Ranked Leakage-Sensitive Hashing

Rohith Prakash

Abstract

In this paper, we study the problem of real-time anomaly detection of intelligent malware samples disguised within benign applications. We demonstrate that commonly used Dynamic Time Warping (DTW) distance is not suitable on time series of system resource traces when malware samples dynamically adapt their behavior to evade to evade detection. To deal with malware samples in real-time which attempt to hide within benign behavior, we propose a new LSH-based scheme that has low hardware and complexity overhead, iteratively hashes time series based on *behavioral patterns*, triangulates hashing buckets to better categorize behavior, and is able to categorize new, previously unobserved behavior.

1   Introduction

Time series are used almost ubiquitously to represent time-based measurements from any number of avenues. However, it is a well known problem that when these times series represent behavior-related observations of complex systems, unintended information about the system may be leaked through this channel [1, 2]. This problem of side channel leakage has been extensively studied in the past by, and others have proposed correlation-based metrics to quantify the amount of leakage [3, 4].

In this paper, we consider the problem of efficiently learning and classifying the behavior of side channels through observed time series using probabilistic hashing techniques. We formally define the probability of information leakage on a set of such time series traces observed at a fixed granularity with respect to a distance metric. By expanding on that intuition, we propose a *ranked leakage-sensitive hashing* scheme based on previous locality-sensitive hashing schemes [5, 6, 7] that exploits the probability of information leakage for nearest neighbor computations in anomaly detection and behavior classification. We believe that these contributions lead to a novel, strong characterization of leakage over side channels, giving rise to a new notion of *dimensionality* of a channel. We show that this can determine the effectiveness of leakage prevention schemes as well as discuss possibilities of projecting onto higher and lower dimensional spaces to improve anomaly detection or improve the effectiveness of leakage prevention schemes.

2   Background and Definitions

2.1   Information Leakage

Side channel exploitation, anomaly detection, and covert channel communication are problems of detecting or exploiting leakage over information channels. Side and covert channels exists when observable differences in system behavior occur as the result of actions performed by a *victim* or sending process. These attacks typically involve an adversary learning secret information over the channel, based on the behavior of a victim process. Anomaly detection, on the other hand, involves detectors running on a system analyzing and categorizing observed behavior in real time. In this setting, a malicious program leaks information about its behavior through an observed channel.

We consider a single information channel as a sequence of observations of a system resource — a *time series* of resource observations. For example, the *trace* of system calls on a system over time is an n-dimensional time series, where each observation determines the number of times each of the n system calls was invoked.

Our primary observation with regards to time series leakage is that information may only be learned from time series observations if the underlying distributions are distinguishable.

Definition 1 *Distribution-based leakage: Consider two distinct program behaviors x and x' and resulting time series for each behavior drawn from D_x and $D_{x'}$ respectively. Let $t(x)$ and $t(x')$ be two time series resulting from behaviors x, x', drawn from $D_x, D_{x'}$ respectively. Observing $t(x)$ and $t(x')$ can only leak information about x and x' if D_x, $D_{x'}$ are statistically distinguishable.*

While Definition 1 is useful when one can carefully observe many samples from different distributions to assess the distinguishability of the underlying distributions, it is difficult to use in practice. Instead, we propose a slightly different definition of leakage:

Definition 2 *Time series leakage: Consider two distinct program behaviors x and x' with output distributions D_x and $D_{x'}$, and let $d(\cdot, \cdot)$ be a distance function on space of output time series. Let $T_x = \{t_k\}$ be a sequence of time series drawn from D_x. Define r as the minimum distance such that $d(t_i, t_j) \leq r \ \forall \ t_i, t_j \in T_x$*

*We say that observing time series $t(x)$ and $t(x')$ can **leak information with respect to** $d(\cdot, \cdot)$ about x and x' if $d(t(x), t(x')) > r$.*

Definition 2 describes time series leakage with respect to a specific distance function applied on observation points. If time series resulting from behavior x can be separated from time series from x' by a distance of more than r, there is potential information leakage through this channel.

2.2 Hashing

Hashing has long been used as a method of easing the curse of dimensionality for tasks such as clustering on a large set of high-dimensional data [8, 9, 10]. Exploiting the probabilistic nature and the computational efficiency of hashing enables approximations to difficult high-dimensional problems quickly and in real-time.

Consider the space of time series S and a distance function d on S. A LSH family is defined as such:

Definition 3 *A hash family $\mathcal{H} = \{h : S \to U\}$ is called (r_1, r_2, p_1, p_2)-sensitive w.r.t. $d(\cdot, \cdot)$ if for any $x, y \in S$*

   (i) If $d(x, y) \leq r_1$, then $\Pr[h(x) = h(y)] \geq p_1$

   (ii) If $d(x, y) \geq r_2$, then $\Pr[h(x) = h(y)] \leq p_2$

Such a family is only interesting if $p_1 > p_2$. To increase the effectiveness of an LSH technique, the gap between p_1 and p_2 may be *amplified*:

Definition 4 *Consider a (r_1, r_2, p_1, p_2)-sensitive hash family $\mathcal{H}$ w.r.t $d(\cdot, \cdot)$. The LSH hash family can be amplified in the following ways:*

   *(i) **AND construction:** Define $\mathcal{H}' = \{h' : S \to U^r\}$ such that $h' = [h_1, \ldots, h_r] \subset \mathcal{H}$. $h'(x) = h'(y)$ iff $h_i(x) = h_i(y) \ \forall \ h_i \in h' \ \mathcal{H}'$ is a (r_1, r_2, p_1^r, p_2^r)-sensitive LSH family*

(ii) **OR construction:** *Define $\mathcal{H}' = \{h' : S \to U^b\}$ such that $h' = [h_1, \ldots, h_b] \subset \mathcal{H}$. $h'(x) = h'(y)$ iff $h_i(x) = h_i(y)$ for any $h_i \in h'$ $\mathcal{H}'$ is a $(r_1, r_2, 1 - (1-p_1)^b, 1 - (1-p_2)^b)$-sensitive LSH family*

(iii) **AND-OR composition:** *The composition of* and *with* or *constructions defines a $(r_1, r_2, 1-(1-p_1^r)^b, 1-(1-p_2^r)^b)$-sensitive LSH family*

We consider hash families $\mathcal{H} = \{h : S \to \mathbb{R}_+^n\}$ such that $h \in \mathcal{H}$ approximates $d(\cdot, \cdot)$ on time series in S. Intuitively, time series which are "closer" to each other (as defined by d) will be harder to distinguish by any arbitrary classifier and thus present fewer possibilities for information leakage. We expand on this intuition by considering a hashing family $\mathcal{H}$ and hashing function $h \in \mathcal{H}$ which has the following property $\mathscr{A}$:

Definition 5 *Property $\mathscr{A}$: Two hash families $\mathcal{H}_{r_1}, \mathcal{H}_{r_2}$ ($r_1 \leq r_2$ WLOG) have property $\mathscr{A}$ if they are (r_1, r_1, p_1, q_1) and (r_2, r_2, p_2, q_2)-sensitive respectively and the following holds: If $h_{r_1}(x) = h_{r_1}(y)$, then $\Pr[h_{r_2}(x) = h_{r_2}(y)] > \max(1 - q_1, 1 - q_2)$*

Note that the function $d(\cdot, \cdot)$ used in constructing an LSH family need not be a proper metric, but property $\mathscr{A}$ requires that $d(\cdot, \cdot)$ be sub-additive.

Definition 5 allows for a stratified LSH scheme using a set of LSH families index by a distance r which allows us to confer a notion of closeness between buckets of a lower LSH strata.

Definition 6 *A set of hash families $\{\mathcal{H}_r\}_{r \in R}$ is a ranked LSH family if any two H_{r_1}, H_{r_2} have property $\mathscr{A}$ $\forall$ $r_1, r_2 \in R$. Denote such a set of families as (R, p_1, p_2)-sensitive LSH families $\mathcal{H}_R$.*

Proposition 7 *There exists (R, p_1, p_2)-sensitive ranked LSH families $\mathcal{H}_R = \{\mathcal{H}_r\}_{r \in R}$ such that the following hold:*

(i) *$h \in \mathcal{H}_r \in \mathcal{H}_R$ collides with high probability if $y \in B_r(x)$: If $d(x, y) < r, \Pr[h(x) = h(y)] \geq p_1$.*

(ii) *$h \in \mathcal{H}_r$ has few false collisions: If $d(x, y) > r$, $\Pr[h(x) = h(y)] < p_2$.*

(iii) *Property $\mathscr{A}$ holds for any two families $\mathcal{H}_{r_1}, \mathcal{H}_{r_2} \in \mathcal{H}_R$.*

Proposition 7 corroborates the notion of closeness to collision probabilities, which allows for the grouping of similar time series. Due to Property $\mathscr{A}$, we may apply iterative hashing scheme to *rank* the probabilities of closeness based on the varied parameter r. We discuss this in greater depth in Section 3.1

2.3 Distance Metric

Denote the space of a time series as S, and define a function d:

$$
\begin{aligned}
d &: S^2 \to \mathbb{R}_+ \\
(x, y) &\mapsto r
\end{aligned}
\tag{1}
$$

Where d maps two time series to a non-negative real number that represents some notion of distance between them. Additionally, require d to be sub-additive:

$$
d(x, y) \leq d(x, z) + d(z, y)
\tag{2}
$$

The function $d(\cdot, \cdot)$ defines leakage in our threat model.

We consider an arbitrary classification attack on a set of time series. Applying Definitions 1 and 2, there is potential for leakage by observing the resulting time series if and only if the distributions that generate the time series differ on some statistic.

Simple Euclidean distance will not suffice, as it has many limitations for time series comparison [11]. Dynamic Time Warping (DTW) presents a much more accurate method for comparing time series, but it does not satisfy the triangle inequality and is thus not a metric. We could leverage an approximate lower-bound DTW [12] as this satisfies the triangle inequality, but recent literature [5, 6, 7] provide additional metrics which have proven efficiency and accuracy in fields such as image processing and classification.

3   Hashing model for privacy

Consider a database of n samples $D = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ with feature map:

$$\Phi \colon \mathbb{R}^d \to \mathcal{H} \tag{3}$$

where $\mathcal{H}$ is a RKHS. A kernelized hashing scheme for a similarity metric requires that each hash function h_r satisfy:

$$\Pr[h_r(x_i) = h_r(x_j)] = \kappa_r(x_i, x_j) \tag{4}$$

where $\kappa(\cdot, \cdot)$ is a kernel function defined by:

$$\kappa(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \tag{5}$$

We assume κ_r is normalized with respect to distance bias r such that $\kappa_r(\cdot, \cdot) \in [0, 1]$, and κ_r follows properties (i), (ii), and (**??**) of Section 1.

Note that equation 4 confers similar "bounds" to the differential privacy condition [13]:

$$\Pr[f(x) = o] \le e^\epsilon \Pr[f(y) = o] \tag{6}$$

Intuitively, this bounds the probability ratio of an observed output o being the result of inputs x or y to some function f. While the hashing metric in equation 4 does not give a hard bound, it instead allows the evaluation of privacy leakage and the clear separation of time series into ranked buckets by varying distance parameter r.

3.1   Ranked Hashing by Iteration

Consider a hash family $\mathcal{H}$ which approximates a metric for time series distance and is parametrizable by distance threshold r for collision probability (see properties (i), (ii), and (**??**) in Section 1).

Let $\mathcal{R} = \{r_1, r_2, \ldots\}$ be a set of stratifying distance thresholds, and let S be the space of all possible time series. Define an equivalence relation $\sim_r$ on S where $x \sim_r y \Leftrightarrow \kappa(x, y) < r$. Then, π_r is a projection mapping of $\sim_r$ such that $\pi_r \colon X \to X/\sim_r$, where $\pi_r(x) = \pi_r(y)$ iff $\kappa(x, y) < r$.

Iteratively applying LSH with hash functions $h_r \in \mathcal{H}_r$ $\forall r \in \mathcal{R}$ will approximate the projections π_r for each threshold. Let $L_r = \{[h_r(x)], \ldots\}$ be the set of LSH buckets (the set unique hash mappings of h_r). If $h_r \approx \pi_r$, then L_r will closely approximate the quotient $X/\sim_r$ and thus $\{L_r | r \in \mathcal{R}\} \approx \{X/\sim_r\}$.

3.2 Anomaly Detection

What this theory describes is that similar time series can be grouped together and ranked by their closeness by kernel function κ [14]. In the context of anomaly detection, we can apply this scheme to a set of *normal* traces (time series of utilization, label encoded syscalls, etc.) to determine what thresholds and buckets constitute normal execution with a finer granularity. Instead of a binary label of *normal* vs *anomalous*, we can instead stratify applications by types of execution. We speculate that this makes our detection algorithm more robust to adversarial interference.

Given the initial LSH bucketing of normal traces, we can continuously apply the hashing scheme to testing traces over a rolling window to categorize their behavior. Traces which do not match enough previously "normal" buckets over the set of thresholds may be considered to represent anomalous activity.

3.3 Unsupervised, Intelligent Adversary

A similar approach to Section 3.2 can be applied without a predetermined training set for time series classification through clustering. Adjusting the set of threshold values confers the notion of closeness, which may be used to determine which sets of output traces resulted from the same input to a program, for example.

4 Proposed Metrics

Consider the Minkowski distance:

$$M(x, y) = \Big(\sum_k |x_k - y_k|^p\Big)^{1/p}$$

For $p \geq 1$, $M(\cdot, \cdot)$ is a metric due to Minkowski's inequality [15, p. 190]. Additionally, for $p = 2$, this is the standard Euclidean distance. Like the Euclidean distance, this distance metric fails to account for phase and shape changes between time series.

To this end, we consider two different metrics based on a p-dimensional Minkowski distance. The first is due to Batista et al. [16] which attempts to account for time series complexity and has been shown to greatly improve the mean accuracy rates of time series comparison [17]. Consider the following *complexity measure*:

$$C(x) = \sqrt{\sum_i (q_i - q_{i+1})^2} \tag{7}$$

Using this complexity measure, we define a *complexity-invariant* distance metric based on the Minkowski distance $M(\cdot, \cdot)$:

$$d(x,y) = M(x,y) * \frac{\max(C(x), C(y))}{\min(C(x), C(y))} \tag{8}$$

The second metric we consider attempts to correlate Minkowski distance across time with an additional penalty term for time series which are out of phase:

$$d(x,y) = \min_{i,j < \frac{n}{2}} \{ M(x[i:N], y[j:N]) + c * (i+j) \} \tag{9}$$

where x, y have length n and $N = n - \max(i, j)$. c is a scalar penalty factor for out of phase alignment between x and y.

For both of these metrics, we consider p-dimensional Minkowski distance where the original time series have length p.

4.1 Proposed Hash

Let S be the space of n-point time series to be processed. Pick a random lines in $\mathbb{R}^2$ and project each point x_i onto each line h_k. Each line will be partitioned into buckets of size $\frac{r}{n}$ (here, we see the parametrization of the hash). In practice, we fix a number M (say, 2^{32}), and apply $h_r(x) = \frac{\pi(x)*n}{r} \mod M$.

Points are considered to intersect if they hash to the same bucket on a fixed number of hash function. We propose that the number of hash functions on which points must collide to determine intersection be a tunable parameter. Time series which intersect on more than $\frac{n}{2}$ points (in time order) are said to be candidate pairs for r-closeness.

5 Key Takeaways

- Use hash that approximates time series distance.

- Compose multiple kernels or hash families to obtain ranked "normality" metric.

- Apply ranking to classification by an unsupervised classifier to determine amount of leakage possible without prior knowledge.

- Apply ranking to normality (anomaly) detection.

- Perform continuous hashing on rolling windows of execution across channels of multiple resources to classify activity in real-time.

- To be considered: Overlay results with ShapeGD.

References

[1] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, November 9-13, 2009*, pages 199–212. ACM, 2009.

[2] Yinqian Zhang, Ari Juels, Alina Oprea, and Michael K. Reiter. Homealone: Co-residency detection in the cloud via side-channel analysis. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*, pages 313–328. IEEE Computer Society, 2011.

[3] John Demme, Robert Martin, Adam Waksman, and Simha Sethumadhavan. Side-channel vulnerability factor: A metric for measuring information leakage. In *Proceedings of the 39th Annual International Symposium on Computer Architecture*, ISCA '12, pages 106–117, Washington, DC, USA, 2012. IEEE Computer Society.

[4] Tianwei Zhang, Fangfei Liu, Si Chen, and Ruby B. Lee. Side channel vulnerability metrics: The promise and the pitfalls. In *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, pages 2:1–2:8, New York, NY, USA, 2013. ACM.

[5] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1092–1104, June 2012.

[6] Ke Jiang, Q. Que, and B. Kulis. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4933–4941, June 2015.

[7] Y. B. Kim, E. Hemberg, and U. M. O'Reilly. Stratified locality-sensitive hashing for accelerated physiological time series retrieval. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2479–2483, Aug 2016.

[8] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[9] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[10] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, SCG '04, pages 253–262, New York, NY, USA, 2004. ACM.

[11] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recogn.*, 44(9):2231–2240, September 2011.

[12] Daniel Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recogn.*, 42(9):2169–2180, September 2009.

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[14] H. Hachiya and M. Matsugu. Nsh: Normality sensitive hashing for anomaly detection. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 795–802, Dec 2013.

[15] R.L. Wheeden. *Measure and Integral: An Introduction to Real Analysis, Second Edition.* Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis, 2015.

[16] Gustavo E. Batista, Eamonn J. Keogh, Oben Moses Tataw, and Vinícius M. Souza. Cid: An efficient complexity-invariant distance for time series. *Data Min. Knowl. Discov.*, 28(3):634–669, May 2014.

[17] Rafael Giusti and Gustavo E. A. P. A. Batista. An empirical comparison of dissimilarity measures for time series classification. In *Proceedings of the 2013 Brazilian Conference on Intelligent Systems*, BRACIS '13, pages 82–88, Washington, DC, USA, 2013. IEEE Computer Society.