International Journal for Modern Trends in Science and Technology
Volume: 03, Issue No: 05, May 2017
ISSN: 2455-3778
http://www.ijmtst.com

Detection of Sensitive Data Leakage in a Network using Sampling and Alignment Algorithm

S.Maheswari[1] | M.Divya Bharathi[2] | S.Mahalaakshmi[3]

[1,2]UG Scholar, Department of CSE, E.G.S.Pillay Engineering College, Nagapattinam, India.
[3]Assistant Professor, Department of CSE, E.G.S.Pillay Engineering College, Nagapattinam, India.

To Cite this Article
S.Maheswari, M.Divya Bharathi and S.Mahalaakshmi, "Detection of Sensitive Data Leakage in a Network using Sampling and Alignment Algorithm", *International Journal for Modern Trends in Science and Technology*, Vol. 03, Issue 05, May 2017, pp. 295-300.

ABSTRACT

The leak of sensitive data on computer systems stancesa serious threat to organizational security. Statistics show that thelack of proper encryption on files and communications due to humanerrors is one of the leading causes of data loss. Organizationsneed tools to identify the exposure of sensitive data by screeningthe content in storage and transmission, i.e., to detect sensitiveinformation being stored or transmitted in the clear. However,detecting the exposure of sensitive information is challenging dueto data transformation in the content. Transformations (such asinsertion, deletion) result in highly unpredictable leak patterns.In this work, we utilize sequence alignment techniques fordetecting complex data-leak patterns. Our algorithm is designedfor detecting long and inexact sensitive data patterns. This detection is paired with a comparable sampling algorithm, whichallows one to compare the similarity of two separately sampledsequences. Our system achieves good detection accuracy in recognizingtransformed leaks.

Keywords: *Data leak detection, content inspection, sampling,alignment, dynamic programming*

I. INTRODUCTION

Reports show that the number of leaked sensitive datarecords has grown 10 times in the last 4 years, and it reached arecord high of 1.1 billion in 2014 [1]. A significant portion ofthe data leak incidents are due to human errors, for example, alost or stolen laptop containing unencrypted sensitive files, ortransmitting sensitive data without using end-to-end encryptionsuch as PGP. In order to minimize the exposure of sensitive data anddocuments, an organization needs to prevent cleartext sensitivedata from appearing in the storage or communication. Ascreening tool can be deployed to scan computer file systems,server storage, and inspect outbound network traffic. The toolsearches for the occurrences of plaintext sensitive data inthe content of files or network traffic. It alerts users andadministrators of the identified data exposure vulnerabilities. Data leak detection differs from the anti-virus (AV) scanning(e.g., scanning file systems for malware signatures) or thenetwork intrusion detection systems (NIDS) (e.g., scanningtraffic payload for malicious patterns)data leak detection imposes new securityrequirements and algorithmic challenges: Data Transformation and Scalability. The heavy workload of data leak screening isdue to two reasons.a) Long sensitive data patterns. b) Large amount of content.

Existing System

Existing network traffic sampling techniques only sample the content.Existing data leak detection approaches are based on setintersection. Set intersection is performed on two sets of n-grams, one from the content and one from sensitive

data.The set intersection gives the amount of sensitive n-gramsappearing in the content.This method is used to detect similar documents on:

- ✓ Shared malicious traffic pattern.
- ✓ Malware.
- ✓ E-mail spam.

One such mechanism is platform integrity verification for compute hosts that support the virtualized cloud infrastructure. Several large cloud vendors have signaled practical implementations of this mechanism, primarily to protect the cloud infrastructure from insider threats and advanced persistent threats.

To see two major improvement vectors regarding these implementations. First, details of such proprietary solutions are not disclosed and can thus not be implemented and improved by other cloud platforms. Second, to the best of our knowledge, none of the solutions provides cloud tenants a proof regarding the integrity of compute hosts supporting their slice of the cloud infrastructure. To address this, to propose a set of protocols for trusted launch of virtual machines (VM) in IaaS, which provide tenants with a proof that the requested VM.

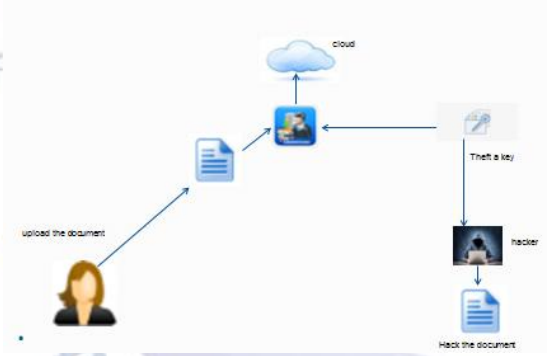


Fig. 1: Existing System for data leakage detection

However, set intersection is order less, i.e., the ordering ofshared n-grams is not analyzed. Thus, set-based detectiongenerates undesirable false alerts, especially when n is setto a small value to tolerant data transformation.

Proposed System

Our solution to the detection of transformed data leaks isa sequence alignment algorithm, executed on the sampledsensitive data sequence and the sampled content being inspected.The alignment produces scores indicating the amountof sensitive data contained in the content. Our alignmentbasedsolution measures the order of n-grams. It also handlesarbitrary variations of patterns without an explicit specificationof all possible variation patterns.We solve the scalability issue by sampling both the sensitivedata and content sequences before aligning them. We enablethis procedure by providing the pair of a comparable samplingalgorithm and a sampling-oblivious alignment algorithm.This one is holding sequential alignment algorithm. Executed on :

- Sampled sensitive data sequence.
- Sampled content being inspected.
- Alignment produces the amount of sensitive data in content.
- More accuracy is achieved.

In this proposed system a “Trusted Cloud Compute Platform” (TCCP) to ensure VMs are running on a trusted hardware and software stack on a remote and initially untrusted host.

To enable this, a trusted coordinator stores the list of attested hosts that run a “trusted virtual machine monitor” which can securely run the client’s VM.

Trusted hosts maintain in memory an individual trusted key use for identification each time a client launches a VM. The paper presents a good initial set of ideas for trusted VM launch and migration, in particular the use of a trusted coordinator. A limitation of this solution is that the trusted coordinator maintains information about all hosts deployed on the IaaS platform, making it a valuable target to an adversary who attempts to expose the public IaaS providerto privacy attacks. Host,beyond the initial launch arguments

A decentralized approach to integrity attestation is adopted by Schiff man. [2] to address the limited transparency of IaaS platforms and scalability limits imposed by third party integrity attestation mechanisms. The authors describe a trusted architecture where tenants verify the integrity of IaaS hosts through a trusted cloud verifier proxy placed in the cloud provider domain. Tenants evaluate the cloud verifier integrity, which in turn attests the hosts. Once the VM image has been verified by the host and countersigned by the cloud verifier, the tenant can allow the launch.

Advantages

- ✓ We describe a trusted VM launch (TL) protocol which allows tenants – referred to as domain managers – to launch VM instances exclusively on hosts with an attested platform configuration and reliably verify this.
- ✓ We describe the implementation of the proposed protocols on an open-source cloud platform.

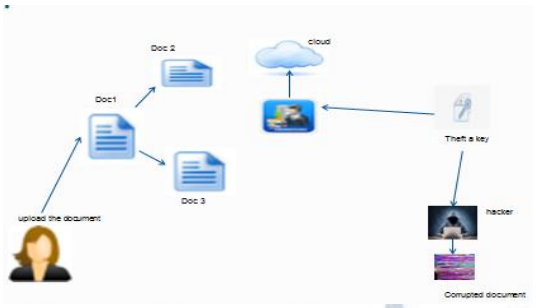


Fig.2:Proposed system for data leakage detection

- ✓ We introduce a domain-based storage protection protocol to allow domain managers store encrypted data volumes partitioned according to administrative domains.
- ✓ We introduce a list of attacks applicable to IaaS environments and use them to develop protocols with desired security properties, perform their security analysis and prove their resistance against the attacks.
- ✓ Our solution detects inadvertent data leaks, where sensitivedata may be accidentally exposed. It is not designed fordetecting data leaks caused by malicious insiders or attackers.

II. RELATED WORK

Existing commercial data leak detection/prevention solutionsinclude Symantec DLP [2], IdentityFinder [3], GlobalVelocity[4], and GoCloudDLP [5]. GlobalVelocity usesFPGA to accelerate the system. All solutions are likely basedon n-gram set intersection.Network intrusion detection systems (NIDS) such asSnort [6] and Bro [7] use regular expression to performstring matching in deep packet inspection [7].However, existing string matching approaches based onDFA or NFA cannot automatically match arbitrary and unpredictablepattern variations.

III. MODELS AND OVERVIEW

In our data leak detection model, we analyze two types ofsequences: sensitive data sequence and content sequence.

- ✓ Content sequence is the sequence to be examined for leaks.The content may be data extracted from file systems onpersonal computers, workstations, and servers; or payloadsextracted from supervised network channels.
- ✓ Sensitive data sequence contains the information (e.g., customers'records, proprietary documents) that needs to beprotected and cannot be exposed to unauthorized parties.The sensitive data sequences are known to the analysissystem.

Pervasive and Localized Modification

Sensitive data could be modified before it is leaked out. The modification can occur throughout a sequence.The modification can also only affect a local region. Some modificationexamples:

- ✓ Character replacement.
- ✓ String insertion
- ✓ Data truncation or partial data leak.

We do not aim at detecting stealthydata leaks that an attacker encrypts the sensitive data secretlybefore leaking it. Preventing intentional or malicious data leak,especially encrypted leaks, requires different approaches andremains an active research problem

Technical Challenges

High detection specificity. In our data-leak detection model,high specificity refers to the ability to distinguish true leaksfrom positives, which may lead to false alarms.

IV. COMPARABLE SAMPLING

One great challenge in aligning sampled sequences is thatthe sensitive data segment can be exposed at an arbitraryposition in a network traffic stream or a file system. Thesampled sequence should be deterministic despite the startingand ending points of the sequence to be sampled. Moreover,the leaked sensitive data could be inexact but similar to theoriginal string due to unpredictable transformations.

Substring

A substring is a consecutive segmentof the original string.If x is a substring of y, one can find a prefix string (denotedby yp) and a suffix string (denoted by ys) of y, so that y equalsto the concatenation of yp, x, and ys. yp and ys could be empty

Comparable sampling

Given a string x andanother string y that x is similar to a substring of y accordingto a similarity measure M, a comparable sampling on x andy yields two sub sequences x0 (the sample of x) and y0 (thesample of y), so that x0 is similar to a substring of y0 accordingto M.

If we restrict the similarity measure M in Definition 3 toidentical relation,

Example 1.

Algorithm :A subsequence-preserving sampling algorithm.

Input:n array S of items, a size jwj for a sliding window

w,aselection function f(w;N) that selects N smallest items froma window w, i.e., f =in(w;N)

Inputs:1 1 9 4 5 7 3 5 9 7 6 6 3 3 7 1 6
1 1 9 4 5 7 3 5 8 6 6 3 7 1 6 1 4

Comparable sampling may give:

1 1 - 4 - - 3 5 - - - - 3 3 - 1 -

1 1 - 4 - - 3 - - 6 - 3 - 1 - 1 4

Random sampling may give:

1 - - 4 - - 3 5 - 7 - 6 - - 7 1 -

- 1 9 - 5 - - 5 - 6 - 3 7 - 6 1

Output: a sampled array T

The advantageof our algorithm is its context-aware selection, i.e.,the selection decision of an item depends on how it compareswith its surrounding items according to a selection function.As a result, the sampling algorithm is deterministic andsubsequence-preserving.

Algorithm :

1: initialize T as an empty array of size jSj
2: w read(S; jwj)
3: let w:head and w:tail be indices in S corresponding to the
higher-indexed end and lower-indexed end of w, respectively
4: collection mc min(w;N)
5: while w is within the boundary of S do
6: mp mc
7: move w toward high index by 1
8: mc min(w;N)
9: if mc 6= mp then
10: item encollectionDi_(mc;mp)
11: item eocollectionDi_(mp;mc)
12: if en<eo then
13: write value en to T at w:head's position
14: else
15: write value eo to T at w:tail's position
16: end if
17: end if
18: end while

We show how our sampling algorithm works in Table I.We set our sampling procedure with a sliding window ofsize 6 (i.e., jwj = 6) and N = 3. The input sequence is1,5,1,9,8,5,3,2,4,8. The initial sliding window w =[1,5,1,9,8,5] and collection mc = f1,1,5g.

Sampling Algorithm Analysis

Our sampling algorithm is deterministic, i.e., given a fixedselection function f: same inputs yield the same sampledstring. However, deterministic sampling (e.g., [3]) does notnecessarily imply subsequence preserving

TABLE I
Illustration Our Sampling Procedure.

Step	w	Mp	Mc	En	eo	Sampled List
0	[1,5,1,9,8,5]	1,1,5	N/A	N/A	N/A	<-,-,----->
1	[5,1,9,8,5,3]	1,3,5	1,1,5	3	1	<1,-,---->
2	[1,9,8,5,3,2]	1,2,3	1,3,5	2	5	<1, -, -, -, -, -, -, 2, -, ->
3	[9,8,5,3,2,4]	2,3,4	1,2,3	4	1	<1, -, 1, -, -, -, -, 2, -, ->
4	[8,5,3,2,4,8]	2,3,4	2,3,4	N/A	N/A	<1, -, 1, -, -, -, -, 2, -, ->

V. ALIGNMENT ALGORITHM

A. Requirements and Overview

We design a specialized alignment algorithm that runs oncompact sampled sequences La and Lb to infer the similaritybetween the original sensitive data sequence Sa and theoriginal content sequence Sb. It needs to satisfy the requirementof sampling oblivion, i.e., the result of a samplingobliviousalignment on sampled sequences La and Lb shouldbe consistent with the alignment result on the original Sa andSb.Conventional alignment may underestimate the similaritybetween two substrings of the sampled lists, causing misalignment.Regular local alignment without the sampling oblivion property may give inaccurate alignment on sampled sequencesas illustrated in Example 2.

Example 2.

Sampling-oblivious alignment vs. regular localalignment

Original lists:
562798385743254639782436 6
5627983966432546395

Sampled sequences need to be aligned as:
--2---3-5---2---3-7-2-3--
--2---3-6---2---3--

However, regular local alignment may give:
23523723
23623

We develop our alignment algorithm using dynamic programming.A string alignment problem is divided into threeprefix alignment sub problems: the current two items (fromtwo sequences) are aligned with each other, or one of themis aligned with a gap.

In our algorithm, not only the sampleditems are compared, but also comparison outcomes betweennull regions are inferred based on their non-null neighbouringvalues and their sizes/lengths. The comparison results includematch, mismatch and gap, and they are rewarded (match) orpenalized (mismatch or gap) differently for sampled items ornull regions according to a weight function fw().Our alignment runs on sampled out elements. We introduce:

i) extra fields of scoring matrix cells in dynamic programming,

ii) extra steps in recurrence relation for bookkeeping thenull region information, and

iii) a complex weight functionestimating similarities between null regions.

Security Advantages of Alignment

There are three majoradvantages of our alignment-based method for detecting data leaks: order-aware comparison, high tolerance to pattern variations, and the capability of partial leak detection.

Algorithm 2: Recurrence relation in dynamic programming.

Input: A weight function fw, visited cells in H matrix
that areadjacent to H(i; j): H(i-1; j-1);H(i; j-1),
and H(i-1; j),and the ith and j-th items Li^aLj^bn
two sampled sequences Laand Lb, respectively.
Output: H(i; j)

1:hh^{up}:score←fw(L_i^a,-,H(i-1,j))
2: h^{left}.score←fw(-,L_j^b,H(I,j-1))
3: h^{dia}.score←fw(Li^a, L_j^b,H(I-1,j-1))
4: h^{up}.nullrow←0
5: h^{up}:nullcol←0
6: h^{left}:nullrow←0
7: h^{left}.nullcol←0
8: h^{dia}:nullrow←0;
ifLi^a==L_j^b
H(i-1,j).$null_{row}$,+Li^a.span+1,
else
9: h^{dia}:nullcol← 0;
if Li^a==L_j^b
H(i,j-1).$null_{col}$+ L_j^b.span+1,
else
10: H(i,j)←$arg_{h.score}$max h^{up},h^{left},h^{dia}
11.H(I,j).score← max 0 H(I,j).score

Security Advantages of Alignment

There are three majoradvantages of our alignment-based method for detecting dataleaks: order-aware comparison, high tolerance to pattern variations,and the capability of partial leak detection. All featurescontribute to high detection accuracy.Order-aware comparison

- ✓ High tolerance to pattern variation
- ✓ Capability of detecting partial leaks.

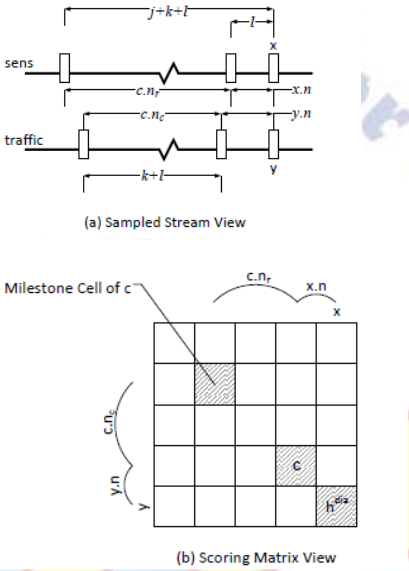


Fig. 3Scoring and stream view

VI. EVALUATION ON DETECTION ACCURACY

We extensively evaluate the accuracy of our solution withseveral types of datasets under a multitude of real-world dataleak scenarios.

TABLE III

Semantics of true and false positives and true and falsenegatives in our model.

Leak	True Leak	No Leak
Leak detected	TP	FP
No Leak detected	FN	TN

We implement a single-threaded prototype (referred to asAlignDLD system) and a collection intersection method (referredto as Coll-Inter system), which is a baseline.We use four datasets (Table II) in our experiments. A. Enronand B. Source-code are used either as the sensitive data orthe content to be inspected. C. Outbound HTTP requests andD. MiscNet are used as the content.

Detailed usages of thesedatasets are specified in each experiment.

We report the detection rate in Equation (1) with respect toa certain threshold for both AlignDLD and Coll-Inter systems.The detection rate gives the percentage of leak incidents thatare successfully detected. We also compute standard falsepositive rate defined in Equation (2). We detail the semanticmeaning for primary cases, true positive (TP), false positive(FP), true negative (TN), and false negative (FN), in Table III.

Detection rate (Recall) =TP/TP + FN(1)

False positive rate =FP/FP + TP(2)

The detection systems intercept the outbound networktraffic, perform deep packet inspection, and extract the contentat the highest known network layer4.

Then the detectionsystems compare the content with predefined sensitive datato search for any leak.

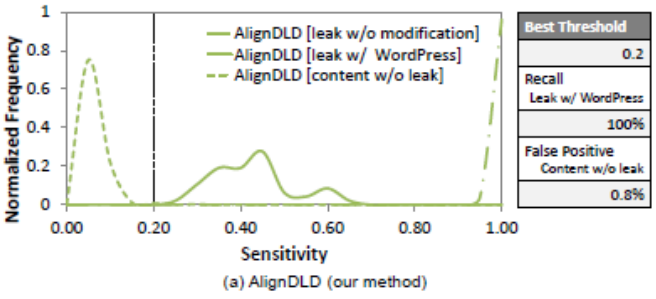
1) Web leak: a user publishes sensitive data on the Internetvia typical publishing services, e.g., WordPress,

2) FTP: a user transfers unencrypted sensitive files to an FTPserver on the Internet,

3) Backdoor: a malicious program, i.e., Glacier, on theuser's machine exfiltrates sensitive data,

4) Spyware: a Firefox extension FFsniFFexfiltratessensitive information via web forms.

It is not a challenge to detect intact data leaks. OurAlignDLD system successfully detects intact leaks in all theseleaking scenarios with a small sampling rate between 5% and20%.



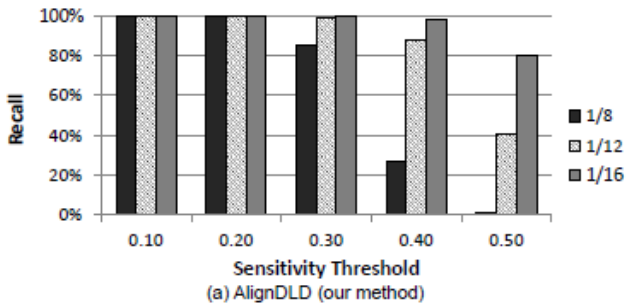


Fig 4.AlignDLD

VIII. CONCLUSIONS AND FUTURE WORK

We presented a content inspection technique for detectingleaks of sensitive information in the content of files or networktraffic. Our detection approach is based on aligning twosampled sequences for similarity comparison. Our experimentalresults suggest that our alignment method is useful fordetecting multiple common data leak scenarios. The parallelversions of our prototype provide substantial speedup andindicate high scalability of our design. For future work, weplan to explore data-movement tracking approaches for dataleak prevention on a host.

Future enhancement

The detection of data leaks due to malicious insiders remainsa challenging open research problem.

REFERENCES

[1] RiskBasedSecurity, "Data breach quickview: 2014 data breach trends,"February 2015.

[2] Symantec, "Symantec data loss prevention," 2015,http://www.symantec.com/data-loss-prevention, accessedFebruary 2015.

[3] Identifyfinder, "Identity Finder," 2015,http://www.identityfinder.com/,accessed February 2015.

[4] GTB Technologies Inc., "GoCloudDLP," 2015,http://www.goclouddlp.com/, accessed February 2015.

[5] M. Roesch, "Snort - lightweight intrusion detection fornetworks," inProceedings of the 13th USENIXConferenceon System Administration,ser. LISA '99,

[6] V. Paxson, "Bro: A system for detecting network intrudersin real-time,"in Proceedings of the 7th Conference on USENIX Security Symposium,ser. SSYM'98, vol. 7,1998, pp. 3–3.

[7] P.-C. Lin, Y.-D.Lin, Y.-C.Lai, and T.-H. Lee, "Usingstring matchingfor deep packet inspection," IEEEComputer, vol. 41, no. 4, pp. 23–28,2008.