**.NET Project On   Robust Correlation Of Encrypted Attack Traffic through Stepping Stones By Flow Watermarking**

**Abstract:**

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.

**Existing System:**

 Existing connection correlation approaches are based on three Different characteristics: 1) host activity; 2) connection content (i.e. packet payload); and 3) inter-packet timing characteristics. The host activity based approach collects and tracks users’ login activity at each stepping stone.

**Disadvantages of Existing System:**

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.

**Proposed System:**

 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.

**Advantages of proposed System:**

1) While the attacker can add the secret key in watermarking, we can easily analysis and identify the intruder.

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).

**Algorithm:**

**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 +kX¡1i = 1(I0i ¡ Ii) The original and the perturbed inter-packet timing characteristics of packet flow P1; : : : ; Pn can be represented by < t1; I1; : : : ; In¡1 > and < t01; I01 ; : : : ; I0n¡1 > respectively. In, particular, < I01; 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 < t1; I1; In¡1 > into an independent one. That means < I01 ; : : : ; I0n¡1 > needs to be independent from < I1; : : : ; In¡1 >.

Therefore, the distortion pattern < I01; : : : ; I0n¡1 > can be thought to be pre-determined before the original inter-packet timing characteristics .

**Modules:**

**1.     Watermark Bit Embedding and Decoding**

**2.     Correlation Analysis**

**3.     Watermark Tracing Model**

**4.     Parameter & Mapping Randomization**

**Module Description:**

**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.

**2) Correlation Analysis:**

   In practice, the number of packets available is the fundamental. Limiting factor to the achievable effectiveness of our watermark based correlation. 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. 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.

   We can correlate the watermark signatures and identify it’s the positive or negative correlation, if positive occurs it detect it is the authenticated user otherwise, if negative occurs it detect it is an Intruder.

**3) Watermark Tracing Model:**

    The watermark tracing approach exploits the observation that interactive connections are bidirectional. The idea is to watermark the backward traffic (from victim back to the attacker) 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, 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 across the network about the watermark. The stepping stone 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.

**4) Parameter & Mapping Randomization:**

One simple technique to achieve this is to use a secret key to generate a pseudo-random sequence of numerical values and add them to either or both of and for the pixels in the watermarking area. This technique is hereinafter referred to as parameter randomization.

This parameter exchange does not affect the effectiveness of lossless recoverability, because we can now recover the original pixel values by the compound mappings. We will refer to this technique in the sequel as mapping randomization. We may also combine this technique with the parameter randomization technique to enhance the security.Finally, the Authenticated user take the file in zip format with proper password.

**System Requirements:**

**Hardware Requirements:**

•         System                 : Pentium IV 2.4 GHz.

•         Hard Disk            : 40 GB.

•         Floppy Drive       : 1.44 Mb.

•         Monitor                : 15 VGA Colour.

•         Mouse                  : Logitech.

•         Ram                     : 512 Mb.

**Software Requirements:**

•         Operating system           : - Windows XP.

•         Coding Language :  C#.net

•         Data Base             :  SQL Server 2005