Abstract
Tracing attackers’ traffic through stepping stones is a challenging problem, especially when the attack traffic is encrypted, and its timing is manipulated (perturbed) to interfere with traffic analysis. The random timing perturbation by the adversary can greatly reduce the effectiveness of passive, timing-based correlation techniques. We presented a novel active timing-based correlation approach to deal with random timing perturbations. By embedding a unique watermark into the inter-packet timing, with sufficient redundancy, we can make the correlation of encrypted flows substantially more robust against random timing perturbations. Our analysis and our experimental results confirm these assertions. Our watermark-based correlation is provably effective against correlated random timing perturbation as long as the covariance of the timing perturbations on different packets is fixed. Specifically, the proposed watermark-based correlation can, with arbitrarily small average time adjustment, achieve arbitrarily close to 100% watermark detection (correlation true positive) rate and arbitrarily close to 0% collision (correlation false positive) probability at the same time against arbitrarily large (but bounded) random timing perturbation of arbitrary distribution (or process), as long as there are enough packets in the flow to be watermarked.

Keywords
Network-Level Security and Protection, Intrusion Tracing, Correlation, Stepping Stone

I. Introduction
Another common and effective countermeasure used by network-based intruders to hide their identity is to connect through a sequence of intermediate hosts, or stepping stones, before attacking the final target. For example, an attacker at host A may Telnet or SSH into host B, and from there launch an attack on host C. In effect, the incoming packets of an attack connection from A to B are forwarded by B, and become outgoing packets of a connection from B to C. The two connections or flows are related in such a case. The victim host C can use IP trace back to determine the second flow originated from host B, but trace back will not be able to correlate that with the attack flow originating from host A. To trace attacks through a stepping stone, it is necessary to correlate the incoming traffic with the outgoing traffic at the stepping stone. This would allow the attacker to be traced back to host A in the example. The earliest work on connection correlation was based on tracking user’s login activities at different hosts [5,11]. Later work relied on comparing the packet contents, or payloads, of the connections to be correlated [2-3]. Most recent work has focused on the timing characteristics [1-3,5,8], of connections, in order to correlate encrypted connections (i.e. traffic encrypted using IPSec [2] or SSH [4,8]). Timing based correlation approaches, however, are sensitive to the use of countermeasures by the attacker, or adversary. In particular, the attacker can perturb the timing characteristics of a connection by selectively or randomly introducing extra delays when forwarding packets at the stepping stones. This kind of timing perturbation will adversely affect the effectiveness of any timing-based correlation. Timing perturbation can either make unrelated flows have similar timing characteristics, or make related flows exhibit different timing characteristics. This will increase the correlation false positive rate, or decrease the correlation true positive rate, respectively. Previous timing-based correlation approaches are passive in that they simply examine (but do not manipulate) the traffic timing characteristics for correlation purposes. While passive approaches are simple and easy to implement, they may be vulnerable to active countermeasures by the attacker, and/or require a large number of packets in order to correlate timing-perturbed flows. In this paper, we address the random timing perturbation problem in correlating encrypted connections through step ping stones. Our goal is to develop an efficient correlation scheme that is probabilistically robust against random timing perturbation, and to answer fundamental questions concerning the effectiveness of such techniques and the tradeoffs involved in implementing them. We propose a novel watermark-based correlation scheme that is designed specifically to be robust against timing perturbations by the adversary. Unlike most previous correlation approaches, our watermark-based approach is active; that is, it embeds a unique watermark into the encrypted flows by slightly adjusting the timing of selected packets. The unique watermark that is embedded in the encrypted flow gives us a number of advantages over passive timing based correlation in overcoming timing perturbations by the adversary. First, our active watermark based correlation does not make any limiting assumptions about the distribution or random process of the original inter-packet timing of the packet flow, or the distribution of random delays an adversary can add. This is in contrast to existing passive timing based correlation approaches. Second, our method requires substantially fewer packets in the flow to achieve the same level of correlation effectiveness as existing passive timing based correlation. In theory, our watermark based correlation can achieve arbitrarily close to 100% correlation true positive rate and arbitrarily close to 0% false positive rate at the same time for sufficiently long flows, despite arbitrarily large (but bounded) timing perturbation of arbitrary distribution by the adversary. To the best of our knowledge, our work is the first that identifies:
1. The accurate quantitative tradeoffs between the achievable correlation effectiveness and the defining characteristics of the timing perturbation.
2. A provable upper bound on the number of packets needed to achieve a desired correlation effectiveness, given a bound on the amount of timing perturbation. We also investigate the maximum negative impact on the embedded watermark an adversary can have, and the minimum effort needed to achieve that impact. Under the condition that the watermark embedding parameters are unknown to the adversary, we determine the minimum distortion required for the adversary to completely eliminate any embedded watermark from the inter packet timing, and the optimal strategy for doing so. We further investigate the implications of the constraints of real-time communication and bounded delay for the adversary’s ability to remove the embedded watermark. While there exist ways to completely eliminate hidden information from any signal offline, we show that (without knowledge of the
watermark embedding parameters) it is generally infeasible for the adversary to completely eliminate the embedded watermark from the packet timing in real-time, even if he can introduce arbitrarily large (but bounded) distortion to the packet timing of normal network traffic. This result ensures that our watermark-based correlation is able to withstand arbitrarily large timing perturbations in real-time, provided there are enough packets in the flows to be correlated.

II. Existing System

As discussed earlier, the secrecy of the watermark parameters has not been seriously investigated. If an attacker is able to derived these parameters, the attacker can either remove the watermark (i.e., reduce the detection rate) or duplicate the watermark in benign flows (i.e., increase the false positive rate). In both cases, the attacker can successfully defeat the tracing system. In this paper, we take an attacker’s position, aiming at understanding how well an attacker can derive the watermark parameters used by timing based active watermarking through timing analysis. When an attacker uses a series of stepping stone hosts, the attacker has to establish a sequence of connections between adjacent hosts, and application data (e.g., a shell command) is relayed by these connections from the attacker to the final target or vice versa.

An attacker can easily obtain the one-way packet transit delay (abbreviated as packet delay) of a piece of application data as it is forwarded from one host to another. For example, as illustrated in fig. 1, when ha−1 forwards a shell command to ha in a packet, the attacker can include ha−1’s current local time ts along with the command. When he receives this shell command, the attacker can retrieve ts, check the receipt time tr at ha, and calculate the delay as d = tr − ts. The packet delays are the target of our timing analysis. When the stepping stone hosts do not have well synchronized time, the packet delay will include the clock offset (i.e., the difference between the clocks at a specific time) and the clock skew (i.e., the first derivative of the offset). Clock skew is a critical issue since it constantly changes the packet delays. We may use the approach proposed in [7], to handle this problem. That is, we use cumulative minima (or maxima) to identify the skew, and then use linear fit to compute and remove the skew. As a result, clock discrepancy only introduces a constant clock offset into all packet delays after the clock skew is removed. We examined the one-way packet transit delays between computers in the Planet Lab, which are distributed worldwide, a typical distribution of the packet delays from a computer at MIT to a computer on our campus network after the clock skew between these computers are removed. To simplify our analysis, we approximate such distributions with normal distributions. In our experiments, such approximations pass Kolmogorov-Smirnov [6] goodness-of-fit test with a significance level of 0.05, which is the probability that we wrongly reject the normal distribution approximation when it is actually true. It is easy to see that an attacker can observe the packet delays and estimate the parameters (i.e., mean μ and variance σ²) of the delay distribution. To distinguish such packet delays from the delays introduced by watermark embedding, hereafter we call them normal network delays. When a timing-based active watermarking approach is used for trace-back, certain packets will have to be delayed to embed the watermark. Assuming the packets to be delayed come at random time, we can easily derive that the watermark delays (i.e., additional delays of the embedding packets) follow a uniform distribution over [0, 2S), where S is the quantization step. We performed experiments to validate this assumption. The results pass Kolmogorov-Smirnov goodness-of-fit test for uniform distribution with a significance level of 0.05. When a watermark is embedded, the packet delay of each embedding packet should be the combination of the normal network delay and the watermark delay. Let normal random variable X represent the normal network delay, and uniform random variable Y represent the watermark delay. The delta y of the embedding packet is Z = X + Y. We can easily derive the probability density function of Z as:

\[
f_Z(x) = \int_{-\infty}^{+\infty} f_X(y) f_Y(x-y) dy
\]

\[
= \frac{1}{2S \sqrt{2\pi}} \int_{x-2S}^{x} e^{-(y-x)^2/(2S^2)} dy,
\]

where f_Y is the probability density function of a normal distribution with mean μ and variance σ². S is the quantization step. The mean μ is not very important since it does not affect the shapes of f_Z and f_Z', but only moves them along x-axis simultaneously. Fig. 3, shows an example of f_Z and f_Z'. Apparently, when a network flow is watermarked by a trace-back system, some packet delays will follow the distribution of Z, which is different from that of X. (A packet pi delayed by watermark may cause some following packets to be postponed and sent out immediately after pi to keep the correct packet order. These collateral delays can be identified by checking whether there is a large delay that affects its following packets. For simplicity, we do not consider such collateral delays.)

In this paper, we investigate techniques that can take advantage of this observation to detect the existence of timing-based active watermarks, recover the watermark parameters, and remove and duplicate the observed watermarks. In order to fully understand attackers’ threats on the active watermarking scheme, we focus on investigating the following problems:

1. How can an attacker infer the watermark parameters and how much can be recovered?
2. How can an attacker duplicate a watermark to mislead the trace-back system? How well?
3. How can an attacker that connects through stepping stones determine whether he/she is being tracked by a timing-based active
watermarking system as quickly as possible? We choose to first tackle the watermark recovery/duplication problem since it puts a major threat on the watermark scheme. In the following, we assume the attacker obtain the packet delays from two adjacent hosts in the stepping stone connections, and try to compromise the watermark scheme at the second host.

4. Inferring Watermark Parameters As discussed earlier, an attacker can observe the packet delays and obtain the distribution (i.e., $\mu$ and $\sigma$). We also assume the attacker has obtained a sequence of packet delays $d_1, d_2, ..., d_n$ between two stepping stone hosts where a watermark is embedded. However, the attacker does not know the watermark parameters, including the quantization step $S$, the degree of robustness $M$, the length $L$ of the watermark, and the exact watermark bits. The goal of this section is to investigate whether, how, and how well the attacker can recover these parameters. When there is no watermark, the observed packet delays are entirely caused by normal network delays. However, when an watermark exists, some delays will be the combination of both normal network delays and watermark delays. That is, the observed packet delays are drawn from a mixture of two random variables $X$ and $Z$. Thus, the distribution of $d_i$’s is

$$f(x, \theta) = (1 - \theta)f_X(x) + \theta f_Z(x),$$

where $\theta$ is the proportion of $d_i$’s that are from watermark delays. When no watermark is embedded, $\theta = 0$. In the following, we first estimate the quantization step $S$ and the proportion parameter $\theta$, then identify the packets delayed due to watermark, and finally recover the remaining watermark parameters or duplicate the watermark (without knowing all the parameters).

III. Proposed Work

A. Correlation True Positive

This set of experiments aim to compare and evaluate the correlation effectiveness of our proposed active watermark based correlation and previous passive timing-based correlation under various timing perturbations. We used two methods of correlation. First, we used an existing, passive timing-based correlation method called IPD-based correlation, to correlate each flow in FS1 with the same flow, after it is perturbed by various levels of uniformly distributed random timing perturbations. If the flow and the perturbed flow are reported correlated, it is considered as a true positive (“IPDCorr TP”) of the correlation in the presence of timing perturbation. Second, we embedded a random 24-bit watermark into each flow of FS1 and FS2, with redundancy number $m=12$, and quantization step size $s=400$ms for each watermark bit. The embedding of the 24-bit watermark requires 300 packets to be selected, of which 288 were delayed. Third, we randomly perturbed the packet timing of the watermarked flows of FS1 and FS2 with different types of timing. Perturbations. It is considered a true positive (“IPDWMCorr TP”) of watermark-based correlation if the embedded watermark can be detected from the timing perturbed watermarked flows, with a Hamming distance threshold $h=5$. Finally, we calculated the expected watermark detection rate under various maximum delays of uniformly distributed random timing perturbations.

1. Correlation Under Uniformly Distributed Random Timing Perturbation

The measured (as well as expected) true positive rates of IPD-based correlation and watermark-based correlation on FS1 and FS2 under various levels of uniformly distributed random timing perturbations. The results clearly indicate that IPD-based correlation is vulnerable to even moderate random timing perturbations. Without timing perturbation, IPD-based correlation is able to successfully correlate 93% of the SSH flows of FS1. However, with a maximum 100ms random timing perturbation, the true positive rate of IPD-based correlation drops to 45.5%, and with a 200ms maximum delay, the rate drops to 21.5%. In contrast, the proposed watermark-based correlation of the flows in FS1, FS1-Int and FS2 is able to achieve virtually a 100% true positive rate, with up to a maximum 600ms random timing perturbation. With a maximum 1000ms timing perturbation, the true positive rates of watermark-based correlation for FS1, FS1-Int and FS2 are 84.2%, 89.85% and 97.32%, respectively. It can be seen that the measured watermark-based correlation true positive rates are well approximated by the estimated values, based on the watermark detection rate model (equation (15) and (16). In particular, the true positive rate measurements of FS2 are very close to the predicted values at all perturbation levels.

2. Correlation Under Self-Similar Distributed Random Timing Perturbation

To see how our watermark-based correlation works when the random timing perturbation is non-iid, we first investigated correlation under self-similar timing perturbations. We used Glen Kramers implementation of Taqqu et al’s self similar synthetic traffic generating method to generate the self similar timing perturbation with 128 sources of ON/OFF periods aggregated with 30% cumulative load. And we have bounded the timing perturbation through modulo operation. In particular, we have used 128 aggregating sources of ON/OFF periods to generate self-similar delays in units of milliseconds. Fig. 9, shows the measured watermark correlation true positive rates under various bounded self-similar timing perturbation, and expected watermark correlation true positive rates of uniformly distributed random delays, with $h=5$; $l=24$; $m=12$; $s=400$ms. It shows that the bounded self-similar perturbation yields much higher watermark correlation true positive rates than the expected true positive rates of uniformly distributed random delay perturbation with the same delay upper bound (1200ms $\approx$ 2400ms). This indicates that the bounded self-similar timing perturbation has less negative impact than the uniformly distributed random timing perturbation of same upper bound. We calculated the variance of 10000 self-similar delays that are bounded by 1000ms, and obtained $\frac{1}{2}$ selfsim = 46786. However, the variance of a uniformly distributed random variable of range $[0, 1000]$ $\frac{1}{2}$uniform is 83333. The smaller the variance $\frac{1}{2}$ is, the higher is the watermark bit robustness (equivalently, the higher the watermark true positive rate should be). This explains why the self-similar type of random perturbation has less negative impact than the uniformly distributed random delays of same upper bound.

3. Correlation Under Batch-Releasing Random Timing Perturbation:

The second type of non-iid random timing perturbation we investigated is the batch-release timing perturbation. In this model, incoming packets are buffered until expiration of the next timer period, at which point all buffered packets are output at line speed, in a burst. With the batch-release perturbation, the actual delay of any packet depends on where it falls within the timer interval, or window. Assuming the packet arrival times are uniformly distributed within the batch release window, we are able to calculate the variance of the delays over all packets, given the window size or duration. shows the measured and estimated
correlation positive rates of our watermark-based correlation on flow set FS2, under various levels of batch-release timing perturbations. We used 24-bit watermarks with quantization step size \( s = 400 \text{ms} \), redundancy number \( m = 12 \), and Hamming distance threshold \( h = 5 \). The expected true positive rates are calculated. The measured values are close to the expected values, which demonstrate that our analytical model is applicable to noniid timing perturbations.

### B. Correlation False Positive

As mentioned above, there is a non-zero probability that an unwatermarked flow happens to exhibit the randomly chosen watermark. This case is considered a correlation collision, or false positive. According to our correlation collision model, the collision rate is determined by the number of watermark bits \( l \) and the Hamming distance threshold \( h \). We experimentally investigated the following false positive rates for varying values of the Hamming distance threshold \( h \) and the number of watermark bits \( l \):

1. Collision rates between a given flow and 10,000 1,000,000 randomly generated 24-bit watermarks with varying Hamming distance threshold \( h \).
2. Collision rates between a given 24-bit watermark and 10,000 1,000,000 randomly generated (using tcplib) synthetic telnet flows with varying Hamming distance threshold \( h \).
3. Collision rates between a given flow and 100,000 randomly generated watermarks of various lengths with Hamming distance threshold \( h = 5 \).
4. Collision rates between a given watermark of various lengths and 10,000 1,000,000 randomly generated (using tcplib) synthetic telnet flows with Hamming distance threshold \( h = 5 \); the measured and estimated correlation false positive rates. The left sub-figure shows the false positive rates for varying Hamming distance thresholds and fixed length (24-bit) watermarks, and the right sub-figure shows the false positive rates for varying watermark lengths and a fixed Hamming distance threshold \( h = 5 \). The measured values are the average of 100 separate experiments and they are very close to the estimated values. Thus the experimental results validate our model of the collision probability.

### C. Watermark

detection tradeoff experiments equation (15) gives us the quantitative tradeoff between the expected watermark bit robustness, the redundancy number \( m \) and the defining characteristics of the random timing perturbation \( \frac{\lambda}{2} \). With a given watermark bit robustness \( p \), equation (16) gives us the expected watermark detection rate. To verify the validity and accuracy of our tradeoff models of watermark bit robustness and watermark detection rate, we did the following experiments: 1) We embedded a random 24-bit watermark into each flow in FS1 FS1-Int and FS2, with quantization step \( s = 400 \text{ms} \), and varying redundancy numbers \( m = 7; 8; 9; 10; 11; 12 \). After perturbing the watermarked flows with 1000ms maximum random delays, we measured the watermark detection rate of the perturbed, watermarked flows with Hamming distance threshold \( h = 5 \). 2) We also embedded a random 24-bit watermark into each flow in FS1, FS1-Int and FS2, with quantization step \( s = 400 \text{ms} \), redundancy number \( m = 12 \). After perturbing the watermarked flows with 1000ms maximum random delays, we measured the watermark detection rate of the perturbed, watermarked flows for varying Hamming distance thresholds of \( h = 2; 3; 4; 5; 6; 7; 8; 3 \). In a separate experiment, we embedded a random watermark of varying lengths \( l = 18, 19, 20, 21, 22, 23, 24 \) into each flow in FS1, FS1-Int and FS2, with quantization step \( s = 400 \text{ms} \) and redundancy number \( m = 12 \). After perturbing the watermarked flows with 1000ms maximum uniform random delays, we measured the watermark detection rate of the perturbed, watermarked flows with different Hamming distance threshold \( h = 3 \). In all cases, the measured detection rates of FS2 are almost identical to the expected values, and detection rates of FS1 are similar to but lower than the expected values. The detection rates of FS1-Int are always between that of FS1 and FS2. These results validate our quantitative tradeoff models of watermark bit robustness and watermark detection rate.

### D. Comparison with Other Representative Passive Approach

We have also compared the effectiveness and the numbers of packets needed by our active watermark correlation approach, and a representative passive correlation approach [1], under identical levels of timing perturbation on the same sets of traces. For Poisson arrivals, paper [1] claimed that its detection algorithm DETECT-ATTACKS (\( \pm, p \)) is guaranteed to achieve a 100% detection rate and no more than \( \pm \) false positive rate given sufficient number of packets. However, it did not include any experimental results that demonstrated the claimed effectiveness. To empirically compare the effectiveness of our active approach and that of the passive correlation method of [1], we implemented its detection algorithm DETECT-ATTACKS (\( \pm, p \)). After identifying the maximum number of packets \( p \) in any time interval \( \epsilon = 800 \text{ms} \) from each of the 1000 flows in FS2, we applied detection algorithm DETECT-ATTACKS (\( \pm, p \)), with \( \pm = 0.3\% \), to correlate flows in FS2 and the corresponding perturbed flows with maximum 800ms uniformly distributed perturbation. Surprisingly, the detection algorithm DETECT-ATTACKS (\( \pm, p \)) in this test only achieved a 79.5% detection rate, while experiencing no false positives. As shown in section VII-A, under a maximum 800ms uniformly distributed timing perturbation, our watermark-based correlation method achieved at least a 99.9% true positive rate and about a 0.3% false positive rate, using parameter values of \( h = 5, l = 24, s = 400 \text{ms} \) and \( m = 12 \) on flow set FS2. We have further compared the upper bounds of the number of packets needed by our active correlation approach, and by the method of [1]. As an example, consider the requirements to achieve a guaranteed correlation effectiveness of at least a 99.9% true positive rate (TPR) and a 0.35% False Positive Rate (FPR), for a maximum 600ms uniformly distributed timing perturbation. Given the parameter values \( m = 36, s = 400 \text{ms} \), and \( \rho = 600 \text{ms} \), inequality guarantees that the upper bound of the bit error probability of our active approach will be less than 0.0834 (i.e., \( p > 0.9166 \)). Choosing \( l = 32, h = 8, \) and \( p = 0.9166 \), the expected watermark detection rate is \( >99.9\% \), and the expected watermark collision rate is \( 0.35\% \). Therefore, in order to achieve a 99.9% TPR and a 0.35% FPR with a maximum 600ms timing perturbation, our active approach needs to adjust the timing of no more than \( 32 \times 36 = 1,152 \) packets. For the approach of [1], letting \( \epsilon = 600 \text{ms} \), \( \pm = 0.3\% \) and using the value of \( p \) measured from the flows in FS1, detection algorithm DETECT-ATTACKS (\( \pm, p \)) requires at most 16,698 packets to achieve a 100% TPR and a 0.35% FPR. For this target, the upper bound of the number of packets needed by the passive approach of [1] to achieve comparable correlation effectiveness is at least an order of magnitude more than that of the active approach.
IV. Results

**Fig. 2:** Impact of (10 Seconds) Batch-Releasing Timing Perturbation

**Fig. 3:**

**Fig. 4:**

V. Conclusions

Network-based intruders seldom attack their victims directly from their own computer. Often, they stage their attacks through intermediate “stepping stones” in order to conceal their identity and origin. To identify the source of the attack behind the stepping stone(s), it is necessary to correlate the incoming and outgoing flows or connections of a stepping stone. To resist attempts at correlation, the attacker may encrypt or otherwise manipulate the connection traffic. Timing-based correlation approaches have been shown to be quite effective in correlating encrypted connections. However, timing-based correlation approaches are subject to timing perturbations that may be deliberately introduced by the attacker at stepping stones. In this project, our watermark-based approach is “active” in that it embeds a unique watermark into the encrypted flows by slightly adjusting the timing of selected packets. The unique watermark that is embedded in the encrypted flow gives us a number of advantages over passive timing-based correlation in resisting timing perturbations by the attacker. A two-fold monotonically increasing compound mapping is created and proved to yield more distinctive visible watermarks in the watermarked image. Security protection measures by parameter and mapping randomizations have also been proposed to deter attackers from illicit image recoveries.
References


Mr. V. Chandra Kanth is a student of NOVA college of Engineering & Technology, Jangareddygudem. Presently he is pursuing his M.TECH from this college and he received his M.C.A from Vinayaka Missions university, Chennai, During 2009.

Mrs. K. Subhashini obtained her Msc from Andhra University. She received her M.Tech in Computer Science and Engineering from JNTU Kakinada in 2011. Currently she is working as Assistant Professor in the Department of Computer Science and Engineering, NOVA College of Engineering & Technology, Vegavaram, Jangareddygudem, West Godavari (Dist). She is having 6 years of Teaching and Technical experience. She is area of interest on Networking.