Volume : 2 | Issue : 4 | April 2013 • ISSN No 2277 - 8160



Research Paper | Information Technology

Interpacket Delay Analysis Scheme for Encrypted Attack Traffic

Anandakumar.H Assistant professor, Department of Information Technology, SNS College of Engineering, Coimbatore-641 107, Tamilnadu, India

Lekhaa.T. R Assistant professor, Department of Information Technology, SNS College of Engineering, Coimbatore-641 107, Tamilnadu, India

ABSTRACT

Digital watermarking techniques are used for authentication. Embedding and decoding tasks are performed in the watermarking. The watermark embedding process embeds the watermark bits into the carrier signal. Watermark decoder process detects and extracts any watermark bits from the carrier signal. Attack packets are passed through a set of intermediate nodes. Chain of compromised hosts acting as "stepping stones". Attack detection system detects stepping-stone connections at the compromised hosts. Encryption and perturbation methods are used to increase the complexity of the detection systems. The survey is carried out under Stepping stone analysis, Encrypted packet analysis, Time perturbation model and Inter Packet Delay analysis model. The stepping stone analysis is performed to detect attacks. The encrypted packet analysis is carried out to detect attacks with encrypted packets. Time perturbation analysis is done to detect packet attacks with time changes. Inter packet delay based analysis is performed to detect attacks with packet delay values.

KEYWORDS:Digital Watermarking, Decoding, Encrypted Packet Analysis, Packet Delay

I. INTRODUCTION

Network based attacks have become a serious threat to the critical information infrastructure on which we depend. To stop or repel network-based attacks, it is critical to be able to identify the source of the attack. Attackers, however, go to some lengths to conceal their identities and origin, using a variety of countermeasures.

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 attack 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. Later work relied on comparing the packet contents, or payloads, of the connections to be correlated. Most recent work has focused on the timing characteristics of connections, in order to correlate encrypted connections. 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 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 stepping stones.

I 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 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 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. RELATED WORK

Existing connection correlation approaches are based on three different characteristics: 1) host activity; 2) connection content; and 3) inter-packet timing

characteristics. The host activity based approach collects and tracks users' login activity at each stepping stone. The major drawback of host activity based methods is that the host activity collected from each stepping stone is generally not trustworthy. Since the attacker is assumed to have full control over each stepping stone, he/she can easily modify, delete or forge user login information. This defeat the ability to correlate based on host activity. Content based correlation approaches require that the payload of packets remains invariant across stepping stones. Since the attacker can easily transform the connection content by encryption at the application layer, these approaches are suitable only for unencrypted connections.

To correlate encrypted traffic, timing based approaches have been proposed. These methods passively monitor the arrival and/or departure times of packets, and use this information to correlate incoming and outgoing flows of a stepping stone. For instance, IPDbased correlation has shown that 1) the important inter-packet timing characteristics of connections are preserved during transit across many routers and stepping stones; and 2) the timing characteristics of interactive flows are almost always unique enough to differentiate related flows from unrelated flows.

While the first generation of timing based correlation approaches have proved to be effective in correlating encrypted connections, they are vulnerable to the attacker's use of active timing perturbation. Donoho et al. first investigated the theoretical limits on the attacker's ability to disguise his traffic through timing perturbation and bogus packet injection. By using multiple-timescale analysis techniques, they show that correlation based on long term behavior is still possible despite certain timing perturbations by the attackers. However, they do not present any tradeoffs between the magnitude of the timing perturbation, the desired correlation effectiveness, and the number of packets needed. Another important issue not addressed is the correlation false positive rate. While coarse timescale analysis for long term behavior may not be affected by packet jitter introduced by the attackers, it may also be insensitive to the unique details of each flow's timing. Therefore coarse scale analysis tends to increase the correlation false positive rate, while increasing the correlation true positive rate of timing-perturbed flows. Nevertheless, Donoho et al's work represents an important first step towards understanding the inherent limitations of timing perturbations by the adversary on Timing-based correlation. Not addressed in this work were questions about correlation effectiveness for flows of arbitrary distribution in packet timing, and the achievable tradeoff of false and true positive rates given the magnitude of the timing perturbation and the number of packets available.

After the initial publication of our active watermarking based correlation, Blum et al. developed another passive, 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 100% true positive rate. They also derived the lower bounds on the amount of chaff needed to defeat their passive method of correlation. 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. Zhang et al. [7] and He et al. [10] recently proposed several new passive timing based correlation methods based on similar assumptions used by Blum et al. and they showed that their methods have better performance than what proposed by Blum et al.

Chakinala et al. [2] formally analyzed the packet reordering channels as information theoretic game. Peng et al. [9] recently studied the secrecy of the watermark based correlation and proposed an offline statistical method for detecting the existence of watermark from a packet flow.

However, their method assumes the watermark embedding follows some simple and fixed patterns and requires access to both the un watermarked and watermarked flows to be effective. As we will show in the watermark tracking model, we can make un watermarked flow unavailable to the adversary by watermarking the packet flow from its source. In summary, existing timing-based correlation approaches passively information for correlation. They do not attempt to make the inter-packet timing characteristics more amenable to effective correlation. 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.

In this paper, we describe a novel active timing-based correlation approach that 1) makes no assumption about the distribution of inter packet timing intervals; 2) does not require the timing perturbation to follow any specific distribution or random process; 3) is provably effective against certain correlated random timing perturbation; and 4) requires substantially fewer packets than passive approaches to achieve the same level of correlation effectiveness.

II. Overall Watermark Tracing Model

The watermark tracing approach exploits the observation that interactive connections are bidirectional. The idea is to watermark the backward traffic of the Bidirectional attack connections by slightly adjusting the timing of selected packets. If the embedded watermark is both robust and unique, the watermarked back traffic can be effectively correlated and traced across stepping stones, from the victim all the way back to the attacker. As shown in Figure 1, the attacker may connect through a number of hosts (H1,…..,Hn) before attacking the final target. Assuming the attacker has not gained full control on the attack target, the attack target will initiate the attack tracing after it has detected the attack. Specifically, the attack target will watermark the backward traffic of the attack connection, and inform sensors across the network about the watermark. The sensors across the network will scan all traffic for the presence of the indicated watermark, and report to the target if any occurrences of the watermark are detected.

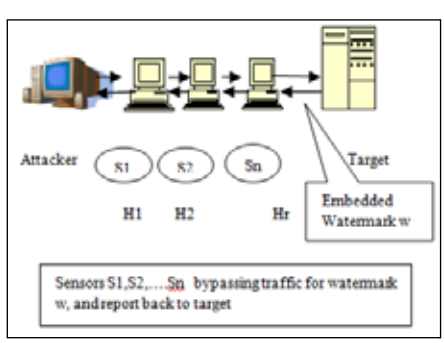


Fig. 1. Overall Watermark Tracing Model

Gateway, firewall and edge router are good places to deploy sensors. However, how many sensors can be deployed depend on not only the resources available but also the administrative privilege. How to optimally deploy limited number of sensors over particular network is an open research problem [6]. Due to space limitation, we leave aside the sensor deployment issues, and instead focus on the watermark tracing approach itself.

Since the backward traffic is watermarked at its very source – the attack target, which is not controlled by the attacker, the attacker will not have access to an un watermarked version of the traffic. This makes it difficult for the attacker to determine which packets have been delayed by the watermarking process, running at the target. The objective of watermark-based correlation is to make the correlation of encrypted connections probabilistically robust against random timing perturbations by the adversary. Unlike existing timing-based correlation schemes, our watermark-based correlation is active in that it embeds a unique watermark into the encrypted flows, by slightly adjusting the timing of selected packets. If the embedded watermark is both unique and robust, the watermarked flows can be effectively identified and thus correlated at each stepping stone.

In contrast to most previous passive correlation approaches, our watermark-based correlation makes no limiting assumption about the distribution or random process of the original inter-packet timing characteristics of the flows to be correlated. We assume the following about the random timing perturbations introduced by the adversary:

1) While the attacker can add extra delay to any or all packets of an outgoing flow at the stepping stone, the maximum delay he or she can introduce is bounded.
2) All packets in the original flow are kept. No packets are dropped from or added to the flow by the stepping stone.
3) While the watermarking scheme is public knowledge, the watermarking embedding and decoding parameters are secrets known only to the watermark embedder and the watermark detector(s).

Here we do not require that the packet order of two flows be the same, as long as the total number of packets is not modified. As shown in works [8], [3], our watermark-based approach is able to correlate encrypted flows even if chaff and timing perturbation are applied at the same time. Due to space limitation, we only consider timing perturbation in this paper.

In contrast to all previous passive approaches, our correlation method does not require the random timing perturbation introduced by the attacker to follow any Particular distribution or random process to be effective. The only assumption about timing perturbations is that they follow some distribution of finite variance and they have the same covariance among each other.

III. BASIC AND PROBABILISTIC WATERMARKING

Basic Watermark Bit Embedding and encoding 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) - - (1)

where round(x) is the function that rounds off real number x to its nearest integer. The quantization for scalar x. It is easy to see that q(k s, s) = q(k s + y, s) for any integer k and any y [-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) + ¢] £ s -- (2)
where ¢ = (w ¡ (q(ipd + s=2; s) mod 2) + 2) mod2.

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 ipdw 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 embedding of watermark bit w by mapping ranges of unwatermarked ipd to the corresponding watermark ipdw. The watermark bit decoding function is defined as d(ipdw; s) = q(ipdw; s) mod 2.

B. Maximum Tolerable Perturbation

If the perturbation of an IPD is within the tolerable perturbation range [-s/2, s/2], the embedded watermark bit is guaranteed to be not corrupted by the timing perturbation. If the perturbation of the IPD is outside this range, the embedded watermark bit may be altered by the attacker. Therefore, the larger the value of s, the more robust the embedded watermark bit will be. However, a larger value of s may disturb the timing the watermarked flow more, as the watermark bit embedding itself may add up to 2s delay to selected packets.

It is desirable to have a watermark embedding scheme that:

1) disturbs the timing of watermarked flows as little as possible, so that the watermark embedding is less noticeable; and
2) ensures the embedded watermark bit is robust, with high probability, against timing perturbations that are outside the tolerable perturbation range [-s/2, s/2].

Embedding a single watermark bit over the average of multiple ipds

To make the embedded watermark bit probabilistically robust against larger random delays than s=2, the key is to contain and minimize the impact of the random delays on the watermark bearing IPDs so that the impact of the random delays will fall, with high probability, within the tolerable perturbation range [-s/2, s/2]. We exploit the assumption that the attacker does not know which packets are randomly selected and which IPDs will be used for embedding the watermark.

I apply two strategies to contain and minimize the impact of random delays over the watermark-bearing IPDs. The first strategy is to distribute watermark-bearing IPDs over a longer duration of the flow. The second is to embed a watermark bit in the average of multiple IPDs. The rationale behind these strategies is as follows. While the attacker may add a large delay to a single IPD, it is impossible to add large delays to all IPDs. In fact, random delays tend to increase some IPDs and decrease others. Therefore the impact on the average of multiple IPDs is more likely to be within the tolerable perturbation range [-s/2, s/2], even when the perturbation range [-D,D] is much larger than [- s/2, s/2].

IV. PROBABILISTIC WATERMARKING AND TIMING PERTURBATION

I now consider the probabilistic watermark decoding in the presence of active timing perturbation. Base on very moderate assumptions about the random timing perturbations, we first establish an upper bound of the watermark bit decoding error probability, and then derive an approximation to the watermark bit decoding error probability.

Upper bound of watermarking bit decodding error probability

The probability that the overall impact of random delays on ipdavg is outside the tolerable perturbation range (-s/2, s/2] is bounded. In addition, that probability can be reduced to be arbitrarily close 0 by increasing m, the number of redundant IPDs averaged together before embedding the watermark bit. Since the watermark bit decoding error probability is less than Pr(|Xm| ¸ the derived upper bound is conservative and it holds true regardless of the distribution, mean or the variance of the random delays added by the attacker, or of the maximum quantization allowed for watermarking embedding. Furthermore, the upper bound of the error probability holds true even if the random delays on different packets are correlated.

Approximation to the watermark bit robustness

We assume the random delays added by the adversary are independent and identically distributed (iid), and we derive an accurate approximation to the watermark bit robustness Pr(|Xm| < via the well-known Central Limit Theorem of statistics. Although

the approximation model assumes the random delays are iid, our experiments demonstrate that the derived approximation model can accurately model non-iid random delays.

C. Analysis of watermark detectability

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 f. 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.

IV ENHANCEMENT OF FLOW WATERMARKING TECHNIQUE

The digital watermarking technique is used to provide authentications. The watermarking models are used in the packets with user authentications. This type of watermarking is referred as flow watermarking model. The packets are updated with source information. The packet information is secured with encrypted data values. The stepping stone attack is initiated with user identity and origin

hidden model. The stepping stone attack detection is a complex task with encrypted packets environment.

Time perturbation models are used to hide actual packet time values. The perturbed time values can be verified in the destination nodes. The intermediate node data changes and path directions can not affect the packet contents. The watermark is embedded with the packets. The packet correlation analysis is carried out to detect stepping stone attacks. The system is designed to enhance the active watermarking based attack detection model to control test packet count. The packet correlation is used in the detection process. Incoming and outgoing packet correlation is optimized with threshold values. Inter packet delay analysis is integrated with the system. The system is updated to analyze the packet dropping process. The system development is planned with Java language and Oracle back end.

The system is designed to detect stepping stone attacks. Flow watermarking and encrypted packets are used in the system. Packet delay is verified for attack analysis. The system is divided into five major modules. They are packet analysis, user authentication, watermarking process, delay analysis and attack verifier. Packet analysis module is designed to analyze packet information. User authentication is designed to verify the user. Watermarking process module is designed to verify watermarking in packets. Delay analysis module is used to validate packet delay details. Attack verifier module is designed to detect attack and its origin.

A. Packet analysis

The packet header information are extracted for analysis. Packet contents are decrypted in the analysis process. Watermark, source and time information are extracted from the packets. Address verification is also carried out in the packet analysis.

B. User authentication

Source information is verified in the user authentication process. User information are maintained in encrypted form. Watermarks are used to represent user identity. Time information is also used in the user authentication process.

C. Watermarking process

Flow watermarking is used in the authentication process. Watermarks are embedded by the source node. The receiver node verifies the watermarking images. Watermarks are updated in the packets.

D. delay analysis

Time information is used in the delay analysis. Time information are perturbed in the header. Transmission delay is verified in the system. Packet modification is identified in the delay analysis.

E. attack verifier

The attack detection is performed in the attack verifier module. Packet delay, watermark and time details are used in the attack verification process. The system detects the attacker origin. The system improves the detection rate.

IV. CONCLUSION

Stepping stones are used to hide identity and origin of the attacker. Flow watermarking technique is used to detect attacks with encrypted packets and time perturbed data. Packet flow correlation is extracted for the detection process. The system is enhanced to perform detection with minimum test packet count. The system manages detects stepping stone attacks. Packet delay based model. The system performs encrypted packet analysis. The system supports flow watermarking.

REFERENCES

[1] A. Blum, D. Song, and S. Venkataraman, "Detection of Interactive Stepping Stones: Algorithms and Confidence Bounds," Proc. Seventh Int'l Symp. Recent Advances in Intrusion Detection (RAID '04), Oct. 2004. [2] R.C. Chakinala, A. Kumarasubramanian and R. Sundaram, "Steganographic Communication in Ordered Channels," Proc. Eighth Information Hiding Int'l Conf. (IH '06), 2006. [3] X. Wang, S. Chen, and S. Jajodia, "Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems," Proc. IEEE Symp. Security and Privacy, 2007. [4] J. Li, M. Sung, J. Xu, and L. Li, "Large Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation," Proc. IEEE Symp. Security and Privacy, 2004. [5] Xinyuan Wang, Douglas S. Reeves, "Robust Correlation of Encrypted Attack Traffic through Stepping Stones by Flow Watermarking," IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 3, pp. 434-449, May/June 2011. [6] Y.J. Pyun and D.S. Reeves, "Deployment of Network Monitors for Attack Attribution," Proc. Fourth Int'l Conf. Broadband Comm., Networks, and Systems (Broadnets '07), pp. 525-534, 2007. [7] L. Zhang and Y. Guan, "Detection of Stepping Stone Attack under Delay and Chaff Perturbations," Proc. 25th IEEE Int'l Performance Computing and Comm. Conf. (IPCCC '06), Apr. 2006. [8] Y.J. Pyun, Y.H. Park, D.S. Reeves, and P. Ning, "Tracing Traffic through Intermediate Hosts that Repacketize Flows," Proc. IEEE INFOCOM '07, May 2007. [9] P. Peng, P. Ning, and D.S. Reeves, "On the Secrecy of Timing-Based Active Watermarking Trace-Back Techniques," Proc. IEEE Symp. Security and Privacy (SP '06), May 2006. [10] T. He and L. Tong, "Detecting Encrypted Stepping-Stone Connections" IEEE Trans. Signal Processing, vol. 55, no. 5, pp. 1612-1623, May 2006.