y : W_{j+k,s}(Y)$ as

$$\text{CPF}(X, Y, j, k, s) = \phi(W_{j,s}(X), W_{j+k,s}(Y)) \tag{7}$$

where ϕ is a function of two vectors: $R^s \times R^s \rightarrow [0, 1]$, $1 \leq j \leq \min(m-s+1, n-k-s+1)$ is the start of the IPD correlation window, $-j+1 \leq k \leq n-j-s+1$ is the offset between the two IPD correlation windows, and $1 \leq s \leq \min(m, n)$ is the size of the two IPD correlation windows. $\text{CPF}(X, Y, j, k, s)$ quantitatively

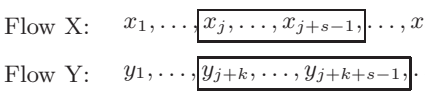

Fig. 1. CPF in PD Correlation Windows $W_{j,s}(X)$ and $W_{j+k,s}(Y)$

expresses the correlation between $W_{j,s}(\langle x_1, \ldots, x_m \rangle)$ and $W_{j+k,s}(\langle y_1, \ldots, y_n \rangle)$ as shown in Figure 1.

Because the value of $\mathrm{CPF}(X, Y, j, k, s)$ changes as j and k changes, we can think of $\mathrm{CPF}(X, Y, j, k, s)$ as a function of j and k. Given any particular value of j, $\mathrm{CPF}(X, Y, j, k, s)$ may have a different value for each different value of k. We are interested in the maximum value of $\mathrm{CPF}(X, Y, j, k, s)$ for any particular value of j. We define $(j, j + k)$ as a correlation point if

$$\max_{-j+1 \le k \le n-j-s+1} \mathrm{CPF}(X, Y, j, k, s) \ge \delta_{cp} \tag{8}$$

where δ_{cp} is the correlation point threshold with value between 0 and 1. The δ_{cp} here is for detecting correlation point and is different from δ in inequality (3).

We further define k for this correlation point $(j, j+k)$ as the *correlation-offset* of $\mathrm{CPF}(X, Y, j, k, s)$ and the correlation point.

Given flow X, Y, correlation window size s and threshold δ_{cp}, by applying formula (8), we can obtain a series of correlation points: $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$ where $n \ge 0$.

Assuming one packet of flow X corresponds to one packet of flow Y[3], if flow X and Y are really part of the same connection chain, the IPDs of flow X should have a one-to-one correspondence with the IPDs of flow Y. In this case, all the correlation points should have same correlation-offset. This can be formally represented with CPF as

$$\exists k' \forall j \left[\mathrm{CPF}(X, Y, j, k', s) = \max_{-j+1 \le k \le n-j-s+1} \mathrm{CPF}(X, Y, j, k, s) \right] \tag{9}$$

That is there exists an offset k' such that $\mathrm{CPF}(X, Y, j, k', s)$ is closest to 1 for all possible j. In this case, all the correlation points (x, y) will be covered by a linear function $y = x + k'$.

After obtaining $n > 0$ correlation points: $(j_1, j_1+k_1), (j_2, j_2+k_2), \ldots, (j_n, j_n+k_n)$, we represent those n correlation points with two n-dimensional vectors $C_x = \langle j_1, \ldots, j_n \rangle$ and $C_y = \langle j_1 + k_1, \ldots, j_n + k_n \rangle$. The Correlation Value Function formula (3) is now transformed to

$$\mathrm{CVF}(C_x, C_y) \ge \delta \tag{10}$$

In summary, the metric function M in formula (1) is now mapped to CPF, the parameter domain P in formula (1) is mapped to s and δ_{cp}, and Z in formula (1) is mapped to n-dimensional vectors $C_x = \langle j_1, \ldots j_n \rangle$ and $C_y = \langle j_1 + k_1, \ldots, j_n + k_n \rangle$.

[3] We have found this is true for most packets in correlated flows.

Correlation Method Assessment Criteria A critical issue in this method is the choice of the function ϕ for computing the correlation point function CPF. Criteria by which the method may be judged include:

- Uniqueness of perfect correlation: for a flow X, no flow Y not in the same connection chain as X should have $\mathrm{CPF}(X, Y, j, k, s) = 1$.
- Correlation Point (CP) *true positives (hits)*: this is the number of packets that should be correlated, and are found to be correlated according to equation 8. The true positive rate is the number of hits divided by the number of packets that should be correlated.
- Correlation Point (CP) *false positives (misses)*: this is the number of packets which are not in fact correlated, but which are found to be correlated according to equation 8.

Ideally, we would expect a perfect correlation method 1) has unique perfect correlation; 2) has a 100% CP true positive rate; and 3) has 0 misses or false positives.

4.2 Correlation Point Functions

We now propose four correlation point functions, each of which enjoys certain advantages or applicability, as discussed below.

Min/Max Sum Ratio (MMS) One simple metric to quantitatively express the ôsimilarityö between two vectors is the ratio between the summation of the minimum elements and the summation of the maximum elements.

$$\mathrm{CPF}(X, Y, j, k, s)_{MMS} = \frac{\sum_{i=j}^{j+s+1} \min(x_i, y_{i+k})}{\sum_{i=j}^{j+s+1} \max(x_i, y_{i+k})} \tag{11}$$

$\mathrm{CPF}(X, Y, j, k, s)_{MMS}$ has the range $[0, 1]$ and takes the value 1 only when $x_i = y_{i+k}$ for $i = j, \ldots, j+k-1$. Therefore, $\mathrm{CPF}(X, Y, j, k, s)_{MMS}$ is likely to exhibit unique perfect correlation.

Statistical Correlation (STAT) Based on the concept of the coefficient of correlation from statistics [1], we can define

$$\mathrm{CPF}(X, Y, j, k, s)_{Stat} = \begin{cases} \rho(X, Y, j, k, s) & , \rho(X, Y, j, k, s) \geq 0 \\ 0 & , \rho(X, Y, j, k, s) < 0 \end{cases} \tag{12}$$

where

$$\rho(X, Y, j, k, s) = \frac{\sum_{i=j}^{j+s+1} (x_i - E(X)) \times (y_{i+k} - E(Y))}{\sqrt{\left[\sum_{i=j}^{j+s+1} (x_i - E(X))^2\right] \times \left[\sum_{i=j}^{j+s+1} (y_{i+k} - E(Y))^2\right]}}$$

The range of $\mathrm{CPF}(X,Y,j,k,s)_{Stat}$ is also $[0,1]$. Unlike $\mathrm{CPF}(X,Y,j,k,s)_{MMS}$, for a given $W_{j,s}(X)$, there may be more than one value of $W_{j+k,s}(Y)$ for which $\mathrm{CPF}(X,Y,j,k,s)_{Stat}$ has the value 1. For example, for a particular $W_{j,s}(X)$, any linear transform of $W_{j,s}(X) : W_{j+k,s}(Y) = a \times W_{j,s}(X) + b$ will result in $\mathrm{CPF}(X,Y,j,k,s)_{Stat}$ being equal to 1 ($a > 0$) or -1 ($a < 0$). $\mathrm{CPF}(X,Y,j,k,s)_{Stat}$ is therefore less likely to exhibit unique perfect correlation, and is more likely to result in false positives.

Normalized Dot Product 1 (NDP1) In digital signal processing, linear correlation (or matched filtering) of two discrete signals will reach a maximum at the point where the signals have the most similarity. It is well known that linear correlation is optimal in detecting the similarity between a discrete signal and the corresponding signal distorted by additive, white Gaussian noise. However the range of linear correlation is not necessarily between 0 and 1.

If the discrete signals are replaced by two vectors, the corresponding operation to linear correlation of signals is the *inner-product* or *dot-product* of two vectors in n-dimensional space. From linear algebra, the inner-product of two n-dimensional vectors is equal to the cosine of the angle between the two vectors, multiplied by the lengths of the two vectors. That is:

$$W(X) \bullet W(Y) = \cos(\theta) \times |W(X)| \times |W(Y)| \tag{13}$$

where θ is the angle between vector $W(X)$ and $W(Y)$, and $|W(X)|$ and $|W(Y)|$ are the lengths of vector $W(X)$ and $W(Y)$ respectively[4].

$\cos(\theta)$ in (13) can be used as a correlation point function. The range of $\cos(\theta)$ is $[-1,1]$ and it provides a measure of the similarity of two vectors. For any vector $W_{j,s}(X)$, $\cos(\theta)$ will be 1 for any vector $W_{j+k,s}(Y) = a \times W_{j,s}(X) + b$.

To make the correlation point function more likely to exhibit unique perfect correlation, we can define it as follows:

$$\begin{aligned}
\mathrm{CPF}(X,Y,j,k,s)_{NDP1} &= \frac{\min(|W(X)|,|W(Y)|}{\max(|W(X)|,|W(Y)|} \times \cos(\theta) \\
&= \frac{\min(|W(X)|,|W(Y)|)}{\max(|W(X)|,|W(Y)|} \times \frac{\sum_{i=j}^{j+s-1} x_i \times y_{i+k}}{|W(X)| \times |W(Y)|} \\
&= \frac{\sum_{i=j}^{j+s-1} x_i \times y_{i+k}}{[\max(|W(X)|,|W(Y)|)]^2} \\
&= \frac{\sum_{i=j}^{j+s-1} x_i \times y_{i+k}}{\max\left(\sum_{i=j}^{j+s-1} x_i^2, \sum_{i=j}^{j+s-1} y_{i+k}^2\right)}
\end{aligned} \tag{14}$$

As x_i and y_i are non negative, the range of $\mathrm{CPF}(X,Y,j,k,s)_{NDP1}$ is [0, 1]. It can be shown that $\mathrm{CPF}(X,Y,j,k,s)_{NDP1}$ will be 1 only when $W_{j,s}(X) = W_{j+k,s}(Y)$. Therefore, $\mathrm{CPF}(X,Y,j,k,s)_{NDP1}$ is likely to exhibit unique perfect correlation.

[4] We have dropped the subscripts of W(X) and W(Y) for clarity purposes in this section.

Normalized Dot Product 2 (NDP2) Another way to normalize the dot-product of two vectors is

$$\mathrm{CPF}(X,Y,j,k,s)_{NDP2} = \frac{\sum_{i=j}^{j+s-1} x_i \times y_{i+k}}{\sum_{i=j}^{j+s-1} [\max(x_i, y_{i+k})]^2} \tag{15}$$

Because x_i and y_i are non negative, the range of $\mathrm{CPF}(X,Y,j,k,s)_{NDP2}$ is $[0,1]$. It is obvious that $\mathrm{CPF}(X,Y,j,k,s)_{NDP2}$ equals 1 only when $W_{j,s}(X) = W_{j+k,s}(Y)$.

Among these four proposed correlation point functions, Mini/MaxSum Ratio (MMS) is likely to be the most sensitive to local details of the IPD vectors to be correlated. This is because it does not average any differences, and it accumulates all the IPD differences. As a result, MMS may potentially have a lower true positive rate due to its emphasis on local details. While the STAT CPF is much more robust to noise, we expect it to have substantially more false positives. The normalized dot product functions (NDP1 and NDP2) are likely to be in between MMS and STAT in terms of sensitivity to local detail and robustness to noise.

4.3 Correlation Value Function

Given flows X, Y, correlation window size s and threshold δ, by applying formula (8), we can potentially obtain a set of correlation points: $(j_1, j_1 + k_1), (j_2, j_2 + k_2), \ldots, (j_n, j_n + k_n)$. We represent this sequence of correlation points through two n-dimensional vectors $C_x = \langle j_1, \ldots j_n \rangle$ and $C_y = \langle j_1 + k_1, \ldots, j_n + k_n \rangle$.

We define the overall *Correlation Value Function* CVF of flows X and Y from this sequence of correlation points, as follows:

$$\mathrm{CVF}(C_x, C_y) = \begin{cases} 0 & n = 0 \\ \rho(C_x, C_y) & n = 1 \\ 1 & n = 1 \end{cases} \tag{16}$$

where

$$\rho(C_x, C_y) = \frac{\sum_{i=1}^{n} (j_i - E(C_x)) \times (j_i + k_i - E(C_y))}{\sqrt{[\sum_{i=1}^{n} (j_i - E(C_x))^2] \times [\sum_{i=1}^{n} (j_i + k_i - E(C_y))^2]}}$$

$\mathrm{CVF}(C_x, C_y)$ quantitatively expresses the overall correlation between flows X and Y, and its value range is $[-1, 1]$. When there exists more than one correlation point and all the correlation points have same correlation offset (i.e., $k_1 = k_2 = \ldots k_n$), $\mathrm{CVF}(C_x, C_y) = 1$. When flow X and Y has only one correlation point, $\mathrm{CVF}(C_x, C_y) = 1$. When flow X and Y have no correlation point, $\mathrm{CVF}(C_x, C_y)$ is defined to be 0.

4.4 Limitations and Countermeasures

The effectiveness of IPD-based correlation relies on the uniqueness of IPDs of connections. It may be ineffective at differentiating uncorrelated connections

that exhibit very similar IPD patterns, such as file transfers accomplished via FTP. Interactive traffic, as we show later in the experiments section, usually has IPD patterns that are distinctive enough for our purposes.

Intruders could thwart IPD-based correlation by deliberately changing the inter-packet delays of a connection in a chain. Such delays may be designed to reduce the true positive rate, or increase the number of false positives. There are relatively simple means of accomplishing such traffic shaping, although they may require kernel-level manipulations. The amount of delay that can be added by the intruder is limited by the maximum delay that is tolerable for interactive traffic. Another countermeasure against IPD-based correlation is to set some connection into line mode while keeping other connection in character mode. This could potentially merge several packets into one bigger packet. However, the server side shell could always dynamically turn off the line mode [10]. Other countermeasures include:

- Injecting "padding" packets that can be removed by the application
- Segmenting one flow into multiple flows and reassembling them, again at the application level

It is an area of future work to address such countermeasures.

5 Experiments

The goal of the experiments is to answer the following questions about IPD based correlation:

1. Are inter-packet delays preserved through routers and stepping stone, and to what extent?
2. Are inter-packet delays preserved across encryption/decryption and various network applications (such as telnet/rlogin/SSH)?
3. How effective is the IPD-based correlation metric in detecting connections which are part of the same chain?
4. How well does the IPD-based correlation metric differentiate a connection from connections that are not part of the same chain?

5.1 Correlation Point Experiment

To answer the first two questions, we have conducted the following experiment. We first telnet'ed from a Windows 2000 PC behind a cable modem connected to an ISP in North Carolina to a Sun workstation via VPN. From the workstation, we used SSH to login to another workstation at N. C. State University. We then telnet'ed to a PC running Linux at UC Davis, and from there we SSH'ed back to a PC running FreeBSD at NC State. As shown in Figure 2, the connection chain has a total of 3 stepping-stones and 59 hops and covers a distance on the order of 10,000 *km*. The connection chain consists of links of different speeds – including residential Internet access, typical campus LAN and public Internet

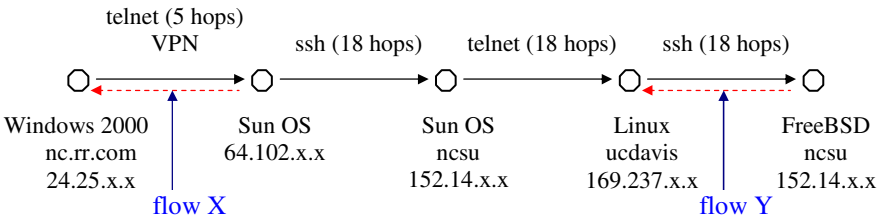

Fig. 2. Correlation Experiment on telnet and ssh

backbone. We have captured the packet traces at the Windows 2000 node and the FreeBSD node; both traces have a timestamp resolution of 1 microsecond. We label the telnet return path[5] flow from the Sun workstation to the Windows 2000 PC flow X, and the SSH backwards flow from the FreeBSD PC to the Linux PC flow Y. Therefore, flow X consists of telnet packets and flow Y consists of SSH packets.

We have calculated the IPD vectors after filtering out the following sources of errors:

- Duplicated packets
- Retransmitted packets
- ACK only packets

We then calculated correlation points (j, k) by applying (8) using each of the four correlation point functions, with different correlation window sizes s and correlation point thresholds δ_{cp}.

Figure 3 shows the correlation points between flow X and Y obtained by the MMS CPF with different correlation window sizes s and thresholds δ_{cp}. In these plots, a point at position (j, k) indicates inequality (8) was true for that value of j, k, s and δ_{cp}. True positives are points located along the major diagonal. False positives are points located off the major diagonal.

With correlation window size of 5 and δ_{cp} threshold of 0.70, there are a large number of falsely detected correlation points (false positives) in addition to the true correlation points (true positives). The overall CVF value (equation (4.13)) for this case is 0.1707. With larger correlation window size, or a higher threshold δ_{cp}, MMS results in fewer false positives and has a higher CVF value, as would be expected. At correlation window size 15, and a threshold δ_{cp} of 0.70, MMS detects most of the true correlation points between flow X and Y, finds no false positives, and has an overall CVF value of 0.9999. When the threshold δ_{cp} is increased to 0.95 with the same correlation window size of 15, MMS detects substantially fewer correlation points between flow X and Y, with no false positives, and has an overall CVF value of 1.0. This suggests that with correlation window

[5] The "return path" is the echoed traffic generated on the destination host and sent to the origination host.

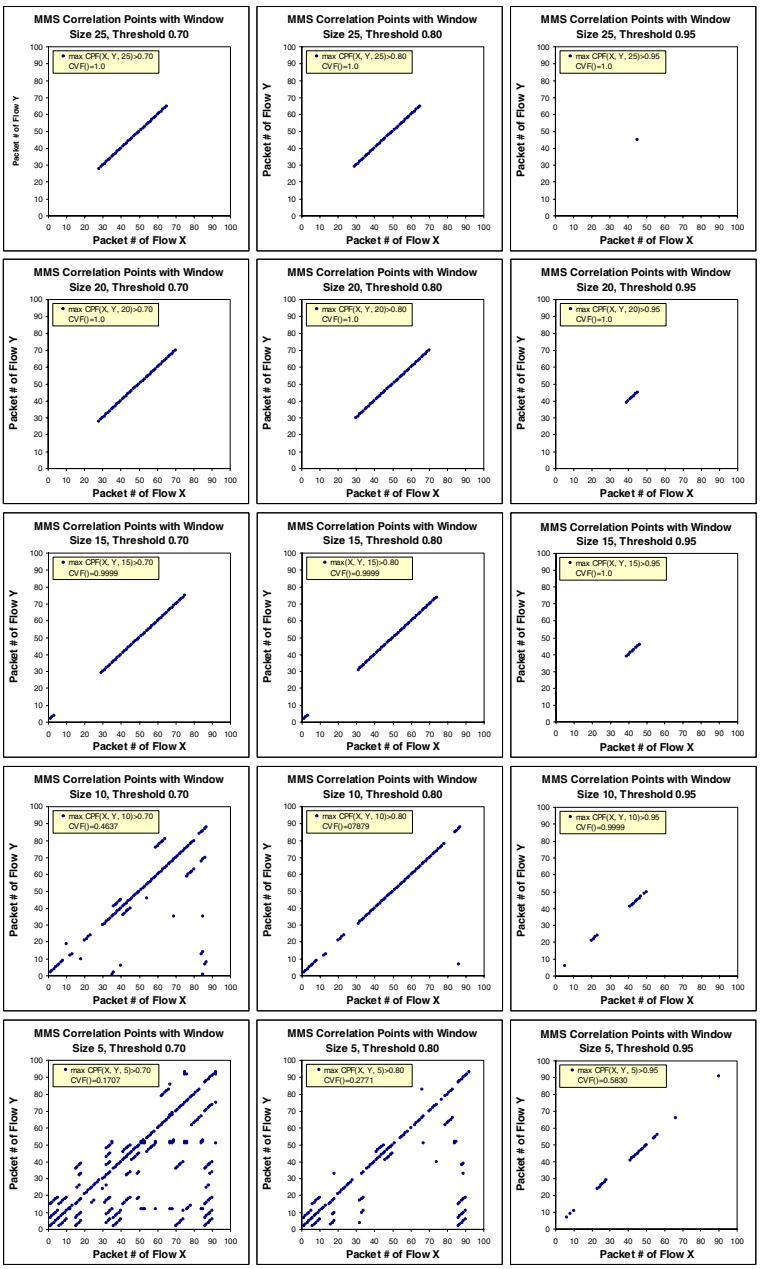

Fig. 3. Correlation Point between Two Correlated Flows Detected by MMS with Different Correlation Window Sizes and Thresholds

Table 2. Traces of Flows Used in Correlation Experiments

Flow Set	Date	Flow Type	Flow #	Packet #
FS1	03/02/02	SSH	16	12372
FS2	03/02/02	Telnet	15	5111
FS3	02/20/01	Telnet/SSH	144	34344
FS4	02/26/01	Telnet/SSH	135	38196
FS5	05/xx/02	SSH	400	364158

size 15, the threshold δ_{cp} of 0.95 is probably too high for MMS. The correlation points missed by correlation windows size 15 and threshold δ_{cp} of 0.95 are actually due to correlation-offset shifts. The correlation-offset between our sample flows X and Y has shifted 3 times between the start and finish. This indicates that a telnet packet may trigger more than one SSH packet, or vice versa. Fortunately, such correlation-offset shifts are infrequent between correlated flows. Generally, a larger correlation window size is very effective in filtering out false positives, and a higher threshold δ_{cp} tends to filter out both true positive and false positive correlation points. An excessively large correlation window size with a high threshold δ_{cp} tends to have a low true positive rate, due to both correlation-offset shifts and IPD variations introduced by the network.

Figure 4 compares the detected correlation points between flow X and Y by different CPFs: MMS, STAT, NDP1 and NDP2, with identical correlation window sizes of 10 and threshold δ_{cp} of 0.80. As expected, the statistical CPF results in substantially more false positives than the other three CPFs. While NDP2 has slightly fewer false positives than NDP1, they both have somewhat more false positives than MMS. Generally, MMS is very sensitive to localized details of IPDs and is able to accurately correlate the flows using a smaller correlation window (i.e. 5). NDP1 and NDP2 are less effective with a small correlation window, but they are able to correlate effectively with a moderate window size (15 or 20). The statistical CPF appears to fail to consider enough localized details to correlate accurately

5.2 Aggregated Flow Correlation Experiment

To evaluate more generally the performance of the different correlation point functions, we have used five sets of flows (Table 2). FS1 and FS2 were collected at two ends of connection chains similar to the scenario shown in Figure 2. FS1 and FS2 contain 16 SSH flows and 15 Telnet flows, respectively; for each flow in FS2, there is one flow in FS1 which was in the same connection chain. FS3 and FS4 are derived from 5 million packet headers and 12 million packet headers of the Auckland-IV traces of NLANR [5]. FS5 is derived from over 49 million packet headers of the Bell Lab-I traces of NLANR.

We have conducted four sets of aggregated correlation experiments. For all of these experiments, two flows were regarded as being correlated if the CVF

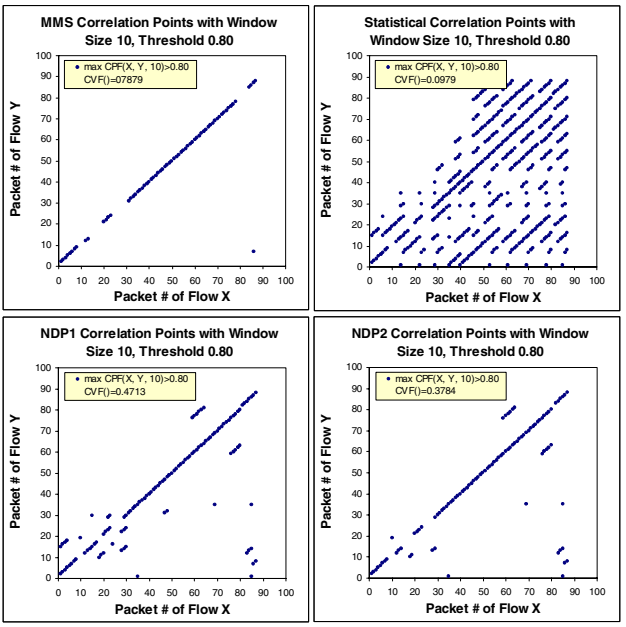

Fig. 4. Correlation Points Detected by MMS, Stat, NDP1 and NDP2 with Same Window Size and Threshold

of their correlation points (equation (16)) was greater than $\delta = 0.6$. The first set tests how well different correlation metrics (CPF) detect correlation between sets FS1 and FS2. Figure 5 shows both the number of true positives (out of a total of 15) and the number of false positives (out of 15×15=225) of flow correlation detection with different correlation window sizes and correlation point thresholds δ_{cp}.

With a δ_{cp} threshold of 0.70, MMS reaches its TP peak of 93% at a correlation window size of 20, and NDP2 reaches its TP peak of 80% with a correlation window size of 20 or 25. However NDP2 has a significantly higher number of false positives at the window size corresponding to its peak true positive rate than does than MMS. Both STAT and NDP1 have very low (<7%) TP rates with all correlation window size. This indicates that STAT and NDP1 are ineffective with a low δ_{cp} threshold.

For all δ_{cp} threshold values, MMS attains its peak TP rate with 0 false positives. NDP1 and NDP2 show a similar success rate, with a somewhat higher failure (false positive) rate. STAT is generally not successful at correlating the flows in the same chain. The best results are obtained for the highest δ_{cp} threshold setting. MMS is able to achieve 100% TP rate with 0 false positives with correlation window size 15, δ_{cp} threshold 0.90 and window size 10, δ_{cp} threshold 0.95. NDP2 is also able to have 100% TP rate with 0 FP at correlation win-

dow size 15, δ_{cp} threshold 0.95. NDP1's overall TP peak is 93% with 7% FP at correlation window size 20, δ_{cp} threshold 0.90.

The second set of experiments shows the correlation detection effectiveness by different correlation metrics. We use combined flow set of FS3 and FS4 (279 flows) and flow set FS5 (400 flows) to correlate themselves respectively. Figure 6 shows the true positive rate of different correlation metric with different correlation window size and δ_{cp} threshold. Again the STAT correlation point function consistently performs poorly. MMS and NDP1 almost have identical correlation detection rates across all the correlation window size and δ_{cp} threshold combinations in both data sets, where NDP1 has little lower detection rate. For flow set FS5, the detection rates of both MMS and NDP2 reach 92% and higher with correlation window size 25 or bigger. At correlation window size 35, MMS's and NDP2's detection rate achieve over 97%. For the combined flow set FS3 and FS4, at a correlation window size of 15, for δ_{cp} threshold 0.95, MMS, NDP1 and NDP2 all have the highest correlation detection rate of 76.7%. This lower detection rate is due to the nature of the flows in FS3 and FS4. We have found a number of SSH flows in FS3 and FS4 show very similar periodicity, with constant very short IPDs. We suspect they are bulk data transfers within the SSH connections. This shows a potential limitation of the use of IPD-based tracing.

The third experiment is intended to evaluate the ability of the different correlation point functions to successfully discriminate flows not part of the same chain. Figure 7 shows the number of false positives (out of 16×279=4464) when correlating FS1 and the combined flow set of FS3 and FS4. Because no flow from FS1 correlates with any flow from FS3 and FS4, any detected correlation by the correlation metric is a false positive. MMS consistently has 0 false positives; and NDP1 and NDP2 false positives decrease as the correlation window size increases. The STAT correlation point function reports an increasing number of FPs with larger correlation sizes.

The fourth experiment similarly investigated the false positive rate, this time between sets FS3 and FS4. Figure 8 shows the results. The number of false positives (out of 144×133 = 19152) for MMS, NDP1 and NDP2 decreases dramatically when the correlation window size increases; that of MMS decreases faster than NDP1 and NDP2. Again, the statistical correlation metric has a consistently higher FP rate with increasing correlation window size. For the MMS method, a window size of 20 0r 25 packets is sufficient to reduce the false positive rate to a fraction of a percent.

In summary, we have found that MMS is very effective in both detecting interactive, correlated flows and differentiating uncorrelated flows with even relatively small correlation window sizes (10, 15). NDP1 and NDP2 are not as sensitive as MMS with small correlation windows; however, they both perform well with larger correlation windows. We have confirmed that the statistical correlation metric is not effective in detecting correlation and differentiating uncorrelated flows.

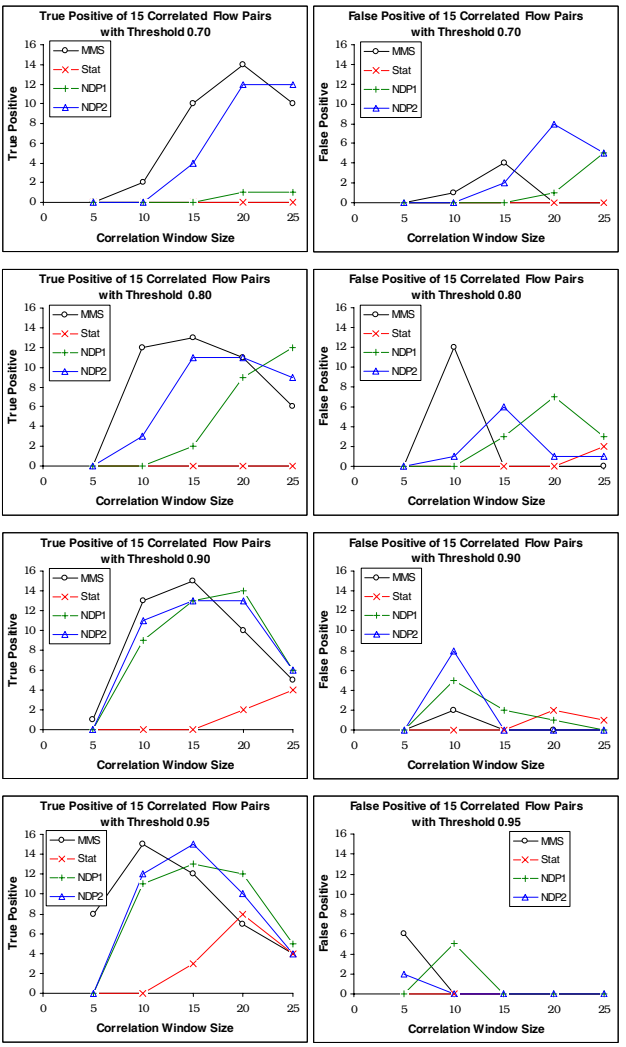

Fig. 5. True Positive and False Positive of Correlation between 16 and 15 Correlated Flows

5.3 Correlation Performance

We have measured the number of calculations of correlation points per second achieved by our unoptimized code. Table 3 shows the average number of millions of correlation point calculation per second of various correlation point functions under different load. Despite dynamic overheads of disk operation, the overall throughput remains largely constant at various loads.

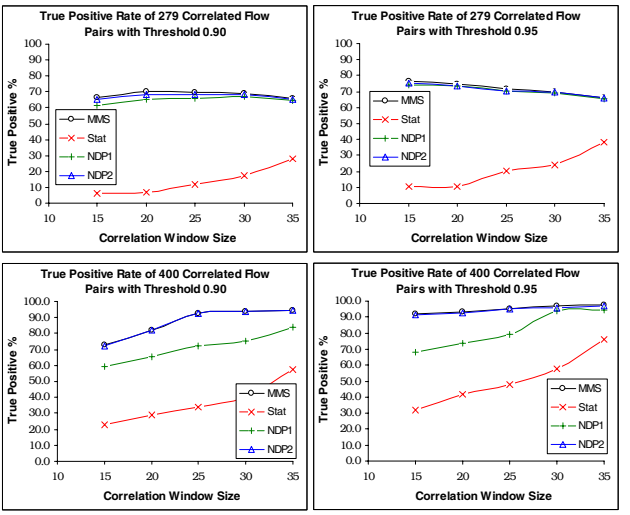

Fig. 6. True Positive Rate of Correlation between 279 and 279, 400 and 400 Correlated Flows

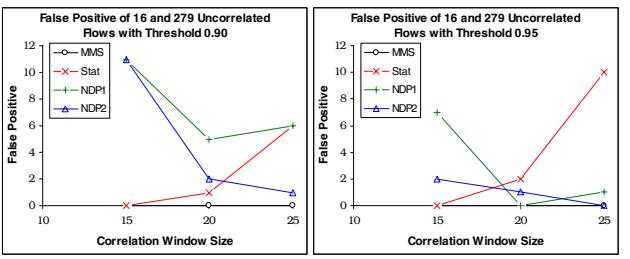

Fig. 7. False Positive of Correlation between 16 and 279 Uncorrelated Flows

Table 3. Throughput (Millions Per Second) of Correlation Point Calculation with Correlation Window Size 15

	40k	379k	937k	2309k	5704k	54210k
MMS	2.00	1.65	1.74	1.75	1.43	1.35
STAT	0.90	0.76	0.69	1.14	1.22	1.83
NDP1	3.99	3.16	2.23	3.25	2.29	3.24
NDP2	1.33	1.31	1.16	1.37	1.17	1.13

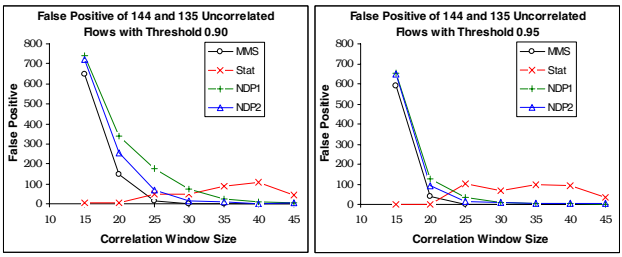

Fig. 8. False Positive of Correlation between 144 and 135 Uncorrelated Flows

6 Conclusions

Tracing intrusion connections through stepping-stones at real-time is a challenging problem, and encryption of some of the connections within the connection chain makes tracing even harder. We have addressed the tracing problem of encrypted connections based on the inter-packet delays of the connections. We proposed and investigated four correlation point functions. Our correlation metric does not require clock synchronization, and allows correlation of measurements taken at widely scattered points. Our method also requires only small packet sequences (on the order of a few dozen packets) for correlation. We have found that after some filtering, IPDs (Inter-Packet Delay) of both encrypted and unencrypted, interactive connections are largely preserved across many hops stepping-stones. We have demonstrated that both encrypted and unencrypted, interactive connections can be effectively correlated and differentiated based on IPD characteristics.

Our experiments also indicate that correlation detection is significantly dependent on the uniqueness of flows. We have found that normal interactive connections such as telnet, SSH and rlogin are almost always unique enough to be differentiated from connections not in the same chain. While bulk data transfer with SSH connection introduces an additional challenge in correlation detection, its impact on correlation differentiation may simply be offset by larger correlation windows and higher correlation point thresholds.

A natural area of future work is to extend the correlation to non-interactive traffic. How to address countermeasures with "bogus packets" and packet splitting and merging remains an open problem.

References

[1] M. H. DeGroot. *Probability and Statistics*. Addison-Wesley, 1989. 251
[2] H. Jung, et al. Caller Identification System in the Internet Environment. In *Proceedings of 4th USENIX Security Symposium*, 1993. 246
[3] S. Kent, R. Atkinson. Security Architecture for the Internet Protocol. *IETF RFC 2401*, September 1998. 245

[4] S. C. Lee and C. Shields. Tracing the Source of Network Attack: A Technical, Legal and Social Problem. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, June 2000. 245

[5] NLANR Trace Archive. `http://pma.nlanr.net/Traces/long/`. 257

[6] OpenSSH. `http://www.openssh.com`. 245

[7] D. Schnackenberg. Dynamic Cooperating Boundary Controllers. `http://www.darpa.mil/ito/Summaries97/E295_0.html`, Boeing Defense and Space Group, March 1998. 246

[8] S. Snapp, et all. DIDS (Distributed Intrusion Detection System) – Motivation, Architecture and Early Prototype. In *Proceedings of 14th National Computer Security Conference*, 1991. 245, 246

[9] S. Staniford-Chen, L. T. Heberlein. Holding Intruders Accountable on the Internet. In *Proceedings of IEEE Symposium on Security and Privacy*, 1995. 245, 246

[10] W. R. Stevens. *TCP/IP Illustrated, Volume 1: The Protocol*. Addison-Wesley, 1994. 254

[11] X. Y. Wang, D. S. Reeves, S. F. Wu and J. Yuill. Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework. In *Proceedings of 16th International Conference on Information Security (IFIP/Sec'01)*, June, 2001. 245, 246

[12] K. Yoda and H. Etoh. Finding a Connection Chain for Tracing Intruders. In *F. Guppens, Y. Deswarte, D. Gollmann and M. Waidner, editors, 6th European Symposium on Research in Computer Security - ESORICS 2000 LNCS-1895*, Toulouse, France, October 2000. 246, 247

[13] Y. Zhang and V. Paxson. Detecting Stepping Stones. In *Proceedings of 9th USENIX Security Symposium*, 2000. 245, 246