International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Active Watermarking Approach in Detecting Encrypted Traffic Attack by Making Correlation Scheme Robust

Saptshree Dengle [1], Dr. Santosh Lomte [2]

[1, 2]Department of Computer Science, BAMU University, Everest College of Engineering and Technology, Aurangabad

Abstract: ***Network security is complex and challenging problem in today's world. Despite of many sophisticated techniques, attack on the network continues to increase. At present, in order to hide the identity of the attacker, attackers send their attack through a chain of compromised hosts that are used as "stepping stones". In this paper we present an approach to find the connection chain of an intruder for tracing back to the origin especially if the attack through the traffic is encrypted one. Our approach will based on analyzing correlations of encrypted connection between number of packets sent in outgoing connections and that of the incoming packets in the connection. We proposed a correlation scheme based on watermarking which will be robust against timing perturbation. This approach yields effective better results in terms of number of packets than in existing passive timing based correlation. This paper presents a new method of embedding a watermark in traffic flow. Here for the purpose of embedding the watermark, the packet timing is adjusted for specific intervals. By slightly changing the packet timing, we achieve robust correlation of encrypted network against random timing perturbation.***

Keywords: Correlation, IPD, Robustness, Stepping stones, Watermark

1. Introduction

In recent years, unauthorized accesses to the computer systems are increasing as various activities takes place on the internet. The common way for network intruders to conceal their identity by connecting across intermediate hosts before attacking the final target. Intruders do not log in directly to their final targets from their own computers, but they firstly make login through various hosts and then to another hosts and continue this series several times which makes a "chain of intermediate hosts" before breaking into their final targets. Therefore, it becomes necessary for the victim (attack target) to trace back the chain to find the origin of attack. For this it is important to correlate incoming packets and outgoing packets. So, correlation methods are needed to link connections between stepping stones. The earlier work on connection correlation was based on tracking of user activities or connection content (packet payload) was used. Later on the correlation scheme based on the timing characteristics. But the attacker can perturb the timing characteristics by introducing extra delays when forwarding the packets at stepping stones. This will increase correlation false positive rate or decrease correlation true positive rate. The timing based correlation approaches are passive because they do not manipulate the traffic timing characteristics .In this paper, we will develop efficient correlation scheme that is robust against timing perturbation. In this, we will use watermark based approach, where we will embed a unique watermark by slightly changing or adjusting the timing of selected packets making the correlation scheme active. There are various advantages of embedding such use watermark such as it does not make any limiting assumptions about distribution process of original inter packet timing and also this scheme needs less number of packets as compared to the passive timing based correlation approach. Our goal is to develop a practical correlation scheme that is robust against random timing perturbation. These approaches embed a timing based watermark into a network flow by adjusting the timing of selected packets. The trace-back is achieved by embedding/decoding watermarks in the network flows and correlating the flows with similar watermarks. This techniques only use packet timing for trace-back purposes, they can handle the encryption due to secure protocols such as SSH and IPSec.

In this paper, we proposed an active watermarking trace-back systems by analyzing the packetize delays between adjacent stepping stones. We develop an algorithm to infer watermark parameters and detecting the existence of watermarks as early as possible. In this paper we will analyze the watermark parameters when are not chosen carefully and the existence of watermarks in a network flow can always be quickly detected.

2. Literature Survey

Most of the existing correlations connections approaches assume the traffic is unencrypted. The existing connection correlation approach based on different characteristics, listed as follows, First is the host based method (DIDS, CIS) [9], where this method setup the components for tracing at each host, But the major drawback is of this host method is that if the tracing system is not used on a particular hosts or is modified by an intruder, the whole system cannot function reliably once the intruder goes through that host. In the internet environment, it is difficult to require that all administrative domains employ a particular tracing system on all hosts. So, host based method is not trustworthy. Following the early work based on packet payloads (ON/OFF, Deviation based) [9] techniques. In these techniques, the attacker can easily transform the connection content by encryption at the application layer. This approach is basically suitable for unencrypted connections. The ON/OFF based scheme by Zhang and Paxson[3] is the first connection intended to correlate traffic across stepping stones even if the traffic is encrypted by the stepping stone.

The method based on correlation of the ends of OFF periods of interactive traffics. 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. Deviation based method focuses on telnet and rlogin as interactive application ,intruders use to log in through hosts where it involves setting up packet monitoring on internet to record the activities of intruders at packet level. But it is not efficient when some of the connecting part in the chain is encrypted or compression is used in the connection. Donoho et.al [6] represents better understanding of inherent limitations by the attacker on time-based correlation. As opposed to the passive approaches, Wang and Reeves [1], [9] proposed active timing based correlation techniques that robust against timing perturbation.

3. Proposed System

While using a sequence of hosts, an attacker needs to establish a connection between adjacent hosts to make an chain of connections. The purpose of the attacker of making connection between hosts is that the commands can be relayed to the intermediate or remote host and their responses back to the attacker. The active watermarking scheme embeds watermarks in the flows and attempts to detect them in order to trace back the attacker's origin. This watermarking approach will embed a watermark by manipulating or changing the inter-packet delays (IPDs) of selected packets.

3.1 Active Watermarking Concept

Let the IPD between two packets be Pa and Pb is,

$$Ipd_{(a,b)} = tb - ta \quad (1)$$

Here, Pb is transmitted first and then later Pa, where ta and tb are the timestamps of Pa and Pb.

IPDs are quantized for robustness, so for quantization step S the quantization function q(ipd ,S) rounds off ipd / S to the nearest integer. For embedding one watermark bit w (0 or 1), an ipd is slightly increased by delaying second packet the smallest amount so that watermarked IPD ,is denoted as ipd^W, satisfies the condition ,

$$q(ipd^W, S) \bmod 2 = w \quad (2)$$

so that ipd is even multiples of S when 0 is embedded and odd multiples of S when 1 is embedded. Now, the watermark embedding function is,

$$e(ipd, w, S) = [q(ipd + S/2, S) + \partial] \times S, \text{ where } \partial = (w - (q(ipd + S/2, S) \bmod 2) + 2) \bmod 2 \quad (3)$$

This quantizes (ipd +S/2) to ensure that $ipd^W \geq ipd$.

So that the watermark bit can be embedded by delaying the second packet involved in the IPD by the amount of (ipd^W – ipd).So hereafter, a delay caused by watermark embedding process is called as watermark delay.

Now the watermarking decoding function involves,

$$d(ipd, S) = q(ipd, S) \bmod 2 \quad (4)$$

When timing perturbation is introduced after watermarking, as long as the change on ipd^W is limited by (-S/2, S/2], this function can decode the watermark bit correctly. Therefore, a watermark bit embedded in a single IPD can resist up to S/2 random timing perturbation. To resist perturbations larger than S/2,M (M>1) IPDs are used to embed one bit.

The average of M IPDs is computed as,

$$ipd_{avg} = 1/M \sum ipd_i \quad (5)$$

for the upper bound M and for the lower bound i=1.and then watermarked IPD average is calculated as e(ipd_{avg} ,w, S). Here m is the degree of robustness (i.e., the number of IPDs used to embed 1 bit).

The watermarked bit is embedded by increasing each of these M IPDs by the amount of (average of ipd^W – average of ipd). Decoding the watermark bit on the M IPDs simply involves computing d(average of ipd^W ,S).With the same S, embedding 1 bit watermark with multiple IPDs provides higher resistance to the random timing perturbation than single IPD.

Let L-bit watermark W is embedded by repeating the procedure of embedding a single bit L times. Here L is the number of binary bits in the watermark .So, during the watermark detection, another L-bit watermark W' is decoded from suspicious flow and compared with W. If the hamming distance between W and W' is less than or equal to predefined threshold h, this approach reports that a stepping stone flow is detected. It shows that this approach is highly robust against random timing perturbation. In this the watermark parameter w1….wL are the watermark in which each bit wi is either 0 or 1.

3.2 Watermarking Model

Let us consider unidirectional flow in which n>1 packets, let ti and t'i be the incoming and outgoing times of ith packet pi of flow incoming and outgoing from stepping stone. Assuming no loss queuing delay added by stepping stone is constant.So, c > 0.

Let di be the extra delay introduced by the attacker at the intermediate host.

$$\text{So we have ,} t'i = ti + c + di \quad (6)$$

We will introduced an incoming inter-packet delay (IIPD) between pi and pj as,

$$ipdi;j = tj - ti \quad (7)$$

and outgoing inter-packet delay (OIPD) between pi and pj as,

$$ipd'i;j = t'j - t'i \quad (8)$$

We will define the impact on ildi;j by attacker .so the impact will be,

$$ipdi;j - ipd'i;j = dj - di \quad (9)$$

Here we will use the ith and jth packets timestamps to calculate incoming and outgoing inter-packet delay in the packet flow.

Here the negative impact of using invalid packet due to packet reordering will be equivalent to random timing impact over inter-packet delay. Let D be the maximum delay that attacker can add to pi (i = 1,….,n),for D > 0.

Hence the impact will be dj - di belongs to [-D,D].Where [-D,D] is called as impact range of attacker. To make the correlation more robust, we embed watermark using IPDs from randomly and independent selected packets. For the packet sequence p1,….,pn along with the timestamp t1,….,tn respectively (ti < tj for 1 <= i < j <= n),we probabilistically choose 2m < n packets by following process: Firstly we consider each of n packets sequentially and secondly independently determining if current packet will be chosen with the probability p = 2m/n (0 < m < n/2) for watermarking purpose .Here for the purpose of watermarking ,selection of one packets is independent from the selection of another packet.So,2m will be distinct packets selected randomly from n packets.

3.3 Detecting Watermark Existence

Let the secrete information shared between the watermark embedder and decoder be represented as <S,m,l,s,w> where S is the packet selection function that returns (l+1) x m packets>=1 is the number of redundant pairs of packets in which to embed one watermark bit,l>0 is the length of the watermark in bits, s>0 is the quantization step size, and w is l-bit watermark to be detected,. Let f denotes the flow to be examined and wf be the decoded l bits from flow f.

The watermark detector works as follows:

First Decode the l-bit wf from flow f and then compare the decoded wf with w.After both this steps report the watermark w is detected in flow f if the hamming distance between wf and w ,represented as H(wf ,w) is less than or equal to h ,where h is a threshold parameter determined by the user and 0<=h<l. Let 0<p<1 be the probability that each embedded watermark bit will survive the timing perturbation by attacker. Then probabity that all l bits survive the timing perturbation by the attacker will be pl will tend to be small unless p is very close to 1.

By using hamming distance h to detect the watermark wf, the expected watermark detection rate will be,

$$\sum(\ ^{l}_{i}\ )p^{l-i}(1-p)^{i} \quad (10)$$

For i=0 to upper the upper bound h.

For example, for the value p=0.9102, l=24, h=5, the expected watermark detection rate with exact bit match would be pl =10.45%.For the same values of p, l, h, the expected watermark detection rate using a hamming distance h=5 would be 93.29%.

4. Experiment and Analysis

4.1 Watermark bit embedding and decoding

Generally, watermarking involves the selection of a watermark carrier, and the design of two complementary processes: embedding and decoding. In the registration, we collect the watermark signature... The watermark embedding process inserts the information by a slight modification of some property of the carrier. The watermark decoding process detects and extracts the watermark (equivalently, determines the existence of a given watermark). To correlate encrypted connections, we propose to use the inter-packet timing as the watermark carrier property of interest. The embedded watermark bit is guaranteed to be not corrupted by the timing perturbation. If the perturbation is outside this range, the embedded watermark bit may be altered by the attacker. The watermark embedding and decoding process will be as follow:

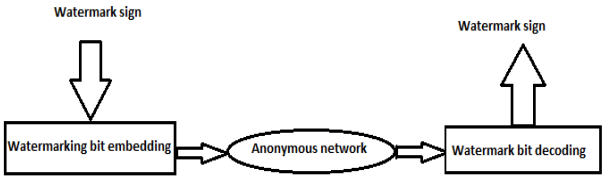

Figure 1: Watermark bit embedding and decoding

As an IPD is conceptually a continuous value, we will first quantize the IPD before embedding the watermark bit. Given any IPD *ipd* > 0, we define the *quantization of ipd* with uniform quantization step size *s* > 0 as the function

$$q(ipd, s) = round(ipd/s) \quad (11)$$

where round(*x*) is the function that rounds off real number *x* to its nearest integer *ipd.*

It is easy to see that $q(k \times s; s) = q(k \times s + y; s)$ for any integer *k* and any $y \in [-s/2; s/2]$. Let *ipd* denote the original IPD before watermark bit *w* is embedded, and *ipdw* denote the IPD after watermark bit *w* is embedded. To embed a binary digit or bit *w* into an IPD, we slightly adjust that IPD such that the quantization of the adjusted IPD will have *w* as the remainder when the modulus 2 is taken. Given any *ipd* > 0, *s* > 0 and binary digit *w*, the watermark bit embedding is defined as function.

$$e(ipd, w, s) = [q(ipd + s/2, s) + \Delta] \times s \quad (12)$$

Where, $\Delta = (w - (q(ipd + s/2, s) \bmod 2) + 2) \bmod 2$.

The embedding of one watermark bit *w* into scalar *ipd* is done through increasing the quantization of *ipd*+s/2 by the normalized difference between *w* and modulo 2 of the

quantization of *ipd*+*s*/2, so that the quantization of resulting ipd^w will have *w* as the remainder when modulus 2 is taken. The reason to quantize *ipd*+*s*/2 rather than *ipd* here is to make sure that the resulting *e*(*ipd*,*w*,*s*) is no less than *ipd*, i.e., packets can be delayed, but cannot be output earlier than they arrive. The watermark bit decoding function is defined as

$$d(ipd^w,s)= q(ipd^w, s) \bmod 2 \quad (13)$$

Algorithm for watermark bit detection will be as follow:

Watermark detection refers to the process of determining if a given watermark is embedded into the IPDs of a specific connection or flow. Let the secret information shared between the watermark embedder and decoder be represented as <S,m, l,s,w>, where S is the packet selection function that returns (l+1)* m packets, m ≥1 is the number of redundant pairs of packets in which to embed one watermark bit, l > 0 is the length of the watermark in bits, s > 0 is the quantization step size, and w is the l-bit watermark to be detected. Let f denote the flow to be examined, and wf denote the decoded l bits from flow.The watermark detector works as follows:

1. Decode the l-bit wf from flow f.
2. Compare the decoded wf with w.
3. Report that watermark w is detected in flow f if the Hamming distance between wf and w, represented as H (wf, w) is less than or equal to h, where h is a threshold parameter determined by the user, and 0 <= h < l.

4.2 Mathematical Model

Once user submit request, it contains watermark image embedded with the file and that file is send in the format of packets in the form of intervals .And admin detects the authenticate user by using detection algorithm.

Intersect and Merge Approach

The Figure shows mathematical model for mapping input real user requests with encrypted file embedded with watermark image.

Assume the set (U1, U2, U3 ,….,Un) is the set of requested users and (w1, w2,…..w3,….,wn) are the watermark image for the respective users .

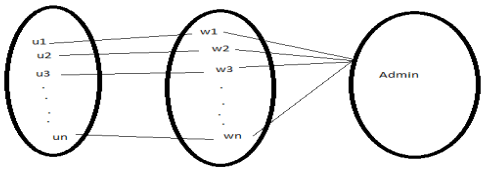

Figure 2: Mathematical Model

4.3 Experimental Components

Detection Algorithm

Detection: The probability of detecting real edge points should be maximized while the probability of falsely detecting non-edge points should be minimized. This corresponds to maximizing the signal-to-noise ratio.

Let ±*i* be the delay added to packet *Pi*, and *t0 i* be the distorted time stamp of packet *Pi*, then *t0 i* = *ti* + ±*i*. The original and distorted inter-packet delays (IPD) between *Pi*+1 and *Pi* are *Ii* =*ti*+1 ¡ *ti* and *I0 i* = *t0 i*+1 ¡ *t0 i* respectively. Therefore,±*k* = *t0k* ¡ *tk* = ±1 +*k*X¡1*i* = 1(*I0i* ¡ *Ii*) The original and the perturbed inter-packet timing characteristics of packet flow *P*1*; : : : ; Pn* can be represented by < *t*1*; I*1*; : : : ; In¡*1 > and < *t*01*; I*01 *; : : : ; I0n¡*1 > respectively. In, particular, < *I*01*; I0n¡*1 > represents the distortion pattern over the original inter-packet timing characteristics.

According to results from section VI-A, in order to completely remove any hidden information from the original interpacket timing characteristics, the adversary needs to disturb < *t*1*; I*1*; In¡*1 > into an independent one. That means < *I*01 *; : : : ; I0n¡*1 > needs to be independent from < *I*1*; : : : ; In¡*1 >.

Therefore, the distortion pattern < *I*01*; : : : ; I0n¡*1 > can be thought to be pre-determined before the original inter-packet timing characteristics .

1) Watermarking Engine:

Watermark generator: The watermark generator generates unique watermarks with a specified hamming distance. The distance is important to ensure low false positives. These watermarks are embedded into traffic flows by the watermarking engine. The delay introduced by the watermark has to be small in order to make it difficult for the attacker to determine if his flow is watermarked. All the watermarks are 24 bits in size. The watermarks are generated such that the minimum hamming distance between any two watermarks is 5. A watermarking delay of 3ms was sufficient to confuse the attacker and achieve high true positive rate.

2) Watermark decoder

The watermark decoder acts as the egress monitor which checks outgoing traffic flows for watermarks. It is assumed that there is coordination between the watermarking engine and the decoder knows which packets are watermarked. The watermark that is found is decoder compared with all the embedded watermarks and analyzed to determine false positives. For a detection scheme to be effective, it should not only have a high detection rate but also a low *false positive* rate. A *false positive* can occur when the watermark decoder erroneously finds a watermark in an un-watermarked flow or in a flow that has a different watermark.

In this project, we address the random timing perturbation problem in correlating encrypted connections through stepping 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.

Previous timing based correlation method that considers both correlation true positive and false positive at the same time. Based on the assumption that the inter-packet timing of flows can be modeled as a sequence of Poisson processes of different rates, they derived upper bounds on the number of packets needed to achieve a specified false positive rate and a true positive rate. However, their work did not present any experimental results, nor did it address such practical issues as how to derive model parameters in real-time or how many packets are needed in practice for real flows and realistic timing perturbations. Passive approaches are simple to implement and undetectable by the attacker. However, they generally make more limiting assumptions about the inter-packet timing characteristics, and require more packets than an active approach to effectively correlate timing perturbed flows, as we will show. We have also compared the effectiveness and the numbers of packets needed by our active watermark correlation approach, and a representative passive correlation approach, under identical levels of timing perturbation on the same sets of traces. For Poisson arrivals, aimed that its detection algorithm DETECT-ATTACKS guaranteed to achieve a detection rate and 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 we implemented detection algorithm, and after identifying the maximum number of packets in any time interval from flow we applied detection algorithm, with to correlate flows and the corresponding perturbed flows with maximum uniformly distributed perturbation. Surprisingly, the detection algorithm DETECT-ATTACKS in this test only achieved a 79.5% detection rate, while experiencing no false positives. Our watermark-based correlation method achieved at least a 91.9% true positive rate and about a 0.3% false positive rate, using parameter values of h=5, l=24, s=400ms and m=12 on flow. Our active watermark-based correlation makes no assumptions about the original distribution of the inter-packet timing of the original packet flow, and it does not require the adversary's timing perturbation to follow any specific distribution or random process to be effective. Our active watermark-based correlation was shown to require substantially fewer packets than a representative passive timing-based correlation method to achieve a given level of robustness.

Following shows the table with its correct watermarking.

Table 1: Correct Watermarking signature

No of Packets	*Correct Watermarking*	*Incorrect Watermarking*
1	604.395	298
2	288	738
3	322	743
4	327	749
5	358	78
6	873	664
7	364	787
8	369	794
9	377	830

Following shows the figure with its correct watermarking and incorrect watermarking plotting.

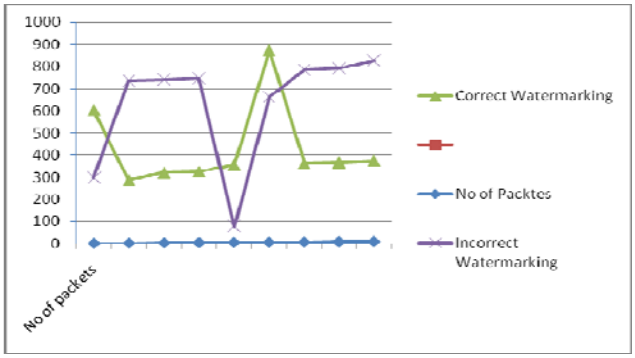

Figure 3: no of packets with its IPT in ms

5. Conclusion

The random timing perturbation greatly reduces the effectiveness of passive timing approaches. In this paper, we analyze the active watermarking scheme for tracing through stepping stones. Our active watermark-based correlation requires fewer packets than a passive timing based correlation method to achieve a given level of robustness. Here we identified the provable upper bounds on the number of packets needed to achieve desire correlation effectiveness under given level of perturbation. One interesting area of future work is to investigate how to make the flow watermarking more robust with fewer packets.

Acknowledgements

The author would like to thank Xinyuan Wang and Douglas S. Reeves, Pai Peng, Peng Ning and its members for contributing in making the correlation schemes robust.

References

[1] X.Wang, D. Reeves, S. F. Wu and J. Yuill .Sleepy Watermark Tracing: An Active Network -Based Intrusion response framework. In *Proceedings of the 16th International Conference on information Security (IFIP/Sec 2001),* pages 369 -384. Kluwer Academic Publishers, June 2001.

[2] K.Yoda and H. Etoh. Finding a Connection Chain for Tracing Intruders. In *Proceeding of the 6th European Symposium on Research in Computer Security (ESORICS 2000), LNCS-1895,* pages 191-205. Springer-Verlag, October 2002

[3] Y.Zhang and V. Paxson. Detecting Stepping Stones.In Proceedings of the 9th USENIX Security Symposium,pages 171 - 184.USENIX ,2000.

[4] L. Zhang, A. G. Persaud, A. Johnson, and Y. Guan. Detection of Stepping Stone Attack under Delay and Chaff Perturbations. In Proceedings of the 25th IEEE International Performance Computing and Communications Conference (IPCCC 2006), April 2006

[5] I. Cox, M. Miller, and J. Bloom. *Digital Watermarking.* Morgan-Kaufmann Publishers, 2002.

[6] D. Donoho .et al. Multiscale stepping Stone Detection: Detecting Pairs of Jittered Interactive Streams By Exploiting Maximum Tolerable Delay. In *proceedings of the 5th International symposium on Recent Advances in Intrusion Detection (RAID 2002): LNCS-2516,*pages17-3.Springer,October 2002.

[7] T. He and L. Tong, Detecting Encrypted Stepping-Stone Connections. In *IEEE Transactions on Signal Processing, 55(5), pages 1612-1623, 2006.*

[8] P. Peng, P. Ning, D. S. reeves, On Secrecy of Timing-Based Active Watermarking Trace-Back Techniques. In *Proceedings of the 2006 IEEE Symposium on Security* & Privacy (S&P 2006), May 2006.

[9] X. Wang, D. Reeves Robust Correlation of Encrypted Attack Traffic Through Stepping Stones by Watermarking The Interpacket Timing, In Proceedings of the *10th ACM Conference on Computer and Communications Security (CCS 2003).*