# *iLOC*: An *i*nvisible *LOC*alization Attack to Internet Threat Monitoring Systems

Xun Wang, Wei Yu, Xinwen Fu, Dong Xuan and Wei Zhao

***Abstract*—Internet threat monitoring (ITM) systems have been deployed to detect widespread threats and attacks on the Internet in recent years. However, the integrity and functionality of these systems largely depend on the location anonymity of their monitors. If the locations of monitors are disclosed, the attacker can bypass the monitors or even abuse them, significantly jeopardizing the performance of ITM systems. In this paper, we study a new class of attacks, the *i*nvisible *LOC*alization (*iLOC*) attack. The *iLOC* attack can accurately and invisibly localize monitors of ITM systems. In the *iLOC* attack, the attacker launches low-rate port-scan traffic, encoded with a selected *pseudo-noise code* (PN-code), to targeted networks. While the secret PN-code is invisible to others, the attacker can accurately determine the existence of monitors in the targeted networks based on whether the PN-code is embedded in the report data queried from the data center of the ITM system. We implement the *iLOC* attack and conduct experiments on a real-world ITM system to validate the feasibility of such attacks. We also conduct extensive simulations on the *iLOC* attack using real-world traces. Our data demonstrate that the *iLOC* attack can accurately identify monitors while remaining invisible to the ITM. Finally, we present a set of guidelines to counteract the *iLOC* attack.**

***Index Terms*—Internet threat monitoring systems, Invisible localization attack, PN-code, Security**

## I. Introduction

In recent years, widespread attacks, such as active worms [1] and distributed denial-of-service (DDoS) attacks [2], have been major threats to the Internet. Due to the widespread nature of these attacks, large scale traffic monitoring across the Internet has become necessary in order to effectively detect and defend against them. Developing and deploying *Internet threat monitoring* (ITM) systems (or *motion sensor networks*) is one of the major efforts in this realm.

Generally, an ITM system consists of a number of monitors and a data center. The monitors are distributed across the Internet and can be deployed at hosts, routers, firewalls, etc. Each monitor is responsible for monitoring and collecting traffic targeting a range of IP addresses within a sub-network. The range of IP addresses covered by a monitor is also referred to as the *location* of the monitor. Periodically, the monitors send traffic logs to the data center. The data center aggregates and analyzes these logs and also publishes reports to the public[1]. The reports provide critical insights into widespread Internet threats and attacks, and are used to detect and defend against such attacks. ITM systems have been successfully used to detect the worm outbreaks [3] and DDoS attacks [4]. These systems have been successfully developed and deployed in practice. Examples include Internet Motion Sensor [5], SANS ISC (Internet Storm Center) [3], Internet sink [6], network telescope [7], and CAIDA [8].

However, the integrity and functionality of ITM systems largely depend on the anonymity of the IP addresses covered by their monitors, i.e., the *locations* of monitors. If the locations of monitors are identified, the attacker can deliberately avoid these monitors and directly attack the uncovered IP address space. It is a known fact that the IP address space covered by monitors represents a very small portion of the whole IP address space [3], [6], [7]. Hence, bypassing even a part of IP address spaces covered by monitors can *significantly* degrade the accuracy of the traffic data collected by the ITM system in reflecting the real situation of attack traffic. Furthermore, the attacker may also poison ITM systems by manipulating the traffic towards and captured by disclosed monitors. The attackers may even launch retaliation attacks (e.g., DDoS attacks) against participants (i.e., monitor contributors) of ITM systems, thereby discouraging them from contributing to these systems. In summary, the attacker can significantly compromise the ITM system performance if he is able to disclose the locations of monitors. It is important to have a thorough understanding of such attacks, in order to design efficient countermeasures against them, thereby protecting ITM systems.

In this paper, we conduct a systematic investigation of a class of attacks that aim to *accurately* and *invisibly* localize monitors. To the attacker, accuracy is very important in identifying monitor locations. Meanwhile, invisibility is vital to the attacker. If the attack attempts are identified by defenders (such as the ITM administrators), countermeasures can be applied by defenders to reduce or eliminate attack effects by filtering suspicious traffic [9], decoying attackers [10], and even tracing back to attack origin and hold the attacker accountable for his malicious acts [11]. Invisibility is critical for the attacker to evade these countermeasures.

A few works have been conducted on monitor localization attacks [12], [13]. However, our work is the first to address an attack aiming to achieve the objectives of both accuracy and invisibility. It is challenging for the attacker to achieve these two objectives simultaneously. Intuitively, the attacker can launch high-rate attack traffic as described in [12], [13] to easily achieve high attack accuracy as follows. He launches high-rate port-scan traffic to a target network and then queries the data center for the report on recent port-scan activities. If there is a traffic *spike* in the report data reflecting the high-rate port-scan traffic sent by the attacker, the attacker can determine

Xun Wang and Dong Xuan are with the Department of Computer Science and Engineering, The Ohio-State University, Columbus, OH 43210. E-mail: {wangxu, xuan}@cse.ohio-state.edu. Wei Yu is with the Department of Computer Science, Texas A&M University, College Station, TX 77843. E-mail: weiyu@cs.tamu.edu. Xinwen Fu is with the College of Business and Information Systems, Dakota State University, Madison, SD 57042. E-mail: xinwen.fu@dsu.edu. Wei Zhao is with the School of Science, Rensselaer Polytechnic Institute, Troy, NY 12180. E-mail: zhaow3@rpi.edu. The authors would like to acknowledge Adam Champion and Ms. Larisa Archer for their dedicated help to improve the paper.

[1] In order to maximize the usage of such reports, most existing ITM systems publish the reports online and make them accessible to the public.

that the target network is deployed with monitor(s) which sends traffic logs to the data center. However, it is hard for this scheme to achieve invisibility, since large spikes caused by the attack traffic make the attack easily detectable.

In this paper, we investigate a new class of attacks, the *i*nvisible *LOC*calization (*iLOC*) attack. In the *iLOC* attack, the attacker launches low-rate port-scan traffic (also referred to as *attack traffic*) to target networks. The scan traffic is encoded with a carefully selected *pseudo-noise code* (PN-code), which is only known to the attacker. The PN-code embedded in traffic can be accurately recognized by the attacker even under interference from background traffic. Thus, the attacker is able to *accurately* determine the existence of monitors in the target networks based on whether the embedded PN-code is contained in the report data queried from the data center of the ITM system. The attack traffic modulated/embedded by the PN-code will appear as innocent noise in both the time and frequency domains, rendering it *invisible* to others. Only those aware of the original PN-code can correctly recover the encoded PN-code and identify the monitor locations. Therefore, using the *iLOC* technique, the attacker can accurately localize monitors while evading detection by others.

We conduct both theoretical analysis and experimental evaluation of the *iLOC* attack. We derive formulas for both accuracy and invisibility of the attack. We analyze and discuss the impacts of various attack parameters (e.g., PN-code length, attack traffic rate etc.) on attack performance. Based on the analytical results, we discuss how the attacker can select attack parameters in order to achieve both attack accuracy and invisibility. We implement the *iLOC* attack and perform experiments on a real-world ITM system, which validate the feasibility of the *iLOC* attack. We also conduct extensive performance evaluations on the *iLOC* attack in a simulated environment. Our evaluations are based on replaying a large set of real-world Internet traffic traces collected by a real-world ITM system. The performance data demonstrate that the attack can accurately identify the locations of monitors, while evading detection by those unaware of the attacker-selected PN-code. Furthermore, we present a set of guidelines on how to counteract the *iLOC* attack.

The remainder of the paper is organized as follows. In Section II, we describe the *iLOC* attack in detail. In Section III, we present a formal analysis of attack accuracy and invisibility, and discuss the impacts of various parameters on the *iLOC* performance. In Section IV, we introduce our implementation of the *iLOC* attack and its validation in the real-world. In Section V, we report our performance evaluation results on the *iLOC* attack. In Section VI, we discuss some preliminary countermeasures against the *iLOC* attack. In Section VII, we discuss related work. Finally we conclude this paper in Section VIII.

## II. *iLOC* Attack

In this section, we discuss the *iLOC* attack in detail. We first give an overview of the *iLOC* attack, and then present the detailed stages of the attack, followed by additional discussions on its mechanisms.

### A. Overview

Fig. 1 shows the basic workflow of the *iLOC* attack. This figure also illustrates the basic idea of the ITM system. In the ITM system, the monitors deployed at various networks record their observed port-scan traffic and continuously update their traffic logs to the data center. The data center first summarizes the volume of port-scan traffic towards (and reported by) all monitors, and then publishes the report data to the public in a timely fashion.

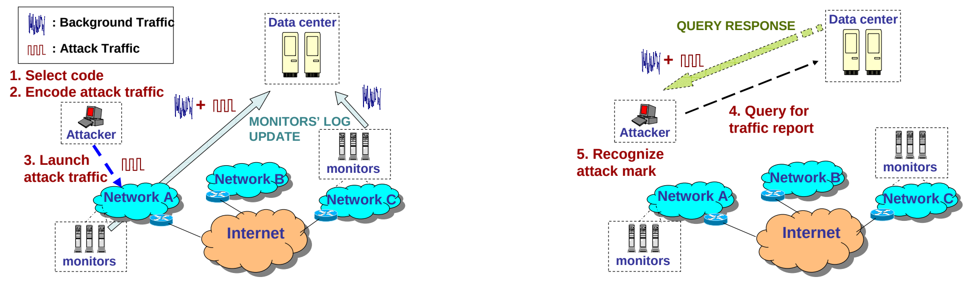
As shown in Fig. 1 (a) and (b) respectively, the *iLOC* attack consists of the following two stages:

*1) Attack Traffic Generation:* In this stage, as shown in Fig. 1 (a), the attacker first selects a code and then encodes the attack traffic by *embedding* the selected code into the traffic. Lastly, the attacker launches the attack traffic towards a target network (e.g., network $A$ in Fig. 1 (a)). We denote such an *embedded code pattern* in the attack traffic as the *attack mark* of the *iLOC* attack, and denote the encoded attack traffic as *attack mark traffic*.

*2) Attack Traffic Decoding:* In this stage, as shown in Fig. 1 (b), the attacker first queries the data center for the traffic report data, which consists of both attack traffic and background traffic. After obtaining the report data, the attacker tries to recognize the attack mark (i.e., the code embedded in the *iLOC* attack traffic) by decoding the report data. If the attack mark is recognized, the report data must include the attack traffic, which means monitors are deployed in the target network and they are sending traffic reports to the ITM data center.

*Code-based Attack:* The *iLOC* attack adopts a code based approach to generate the attack traffic. Coding techniques have been widely implemented in secure communication; for example, *Morse code* is one such example. Without knowledge of Morse code, the receiver would find it impossible to interpret the carried information [14]. In the *iLOC* attack, the PN-code-based approach has three advantages. First, the code is embedded in traffic and can be correctly recognized by the attacker even under interference from background traffic, ensuring accuracy of the attack. Second, the code (of sufficient length) itself provides enough privacy. That is, the code is only known by the attacker, thereby the code pattern embedded in attack traffic can only be recognized by the attacker. Furthermore, the code can carry information. A longer code is more immune to interference, and requires comparatively lower-rate attack traffic as the carrier, which is harder to be detected. All these characteristics help to achieve the objectives of attack accuracy and invisibility.

*Parallel Attack Capacity:* The *iLOC* attack can not only attack *one* target network to determine the deployment of monitors in *one* network at one time, but it can also attack *multiple* networks simultaneously. Intuitively, one simple way to achieve this parallel attack is to launch port-scan/attack traffic towards multiple target networks simultaneously, by scanning a different port number for each different target network. For example, if the data center publishes traffic reports of 1000 (TCP/UDP) ports, then the attacker can launch attack towards 1000 networks simultaneously, attacking each network with a different port number. Since attack traffic on different ports is summarized separately at the data center, the attacker still can separate and thus decode his traffic towards different targets. Hence the attacker can localize monitors in multiple networks simultaneously and accurately. However, can the attacker further improve the attack efficiency? Assume



(a) Attack stage 1: attack traffic generation

(b) Attack stage 2: attack traffic decoding

Fig. 1. Workflow of the *iLOC* Attack

the data center still only publishes reports of 1000 ports, can the attacker attack $10,000$ target networks simultaneously, for example, attacking 10 different networks using *one same* port number? Using high-rate port-scan traffic cannot achieve this, because it is indiscernible whether a spike in the traffic report is caused by traffic logs from one network or the other 9 networks. In order to achieve this goal in the code-based attack, the carefully selected code and corresponding encoded attack traffic towards multiple networks for the same port should not interfere with each other (i.e., each of them can be decoded *individually* and *accurately* by the attacker, although they are integrated and summarized in the traffic report from the ITM data center). The PN-code selected in the *iLOC* attack has this feature, giving it the unique capacity to carry out parallel attack sessions towards multiple target networks using *one same* port. The details of the PN-code selection will be discussed in the following sections.

### B. Attack Traffic Generation Stage

In this attack stage, the attacker: (1) selects the code, a *PN-code* in our case; (2) encodes the attack traffic using the selected PN-code; and (3) launches the encoded attack traffic towards the target network. For the third step, the attacker can coordinate a large number of compromised bots to launch the traffic [15]. However, this is not the focus of this paper. In the following, we will present detailed discussion on the first and second steps, respectively.

*1) Code Selection:* To evade detection by others, the attack traffic should be similar to the background traffic. From a large set of real-world background traffic traces obtained from SANS ISC [3], [16], we conclude that the background traffic shows random patterns in both time and frequency domains. The attack objectives of both accuracy and invisibility, and the attacker's desire for parallel attacks require that: (1) the encoded attack traffic should blend in with background traffic, i.e., be random in both the time and frequency domains, (2) the code embedded in the attack traffic should be easily recognizable to the attacker himself, and (3) the code should support parallel attacks.

To meet the above requirements, we choose the PN-code to encode the attack traffic. The PN-code in the *iLOC* attack is a sequence of $-1$ or $+1$ with the following features [17], [18], [19].

*i)* The PN-code is random and "balanced". The $-1$ and $+1$ are randomly distributed and the occurrence frequencies of $-1$ and $+1$ are nearly equal. This feature yields good spectral density properties (i.e., equally spreading energy over the whole frequency-band). It makes the attack traffic appear as noise and blend in with background traffic in both time and frequency domains.

*ii)* The PN-code has a high correlation with itself and a low correlation with other signals (such as random noise), where the correlation is a mathematical tool for finding repeating patterns in a signal [19]. This feature makes it feasible for the attacker to accurately recognize attack traffic (encoded by the PN-code) from the traffic report data even under interference from background traffic.

*iii)* The PN-code has a low cross-correlation value among different PN-code instances. The lower this cross-correlation is, the less interference will occur among multiple attack sessions in a parallel attack. This feature makes it feasible for the attacker to conduct parallel localization attacks towards multiple target networks on the same port as discussed in Section II-A.

The Walsh-Hadamard code and M-sequence code [17], [18] are two popular types of PN-code. The Walsh-Hadamard code has some limitations. Since its frequency spreads into only a limited number of discrete frequency-components which is different from background traffic, it will compromise the invisibility of the attack traffic if used in the *iLOC* attack. In addition, the Walsh-Hadamard code also strongly depends on global synchronization [18]. However, as the M-sequence code does not have these shortcomings, we adopt M-sequence codes in the *iLOC* attack. We use the *feedback shift register* to repeatedly generate the M-sequence PN-code due to its popularity and ease of implementation [17], [20]. In particular, a feedback shift register consists of two parts. One is an ordinary shift register consisting of a number of flip-flops (two-state memory units). The other is a feedback module which forms a multi-loop feedback logic.

*2) Attack Traffic Encoding:* During the attack traffic encoding process, each bit in the selected PN-code is mapped to a unit time period $T_s$, denoted as *mark bit duration*. The entire duration of launched traffic (referred to as *traffic launch session*) is $T_s \cdot L$, where $L$ is the length of the PN-code.

The encoding is carried out according to the following rules: each bit in the PN-code maps to a mark bit duration ($T_s$); when the PN-code bit is $+1$, port-scan traffic with a high rate, denoted as *mark traffic rate* $V$, is generated in the corresponding mark bit duration; when the code bit is $-1$, no port-scan traffic is generated in the corresponding mark bit duration. Thus, the attacker embeds the attack traffic with a special pattern, i.e., the *original* PN-code. Recall that, after this encoding process, the PN-code pattern *embedded* in traffic is denoted as the *attack mark*. If we use $C_i = \langle C_{i,1}, C_{i,2}, \ldots, C_{i,L}\rangle \in \{-1,+1\}^L$ to represent the PN-code

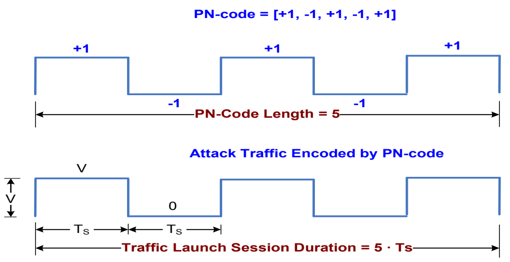

PN-code = [+1, -1, +1, -1, +1]
+1
+1
+1
-1
-1
PN-Code Length = 5
Attack Traffic Encoded by PN-code
V
V
0
Ts
Ts
Traffic Launch Session Duration = 5 · Ts

Fig. 2. PN-code and Encoded Attack Traffic

and use $\eta_i = \langle \eta_{i,1}, \eta_{i,2}, \ldots, \eta_{i,L} \rangle$ to represent the attack traffic, then we have $\eta_{i,j} = \frac{V}{2} \cdot C_{i,j} + \frac{V}{2}$. That is, $\eta_{i,j} = V$ if $C_{i,j} = +1$ and $\eta_{i,j} = 0$ if $C_{i,j} = -1$ $(j = 1, \ldots, L)$. Fig. 2 shows an example of the PN-code and the corresponding encoded attack traffic.

### C. Attack Traffic Decoding Stage

In this stage, the attacker takes the following two steps: (1) The attacker queries the data center for the traffic report data, which consists of both attack traffic and background traffic. (2) From the report data, the attacker attempts to recognize the embedded attack mark. The existence of the attack mark determines the deployment of monitors in the attack targeted network. As querying of traffic report data is relatively straightforward, here we only detail the second step, i.e., attack mark recognition, as follows.

In the report data queried from the data center, the attack traffic encoded with the attack mark is mixed with background traffic. It is critical for the *iLOC* attack to accurately recognize the attack mark from the traffic report data. To address this, we develop the correlation-based scheme. This scheme is motivated by the fact that the original PN-code (used to encode attack traffic) and its corresponding attack mark (embedded in the traffic report data) are highly correlated, in fact, they are actually the same.

The attack mark in the traffic report data is the *embedded form* of the original PN-code. The attack mark is similar to its original PN-code, although the background traffic may introduce interference and distortion into the attack mark. We adopt the following correlation degree to measure their similarity. Mathematically, *correlation degree* is defined as the inner product of two vectors. For two vectors $X = \langle X_1, X_2, \ldots, X_L \rangle$ and $Y = \langle Y_1, Y_2, \ldots, Y_L \rangle$ of length $L$, the correlation degree of vector $X$ and $Y$ is

$$\Gamma(X,Y) = X \odot Y = \frac{\sum_{i=1}^{L} X_i \cdot Y_i}{L}, \tag{1}$$

where $\odot$ represents the operator for the inner product of two vectors. Based on the above definition, we have $\Gamma(X,X) = \Gamma(Y,Y) = 1, \forall\ X, Y \in \{-1,+1\}^L$.

We use two vectors, $\eta_i = \langle \eta_{i,1}, \eta_{i,2}, \ldots, \eta_{i,L} \rangle$ and $\omega_i = \langle \omega_{i,1}, \omega_{i,2}, \ldots, \omega_{i,L} \rangle$ to represent attack traffic (embedded with the attack mark) and background traffic, respectively. We *shift* the above two vectors by subtracting the mean value from the original data, resulting in two new vectors, $\eta'_i = \langle \eta'_{i,1}, \eta'_{i,2}, \ldots, \eta'_{i,L} \rangle$ and $\omega'_i = \langle \omega'_{i,1}, \omega'_{i,2}, \ldots, \omega'_{i,L} \rangle$. We still use a vector $C_i = \langle C_{i,1}, C_{i,2}, \ldots, C_{i,L} \rangle \in \{-1,+1\}^L$ to represent the PN-code. Thus, the correlation degree between the PN-code and the (shifted) attack traffic can be obtained. Similarly, we can also obtain the correlation degree between the PN-code and the (shifted) background traffic as follows.

According to the rules of encoding attack traffic discussed in Section II-B2, $\eta_i = \frac{V}{2} \cdot C_i + \frac{V}{2}$. Thus, $\eta'_i = \eta_i - E(\eta_i) = \eta_i - \frac{V}{2} = \frac{V}{2} \cdot C_i$. Hence, the correlation degree between the original PN-code and the (shifted) attack traffic is $\Gamma(C_i, \eta'_i) = \frac{V}{2} \cdot \Gamma(C_i, C_i) = \frac{V}{2}$. Furthermore, we can also derive the correlation degree between the PN-code and the (shifted) background traffic, i.e., $\Gamma(C_i, \omega'_i)$. The mean of this correlation degree is close to 0, since the PN-code has low correlation with the (shifted) background traffic (i.e., $E[\Gamma(C_i, \omega'_i)] = \frac{1}{L} E[\sum_{j=1}^{L} (\omega'_{i,j} \cdot C_{i,j})] \approx 0$). If the standard deviation of the background traffic rate is $\sigma_x$, the variance of such correlation degree is

$$\begin{aligned} Var[\Gamma(C_i, \omega'_i)] &= E[(\Gamma(C_i, \omega'_i) - 0)^2] \qquad (2) \\ &\approx \frac{1}{L^2} E[\sum_{j=1}^{L} {\omega'_{i,j}}^2] = \frac{{\sigma_x}^2}{L}. \qquad (3) \end{aligned}$$

Thus, the standard deviation of correlation degree between the PN-code and the (shifted) background traffic is $\Gamma(C_i, \omega'_i) \approx \frac{\sigma_x}{\sqrt{L}}$. Based on the above discussion, the attacker can set appropriate attack parameters (e.g., PN-code length $L$ and mark traffic rate $V$) to make the correlation degree $(\frac{V}{2})$ between the PN-code and the attack mark traffic much larger than the correlation degree $(\frac{\sigma_x}{\sqrt{L}})$ between the PN-code and the background traffic. Consequently, the attacker can accurately distinguish the attack mark traffic from the background traffic.

In the attack mark recognition, vector $\lambda_i$ is used to represent the queried report data, and vector $\lambda'_i$ is used to represent the *shifted* report data (by subtracting $E(\lambda_{i,j})$ from $\lambda_i$). According to above discussion, $\lambda'_i = \eta'_i + \omega'_i$ (i.e., report data include the attack traffic and background traffic) or $\lambda'_i = \omega'_i$ (i.e., report data include only attack traffic). The attacker uses the correlation degree between $\lambda'_i$ and his PN-code $C_i$, i.e., $\Gamma(C_i, \lambda'_i)$, to distinguish between the above two cases and determine the existence of PN-code in the report data. If $\Gamma(C_i, \lambda'_i)$ is larger than a threshold $T_a$ [2], which is referred to as the *mark decoding threshold*, then the attacker determines that the report contains attack traffic as well as the PN-code $C_i$, and determines that monitors are deployed in the target network. The accuracy of this correlation-degree-based PN-code recognition is further analyzed and validated in Section III, IV and V.

### D. Discussions

In order to accurately and effectively recognize the attack mark (PN-code) from the report data, we need to find the segment of the report data containing the PN-code (i.e., we need to fulfill the synchronization between the port-scan traffic report data and the PN-code). For this purpose, we introduce an iterative sliding window based scheme. The basic idea is to allow the attacker to obtain enough report data with fine granularity. Then, a sliding window iteratively moves forward to capture a segment of the report data. For each segment, we apply the correlation-based scheme discussed in Section II-C to recognize whether or not the attack mark exists. We skip the details of this synchronization here due to space limitations; interested readers may find them in our technical report [21].

[2] The selection of $T_a$ is impacted by not only the values of $\eta'_i$ and $\omega'_i$, but also by the desired attack accuracy, which is analyzed in Section III.

## III. Analysis

In this section, we first present our formal analysis of the impacts of different attack parameters on attack accuracy and invisibility. Then based on analytical results, we discuss how to determine attack parameters.

Before starting analysis, we need to clarify two parties in the attack process, the *iLOC attacker* and his adversary, the *defender*. The term *defender* generalizes the benign parties who maintain the ITM system and/or exploit the reports from the data center to identify widespread Internet attacks. Based on the reports, the defender not only attempts to determine whether there are anomalies in traffic report, but also takes appropriate actions if any anomaly is identified.

### A. Accuracy Analysis

In order to measure attack accuracy, we introduce the following two metrics. The first metric is the *attack successful rate* $PA_D$, which is the probability that an attacker correctly recognizes the fact that a selected target network is deployed with monitors. The higher $PA_D$ is, the higher the attack accuracy is. The second metric is the *attack false positive rate* $PA_F$, which is the probability that the attacker mistakenly declares a target network as one deployed with monitors. The lower $PA_F$ is, the higher is the attack accuracy.

Recall that $T_a$ is the mark decoding threshold, $V$ is the mark traffic rate, vector $\lambda_i$ represents the queried report data, and vector $\lambda'_i$ represents the *shifted* report data (by subtracting $E(\lambda_{i,j})$ from $\lambda_i$). Assume that random variables $\omega'_{i,1}, \ldots, \omega'_{i,L}$ (i.e., the shifted background traffic) are independent identically distributed (i.i.d) and follow a Gaussian random distribution with standard deviation $\sigma_x$, then we have the following theorem for the attack accuracy of the *iLOC* attack.

*Theorem 1:* In the *iLOC* attack, the attack successful rate $PA_D$ is

$$PA_D = 1 - \frac{1}{\sqrt{\pi}} \int_{\frac{(\frac{V}{2}-T_a)\sqrt{L}}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \qquad (4)$$

The attack false positive rate $PA_F$ is

$$PA_F = \frac{1}{\sqrt{\pi}} \int_{\frac{\sqrt{L}\cdot T_a}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \qquad (5)$$

*Proof:*

*i) Derivation of attack successful rate $PA_D$.*

According to the definition of $PA_D$, we have

$$PA_D \;=\; 1 - Pr[\Gamma(\lambda'_i, C_i) \leq T_a | (\lambda'_i = \eta'_i + \omega'_i)]. \quad (6)$$

Consider that $\Gamma(C_i, \eta'_i) = \frac{V}{2} \cdot \Gamma(C_i, C_i) = \frac{V}{2}$, Equation (6) can be rewritten as

$$PA_D \;=\; 1 - Pr[\Gamma(\lambda'_i, C_i) \leq T_a - \frac{V}{2} | (\lambda'_i = \omega'_i)]. \quad (7)$$

Based on the mean and variance of correlation degree determined in Section II-C, $PA_D$ can be represented by

$$PA_D = 1 - \frac{\sqrt{L}}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{T_a - \frac{V}{2}} e^{\frac{-x^2 L}{2\sigma_x^2}} dx. \qquad (8)$$

Let $y^2 = \frac{x^2 L}{2\sigma^2}$ and $y = \frac{x\sqrt{L}}{\sqrt{2}\sigma_x}$, then we have

$$\begin{aligned} PA_D &= 1 - \frac{\sqrt{L}}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{\frac{(T_a - \frac{V}{2})\sqrt{L}}{\sqrt{2}\sigma_x}} \frac{\sqrt{2}\sigma_x}{\sqrt{L}} e^{-y^2} dy & (9) \\ &= 1 - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{(\frac{(T_a - \frac{V}{2})\sqrt{L}}{\sqrt{2}\sigma_x}} e^{-y^2} dy & (10) \\ &= 1 - \frac{1}{\sqrt{\pi}} \int_{\frac{(\frac{V}{2} - T_a)\sqrt{L}}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. & (11) \end{aligned}$$

*ii) Derivation of attack false positive rate $PA_F$.*

The derivation of $PA_F$ is similar to that for $PA_D$. We skip it here due to space limitations; interested readers can find it in our technical report [21].

■

*Remarks:* We make a few observations based on the above theorem. First, the attack successful rate $PA_D$ increases and the attack false positive rate $PA_F$ decreases with increasing PN-code length $L$. That is, the attack accuracy increases when $L$ increases. Second, the attack accuracy increases as the mark traffic rate $V$ increases.

### B. Invisibility Analysis

Here, *invisibility* refers to how invisible the *iLOC* attack is from the detection of defender. In order to analyze invisibility, we need to consider detection algorithms. While many different algorithms have been proposed to detect anomalies in port-scan traffic, here we use a representative and generic algorithm that has no specific requirement for detection systems. This threshold based detection algorithm is widely adopted by many systems [3], [22]. In this algorithm, if the traffic rate (volume in a given time duration) is larger than a pre-determined threshold $T_d$ (referred to as the *defender detection threshold*), the defender issues threat alerts and responds accordingly [3]. This detection threshold is usually obtained through statistical analysis of the background traffic.

To measure attack invisibility in terms of how well the *iLOC* attack can evade the defender detection, we use the following two metrics. The first metric is the *defender detection rate* $PD_D$, the probability that the defender correctly detects the attack traffic introduced by the *iLOC* attack. The second one is the *defender false positive rate* $PD_F$, the probability that the defender mistakenly identifies the attack traffic.

Similar to our approach in Section II-B2, we use a random variable $\omega'$ to represent the *shifted* background traffic, and a random variable $\lambda'$ to represent the *shifted* traffic data reported by the ITM system. Note that if no *iLOC* attack exists, $\lambda' = \omega'$. Assume that values of $\omega'$ at a different time unit are independent identically distributed (i.i.d) and follow a Gaussian random distribution with standard deviation $\sigma_x$ (i.e., $\omega'$ follows $N(0, \sigma_x^2)$). Then we have the following theorem for attack invisibility.

*Theorem 2:* In the *iLOC* attack, the defender detection rate $PD_D$ is

$$\begin{aligned} PD_D &= 1 - Pr[\lambda' \leq T_d | (\lambda' = V + \omega')] & (12) \\ &= 1 - \frac{1}{\sqrt{\pi}} \int_{\frac{(V - T_d)}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. & (13) \end{aligned}$$

The defender false positive rate $PD_F$ is

$$PD_F = Pr[\lambda' \leq T_d | (\lambda' = \omega')] \quad (14)$$

$$= \frac{1}{\sqrt{\pi}} \int_{\frac{T_d}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \quad (15)$$

We skip the proof of Theorem 2 here due to space limitations; interested readers can find it in our technical report [21].

*Remarks:* From Theorem 2, we make the following observations. First, as the mark traffic rate $V$ increases, the defender detection rate $PD_D$ increases, and thus the attack invisibility decreases. Second, the mark traffic rate $V$ does not affect the defender false positive rate $PD_F$, which is only determined by the defender-configured threshold $T_d$.

### C. Determination of Attack Parameters

*1) Determination of $V$, $T_a$ and $L$:* The attacker can determine the values of attack parameters based on the above analysis. First, the attacker can determine the mark traffic rate $V$. This is because $V$ is only related to the attack invisibility metrics (defender detection rate $PD_D$), and it impacts the determination of other parameters. After determination of $V$ and given the expected false positive rate, the attacker can further determine the mark decoding threshold $T_a$ and PN-code length $L$. Note that the values of other attack parameters (such as the standard deviation of background traffic $\sigma_x$) can be determined by historical background traffic data published by the data center of the ITM system.

*(i) Mark traffic rate $V$:* Using Equation (15), the attacker can first estimate the defender detection threshold $T_d$ based on a reasonable upper bound of the defender false positive rate $PD_F$. For example, using the Central Limit Theorem, we know that $T_d = 3 \cdot \sigma_x$ achieves a reasonable defender false positive rate $PD_F$ (1.7%). Thus, the attacker can use $3 \cdot \sigma_x$ as a reasonable estimation of $T_d$. After that, given the defender detection rate $PD_D$ which can be estimated similarly, and the background traffic deviation $\sigma_x$, the attacker can determine the mark traffic rate $V$ by solving Equation (13) in Theorem 2.

*(ii) Mark recognition threshold $T_a$:* Given the previously-determined mark traffic rate $V$ and desired attack false positive rate $PA_F$, the attacker can further determine the mark decoding threshold $T_a$ by solving Equation (5) in Theorem 1.

*(iii) Length of PN-code $L$:* Given the mark traffic rate $V$, mark decoding threshold $T_a$, and desired attack successful rate $PA_D$, the attacker can further determine the length of PN-code $L$ by resolving Equation (4) in Theorem 1.

*2) Determination of $T_s$:* To determine the mark bit duration $T_s$, the attacker needs to estimate the possible delay from the moment when the attack traffic is first reported by monitors to the moment when this attack traffic is published by the data center. To make the *iLOC* attack effective, the mark bit duration needs to be at least as large as such delay. Otherwise, the traffic in different bit durations (each last $T_s$) may be published at the same moment from the data center, mixing each other and thereby rendering them inseparable.

Several possible methods can be used to obtain such delay information. Some ITM systems may publish this information on their websites. The attacker may also actively conduct experiments on ITM systems and measure the delay. For example, the attacker may deploy monitors in his controlled (small) network and connect them to the targeted ITM system. He can simply use these monitors to report logs embedded with special patterns (e.g., PN-code) and keep querying the data center until the embedded traffic patterns are recognized. After repeating the above process several times, he can obtain the statistics profile of delay information, and then determine the mark bit duration $T_s$. We use this method in our implementation of the *iLOC* attack, which is presented in the next section.

## IV. Implementation and Validation

In this section, we first introduce our implementation of the *iLOC* attack. Then, we report the validation results of our *iLOC* attack design and experiments against a real-world ITM system.

### A. Implementation of the iLOC Attack

We implement an *iLOC* attack prototype based on the design shown in Section II. This prototype works against any ITM system whose data center has a web-based user interface. In particular, there are five independent and important components in our *iLOC* implementation: *Data Center Querist*, *Background Traffic Analyzer*, *PN-code Generator*, *Attack Traffic Generator* and *Attack Mark Decoder*. A detailed description of them is presented in [21].

These components can be integrated into one program running on one machine, as is the case in our experiment. The attack can also be carried out in more flexible ways if the tasks of the above components are performed by processes on different machines. Our *iLOC* prototype is implemented using Microsoft Foundation Classes (MFC) and Matlab on the Windows XP operating system.

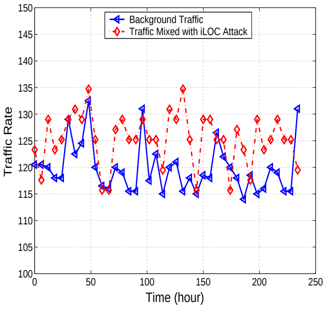


Fig. 3. Background Traffic vs. Traffic Mixed with *iLOC* Attack

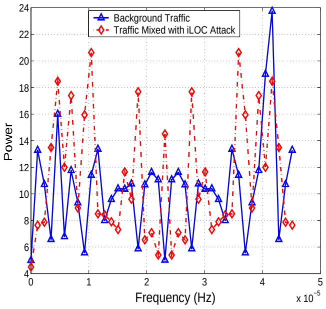


Fig. 4. PSD for Background Traffic vs. Traffic Mixed with *iLOC* Attack

### B. Validation of the iLOC Attack

In order to validate our *iLOC* implementation, we deploy it to identify a set of monitors that are associated with a real-world ITM system.

For the purposes of this research, we requested information about the locations of a set of monitors in an ITM system. We were provided with the identities of two campus network sets $A$ and $B$. There are some monitors deployed within network set $A$ and there is no monitor in network set $B$. All monitors in network set $A$ monitor a set of IP addresses and record the port-scan logs. Then we (as the attacker) execute the *iLOC* attack to decide whether monitors exist in the network set $A$ and set $B$, respectively.

In our experiment, we use a PN-code of length 15. The mark bit duration is set for 2 hours[3]. With the queried report data, we can correctly determine that all networks in set $A$ are deployed with monitors and networks in $B$ are not deployed with monitors. Fig. 3 shows the traffic rate in the time domain. Fig. 4 shows the traffic rate in the frequency domain in terms of *Power Spectrum Density* (*PSD*). The *PSD* describes how the power of a time series data is distributed in the frequency domain. Mathematically, it is equal to the *Fourier Transform* of the auto-correlation of the time series data [23]. From these two figures, we observe that it is hard for others, without knowing the content of PN-code, to detect the *iLOC* attack, since the overall traffic with the *iLOC* attack is very similar to the traffic without the *iLOC* attack traffic. That is, this experiment demonstrates that the *iLOC* attack can accurately and invisibly localize the monitors of ITM systems in practice.

## V. Performance Evaluation

### A. Evaluation Methodology

In our evaluation, we use the real-world port-scan traces from SANS ISC (Internet Storm Center) including the detailed logs from 01/01/2005 to 01/15/2005 [3], [16][4]. The traces used in our study contain over 80 million records and the overall data volume exceeds 80 GB. We use these real-world traces as the background traffic. We merge records of simulated *iLOC* attack traffic into these traces and replay the merged data to emulate the *iLOC* attack traffic. We evaluate different attack scenarios by varying attack parameters. Here, we only show the data on port 135; experiments that use other ports result in similar observations.

We explore both attack accuracy and invisibility to evaluate attack performance. For attack accuracy, we use two metrics: one is the *attack successful rate* $PA_D$ and the other is the *attack false positive rate* $PA_F$, which are defined in Section III-A. For attack invisibility, we use two metrics: one is the *defender detection rate* $PD_D$ and the other is *defender false positive rate* $PD_F$, which are defined in Section III-B.

We evaluate the *iLOC* attack in comparison with two other baseline attack schemes. The first one is the localization attack that launches a very high rate of port-scan traffic towards target networks as introduced in [12], [13]. We denote this attack as *volume-based attack*. The second baseline scheme embeds the attack traffic with a unique frequency pattern. In this attack, the attack traffic rate changes periodically. Then the attacker expects that the report data from the data center will show this unique frequency pattern if the selected target network is deployed with monitors. We denote this attack scheme as *frequency-based attack*. As we will illustrate in the following subsection, this attack scheme has high invisibility in the time domain. However, its invisibility cannot preserved in the frequency domain because there is a unique frequency pattern in the attack traffic. Specifically, when a *Fourier Transform* is applied to a traffic containing a periodic pattern, the periodic pattern emerges as obvious in the frequency domain to the defender.

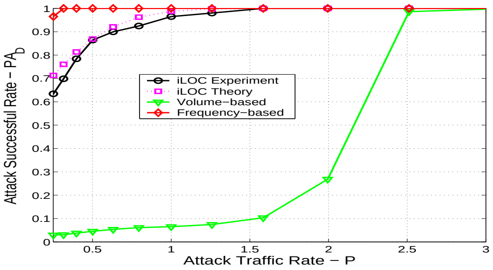


Fig. 5. Attack Successful Rate (Port 135)

In the interest of fairness, we adjust the detection thresholds in all schemes so that the *attack false positive rate* $PA_F$ and *defender false positive rate* $PD_F$ have reasonable values (below $1\%$). For the *iLOC* attack, we generate different attack traffic based on variant PN-code lengths $L$ (i.e., $15, 30, 45$). The default PN-code length is set to 30. To better quantify the attack traffic rate for the *iLOC* attack and other attack schemes, we use the normalized attack traffic rate $P$, which is defined as $P = V/\sigma_x$ for *iLOC* attack, where $\sigma_x$ is the standard variation of background traffic rate.

### B. Results

*1) Attack Accuracy:* To compare the attack accuracy of the *iLOC* attack with those of volume and frequency-based attack schemes, we plot the attack successful rate $PA_D$ under different attack traffic rates (i.e., $P \in [0.01, 3]$) as shown in Fig. 5. From this figure, we observe that both *iLOC* and frequency-based attacks consistently achieve a much higher attack successful rate $PA_D$ than the volume-based scheme. This difference in $PA_D$ is more significant when the attack traffic rate is lower, which can be explained as follows. For the *iLOC* scheme, the PN-code-based encoding and decoding make the recognition of attack marks robust to interference from the background traffic. For the frequency-based scheme, the invariant frequency in the attack traffic is also robust to background traffic interference. Both schemes can distinguish their attack traffic accurately even when the attack traffic rate (i.e., $P$) is small. Nevertheless, the volume-based scheme relies on a high rate of attack traffic (i.e., large $P$), and thus is very sensitive to the interference from background traffic.

*2) Attack invisibility:* To compare the attack invisibility performance of the *iLOC* attack with the that of other two attack schemes, we show the defender detection rate $PD_D$ on port 135 in Table I. This table shows the attacker-achieved defender detection rate $PD_D$ given different localization successful rates $PA_D$ (90%, 95%, and 98%). Recall that the defender sets the detection threshold to make the defender false positive rate $PD_F$ below $1\%$. In the table, "(Time)" and "(Freq)" mean that the defender adopts the *time-domain* and *frequency-domain* analytical techniques to detect attacks. An observation from this table is that our *iLOC* scheme consistently achieves a much lower defender detection rate $PD_D$ than the other two schemes do, which means the *iLOC* attack achieves the best attack invisibility performance. As expected, the defender can easily detect the frequency-based attack by the frequency domain analytical technique, since there is a unique frequency pattern in its attack traffic.

*3) Impact of the Length of PN-code:* To investigate the impact of the PN-code length on the performance of the

[3] The mark bit duration is based on the traffic report publishing delay of the ITM system. If the delay is small, we can set the bit duration to be small. In fact, in order to make the traffic report useful in attack detection for public, the publishing delay needs to be as small as possible such as that in [5].

[4] We thank the ISC for providing us valuable traces in this research.

TABLE I
DEFENDER DETECTION RATE $PD_D$ (PORT 135)

| $PA_D$ | *iLOC*(Time) | *iLOC*(Freq) | Volume-based attack (Time) | Frequency-based attack (Freq) | Frequency-based attack (Time) |
|---|---|---|---|---|---|
| 90% | 2.5% | 2.2% | 90% | 90% | 2.9% |
| 95% | 2.8% | 2.4% | 95% | 95% | 3.1% |
| 98% | 3.1% | 2.8% | 98% | 98% | 3.3% |

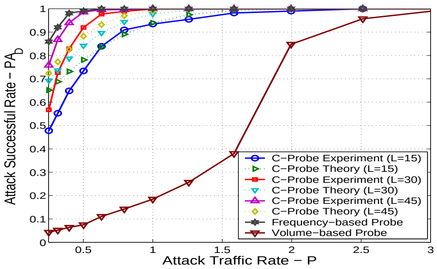


Fig. 6. Attack Successful Rate vs. Code Length

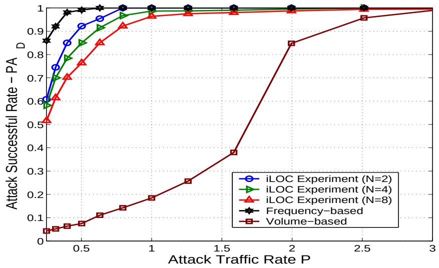


Fig. 7. Attack Successful Rate vs. Number of Parallel Attack Sessions

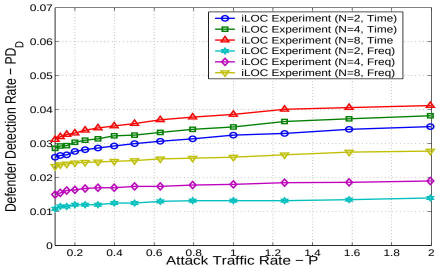


Fig. 8. Defender Detection Rate vs. Number of Parallel Attack Sessions

*iLOC* attack, we plot the attack successful rate $PA_D$ for PN-code of different lengths $(15, 30, 45)$ in Fig. 6. In the legend, *iLOC*$(L = x)$ means that the PN-code length is $x$. Data in this figure are also collected for various attack traffic rates. This figure shows that the attack successful rate $PA_D$ increases as the PN-code length increases. This is because a longer PN-code can more significantly reduce the interference impact of the background traffic on recognizing the attack mark, thereby achieving higher attack accuracy.

*4) Impact of the Number of Parallel Localization Attacks:* To evaluate the impact of the *parallel* localization attacks on the attack accuracy, we show the attack successful rate $PA_D$ for different numbers of parallel attack sessions on the same port in Fig. 7. In the legend, *iLOC*$(N = x)$ means that there are $x$ parallel attack sessions. This figure shows that in terms of the attack successful rate $PA_D$, the *iLOC* attack scheme is insensitive to the number of parallel attack sessions. The attack successful rate $PA_D$ only slightly decreases as the number of parallel attack sessions increases. This is because the traffic for different attack sessions are encoded by PN-codes, which have low cross-correlated to each other as described in Section II-B, and hence there is little interference among them. Fig. 8 shows the impact of the number of parallel attack sessions on attack invisibility. It can be observed that the increasing number of parallel attack sessions results in only a slight increase of the defender detection rate $PD_D$. Therefore, parallel localization capability can improve the attack efficiency without significantly compromising both accuracy and invisibility.

The *iLOC* attack achieves invisibility by using the PN-code, which results in a longer period needed to carry out the attack. Nevertheless, parallel attack capability can significantly improve the attack efficiency. For example, if the attacker launches an *iLOC* attack with a code length of 15 and 8 attack sessions, he needs to spend $\frac{15}{8} = 1.875$ times longer time to finish the attack compared with the volume-based attack, while achieving *significantly* enhanced attack accuracy and invisibility by using the PN-code.

## VI. GUIDELINES OF COUNTERMEASURE

It is relatively easy to defend against volume-based and frequency-based localization attacks. The reason is that they either embed a spike (using high-rate scan traffic) [12], [13] or an invariable frequency (using a certain frequency pattern), and thus show strong signatures in the attack traffic (in either the time domain or frequency domain). However, defending against the *iLOC* attack is much more challenging due to its invisibility feature. In the following, we present some general guidelines for counteracting the *iLOC* attack, while complete countermeasures against it is a part of our future research efforts.

*1) Investigating Advanced Detection Schemes:* Since the *iLOC* attack uses the PN-code to encode the attack traffic, detection of *iLOC* necessitates methods of detecting the PN-code. One potential approach is to adopt Independent Component Analysis (ICA) for PN-code detection and extraction, which relies on investigation of the statistical distortion introduced by the PN-code [24]. We defer the detailed study thereof to our on-going and future work. Except the potential detection approach, we also discuss some proactive countermeasures as follows.

*2) Perturbing the Information:* Recall that in the *iLOC* attack, the attacker has to recognize the encoded attack traffic. Thus, the *quality* of reports plays an important role in this recognition. To reduce the effectiveness of *iLOC* attack, we may *perturb* the published report data by adding some random noise and even randomizing the data publishing delay. This principle is similar to data perturbation in the private data sharing realm [25]. The date center can also confuse the attacker by setting a random and dynamic *dormant monitor set* whose traffic logs will not be aggregated into the *currently* published traffic report. Perturbing report data can degrade the attack accuracy of *iLOC* attack. However, it will also impact the data accuracy and usage of ITM systems. Studying this trade-off will be one aspect of our future work.

*3) Limiting Information Access:* Recall that in the *iLOC* attack, the attacker must query the traffic report from the data center of ITM systems in order to accurately recognize the encoded attack traffic. We may explore this attack behavior feature to reduce the effectiveness of *iLOC* attack. To do so, the data center may throttle the query request rate or require strict authenticated in order to access the traffic report. However, these limitations on information access may also reduce the accessibility and thus the usage of ITM systems.

## VII. RELATED WORK

Many ITM systems have been developed and deployed since CAIDA initiated the network telescope project to monitor background traffic in 2001 [8]. Although the IP addresses of monitors themselves can be protected by mechanisms, such as encryption and the Bloom filter [26], the public data reported by these ITM systems could be used to disclose the IP address space covered by monitors. Existing attack approaches do so by launching high-rate port-scan traffic [12], [13]. However, these attacks do not have invisibility feature, since the high-rate attack traffic is easy to be detected.

The invisibility techniques in our work borrows the camouflage principle, as illustrated in nature and used by the military. In nature, an animal can disguise itself as the object on which it stands in order to fool its predators or preys. In the military, soldiers wear camouflage clothing to blend into the surrounding terrain. As an invisibility technique, our work leverages the PN-code technology and extends it to the Internet cyber-security realm. The PN-code was initially used in military communication systems to provide secure communication that resists jamming [17]. In wireless communication, the PN-code has been widely used to improve communication efficiency [18]. In addition, PN-code has other broad applications, such as cryptography [27], secured data storage and retrieving [28]. There has been some other network security research on achieving similar invisibility goals. For example, Wang et al. in [29] investigated a watermarking scheme that identifies encrypted peer-to-peer VoIP calls via manipulating the timing of packets.

In this paper, we study techniques that apply the PN-code in the *iLOC* attack. Work in [30] also studied how to use the PN-code to effectively track flows through anonymous systems such as mix networks. Since it is applied to a different problem domain, the solution in [30] is significantly different from the one in this paper, including the use of the PN code, designed algorithms, decision rule, and theoretical analysis.

## VIII. Conclusion

In this paper, we investigated a new class of attacks, i.e., the *i*nvisible *LOC*alization (*iLOC*) attack. It can accurately and invisibly localize monitors of ITM systems. Its effectiveness is demonstrated by theoretical analysis and experiments with an implemented prototype. We believe that this paper lays the foundation for ongoing studies of attacks that intelligently adapt attack traffic to defenses. Our study is critical for securing and improving ITM systems.

While the PN-code used in this paper is effective in achieving attack objectives of accuracy and invisibility, other attack patterns embedded in attack traffic may also be accurately distinguished only by the attacker. Detection of such invisible attacks and design of corresponding countermeasures remain challenging tasks and we will investigate them in our future research. Also, we believe that other vulnerabilities exist in ITM systems and we plan to conduct a thorough investigation of them and develop corresponding countermeasures.

## References

[1] D. Moore, C. Shannon, and J. Brown, "Code-red: a case study on the spread and victims of an internet worm," in *Proceedings of the 2-th Internet Measurement Workshop (IMW)*, November 2002.

[2] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–54, 2004.

[3] SANS, *Internet Storm Center*, http://isc.sans.org/.

[4] D. Moore, G. M. Voelker, and S. Savage, "Infering internet deny-of-service activity," in *Proceedings of the 10-th USENIX Security Symposium (SECURITY)*, Auguest 2001.

[5] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson, "The internet motion sensor: A distributed blackhole monitoring system," in *Proceedings of the 12-th IEEE Network and Distributed Systems Security Symposium (NDSS)*, February 2005.

[6] V. Yegneswaran, P. Barford, and D. Plonka, "On the design and utility of internet sinks for network abuse monitoring," in *Proceeding of Symposium on Recent Advances in Intrusion Detection (RAID)*, September 2003.

[7] D. Moore, C. Shannon, G. M. Voelkery, and S. Savagey, "Network telescopes: Technical report," Tech. Rep., Cooperative Association for Internet Data Analysis (CAIDA), http://www.caida.org/publications/papers/2004/tr-2004-04/tr-2004-04.pdf, July 2004.

[8] CAIDA, *The Cooperative Association for Internet Data Center*, http://www.caida.org.

[9] J. Twucrpss and M. M. Williamson, "Implementing and testing a virus throttling," in *Proceedings of the 12-th USENIX Security Symposium*, August 2003.

[10] L. Spitzner, *Know Your Enemy: Honeynets*, Honeynet Project, http://project.honeynet.org/papers/honeynet.

[11] V. Sekar, Y. Xie, D. Maltz, M. Reiter, and H. Zhang, "Toward a framework for internet forensic analysis," in *Proceeding of the 3-th Workshop on Hot Topics in Networks (HotNets)*, November 2004.

[12] J. Bethencourt, J. Frankin, and M. Vernon, "Mapping internet sensors with probe response attacks," in *Proceedings of the 14-th USENIX Security Symposium*, July-August 2005.

[13] Y. Shinoda, K. Ikai, and M. Itoh, "Vulnerabilities of passive internet threat monitors," in *Proceedings of the 14-th USENIX Security Symposium*, July-August 2005.

[14] L. Y. Chuang, C. H. Yang, C. H. Yang, and S. L Lin, "An interactive training system for morse code users," in *Proceedings of Internet and Multimedia Systems and Applications*, August 2002.

[15] R. Naraine, *Botnet Hunters Search for Command and Control Servers*, http://www.eweek.com/article2/0,1759,1829347,00.asp.

[16] Dshield, *Distributed Intrusion Detection System*, http://www.dshield .org/.

[17] R. K. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spead-spectrum communication - tutorial," *IEEE Transaction on Communication*, vol. 30, no. 5, pp. 855–884, 1982.

[18] E. J. Crusellers, M. Soriano, and J. L. Melus, "Spreading codes generator for wireless cdma network," *International Journal of Wireless Personal Communications*, vol. 7, no. 1, pp. 69–88, 1998.

[19] Robert Dixon, *Spread Spectrum Systems, 2nd Edition*, John Wiley & Sons, 1984.

[20] Nova Engineering, *Linear Feedback Register Shift*, http://www.sss-mag.com/pdf/lfsr.pdf.

[21] X. Wang, W. Yu, X. Fu, D. Xuan, and W. Zhao, "*iLOC*: An *i*nvisible *LOC*alization attack to internet threat monitoring systems," Tech. Rep., The Ohio State University, http://www.cse.ohio-state.edu/ wangxu/iloc_tech.pdf, July 2007.

[22] S. Staniford, V. Paxson, and N. Weaver, "How to own the internet in your spare time," in *Proceedings of the 11-th USENIX Security Symposium*, August 2002.

[23] R. L. Allen and D. W. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure*, Wiley and Sons, 2004.

[24] D. Yu, F. Sattar, and K. K. Ma, "Watermark detection and extraction using independent component analysis method," *EURASIP Journal on Applied Signal Processing*, vol. 1, no. 0, pp. 92–104, 2002.

[25] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private database," in *Proceeding of the 22-th SIGMOD International Conference on Management of Data*, July 2003.

[26] P. Gross, J. Parekh, and G. Kaiser, "Secure selecticast for collaborative intrusion detection systems," in *Proceedings of the 3-th International Workshop on Distributed Event-based Systems (DEBS)*, May 2004.

[27] M. Bellare, S. Goldwasser, and D. Micciancio m, "Pseudo-random number generation within cryptographic algorithms: the dss case," in *Proceedings of advances in cryptology'97, Lecture Notes in Computer Science*, Springer-Verlag, May 1997.

[28] L. Wang and B. B. Hirsbrunner, "Pn-based security design for data storage," in *Proceedings of Databases and Applications*, Feberary 2004.

[29] X. Wang, S. Chen, and S. Jajodia, "Tracking anonymous peer-to-peer voip calls on the internet," in *Proceedings of the 12-th ACM Conference on Computer Communications Security (CCS)*, November 2005.

[30] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "Dsss-based flow marking technique for invisible traceback," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, May 2007.