Data security Implementation for Real Time Internet Packet Traces

Anantha Narasimhan S.
Department of Information Technology
Sri Venketeswara College of Engineering
Sriperumbudur,TN INDIA.
anantha.narasimhan@gmail.com

Sundarram P.V.
Department of Information Technology
Sri Venketeswara College of Engineering
Sriperumbudur,TN INDIA.
pvsundarram@gmail.com

ABSTRACT: The present data transfer and security system will no longer be robust to support the data volumes. There has always been a criticism on the transparency of the existing data security and network security systems. Researchers have been working on new methods of establishing secure data transmission, but in vain. The main reason for their failure may be attributed to the non-availability of the vital “Packet traces” recording real-world Internet traffic. These real time “packet traces are obtained from the internet “tcpdump” and are especially useful for research on traffic dynamics, protocol analysis, workload characterization, and network intrusion detection. However, sharing of Internet packet traces is very limited because real-world traces contain many kinds of sensitive information, such as host addresses, emails, personal web pages, and even authentication keys. The lack of such traces greatly limits research on application protocols. It is especially crippling for network intrusion detection research, forcing researchers to devise synthetic attacks.

In this paper we describe a approach to transform and anonymize packet traces. The paper elaborates on the anonymization of the internet packet traces and corresponding trace transformation. The algorithm discussed can anonymize both packet headers and payloads, and can perform application-level transformations such as editing HTTP or SMTP headers, replacing the content of Web items with MD5 hashes, or altering filenames or reply codes that match given patterns.

The paper also goes into solving problems that are discussed in detailed by allied literature. The paper mainly concentrates on the methodology of solving problems related to file transfers and anonymization issues. The huge volumes of file transfer that takes place all over the network are recorded by the trace of the ftp activity on the server. This poseses a potential threat for the network administrators. The paper discusses a new method for anonymizing ftp traces and opens the gates for a new era of high level programming support for the customization of the entire activity of anonymization and supports writing optimized transformation scripts.

Thus the paper aims to shed light on a new trace transformation & anonymization techniques with features for the future, coupled with reliability and frugal use of resources take technology to the masses as well as the researchers, making the world a truly global village. As such, we hope to help open up new opportunities in Internet measurement and network intrusion detection research.

Keywords: packet trace, anonymization, transformation, Internet, privacy, measurement, network intrusion detection
INTRODUCTION

In this information age the technology is developing at a rapid rate, the users increased manifolds and newer forms of threats have been exposed. The researchers are unable to provide a secure mechanism to be implemented in very vital areas such as Network intrusion detection, network management and traffic dynamics. The reason for this is attributed to the non-availability of research material. Though one might point out the World Wide Web, the Internet as a perennial source of information the research scientist are looking for real world Internet packet traces. These packets contain the details about the various transactions across the Internet. These internet packet traces are obtained from the tcp dump that is present the servers and gateways. These are similar to the system logs in a stand alone system. The main advantage of this is that these are real world values. Fortunately or unfortunately the tcp dump contains many sensitive information like the IP addresses and private information like ID ’s and passwords. Though this can be a very potential resource for the development of tools that can effectively be used to counter the present day threat in modern networks, this is never available to the researchers . The main reason is that many fear to contribute their tcp dump to them as they fear about the publicity of the private information present in it. It would become a great threat if these go into wrong hands.

The need of the hour is a methodology by which we can use this resource and exploit its power at the same time safeguarding the private information present in it. With its use newer tools can be created for Intrusion detection and traffic dynamics analysis.

Traces are very useful for the researchers as it gives them the true picture of the real-world scenario. From the picture they are able to get newer ideas for developing counter mechanism for network intrusion ,etc.

The aim of this paper is to develop a method by which these internet packets are made available to the researchers at the same time preserving the integrity of the private information and confidential matter present in the trace.

This paper discusses a scheme which is capable of anonymizing the sensitive information and instills confidence in the minds of people that their private information is not made publicly available. The paper discusses an algorithm that reorganizes the packet traces and the meaningful data elements are then processed by the anonymization procedure followed by the transformation. As a result of this the internet packet traces are now made available to the researchers which is devoid only of private information. The required dynamics are preserved and the relative meaning of the payload is preserved.

This system differs from its predecessors from the fact that the principle used to remove or rather hide private information is the filter-in principle and not the filter out principle.

The former retains the implied relation between the payload and header and the total transaction whereas the later removes the entire payload. The sensitive information is retained but in a safe unrevealing form. Due to this more meaning is implied and thus research can be carried out effectively.

1 OVERVIEW

The traces are received in form that cannot be processed directly. Therefore the traces are initially converted into or reassembled into semantically meaningful application protocol level data elements.

The data elements are then operated upon by the anonymization algorithm followed by trace transformation to make it coherent. Anonymization is carried out both payloads as well as the headers.

The anonymized data entities are then reconstructed to their original form .This is done by the composer. The trace composer along with the framer is used to construct the original packet traces as if they were obtained from the server’s tcp dump but devoid of private information.

The output traces contain well-formed connections: packets have correct checksums and lengths, TCP flows can be reassembled from the resulting packets, and application-protocol data has correct syntax, so that other programs can process the transformed traces in the same way that they handle original tcpdump traces.
The entire process is summed up as follows:

Initially the packet traces are given as the input to the system. The flow reassembler then reassembles the packets into semantically meaningful data entities. As these elements have private information an algorithm operates upon the trace. The algorithm will anonymize the payloads IPs and also the header information and this will also be transformed. During this transformation the correct checksums are computed, length is chosen correctly. After all this is done the obtained data elements are devoid of private information and is now ready for research purposes. But they have to be restructured again in order to get the real meaning from it. After reorganization the packet’s transport protocol dynamics are preserved.

1.1 ANONYMIZATION / TRANSFORMATION

Anonymization means “making unknown or unidentifiable”, therefore by anonymizing the internet packet traces we aim at destroying the private information present in the trace and at the same time preserving the inherent meaning of the information in a more secure and undisputed form.

The information we try to hide through anonymization falls into two categories: identities, including identity of users, hosts, and data; and confidential attributes, e.g., passwords, or specifics of sensitive user activity. The first step of developing an anonymization scheme is to decide what information in the trace we need to hide. For example, in anonymizing FTP traces, we aim to hide identities of clients, private data (hidden files), and private servers; and sensitive attributes: e.g., passwords, authentication keys, and in some cases filenames.

The anonymization scheme is very easy to understand as we discuss the various methods available for doing it. The methods that can commonly be implemented are

1. Substitution by Constant Values
2. Numbering in a Sequential Order
3. Hashing
4. Prefix-Preserved Mapping

1.1.1 Substitution by Constant Values

One way to anonymize a data element is to substitute the data with a constant, e.g., replace any password with the string “<password>”. Constant substitution is usually used to anonymize confidential attributes. Applying constant substitution to identifiers (e.g., IP addresses), however, is generally undesirable, as we would then no longer be able to precisely distinguish objects from one another. Instead, identifiers are usually anonymized with a 1-1 mapping, such as sequential numbering or hashing, so that the anonymized identifiers are still unique, as follows.

1.1.2 Numbering in a Sequential Order

We can sequentially number all distinct identifiers in the order of appearance, e.g., mapping files names to “file1”, “file2”, etc.

1.1.3 Hashing

One shortcoming of sequential numbering is that we have to keep the whole mapping history to maintain a consistent mapping during the anonymization process and across anonymizations. Instead, we can use a hash function as the mapping. Doing so requires no additional state during the anonymization process, and in addition using the same hash function across anonymizations will render a consistent mapping (assuming that the range of the hashing function is large enough so that likelihood of collision is negligible). To preserve confidentiality, the hash function must be one-way and preferably resistant to chosen plain-text attack, so that an adversary can neither discover the input from the output nor compute the hash by themselves. HMAC-MD5 (with a secret key) satisfies these requirements. Assuming the adversary can neither reverse MD5 nor extract the secret HMAC key, hashing is as secure as sequential numbering.

1.1.4 Prefix-Preserving Mapping

Sometimes it is valuable to preserve some of the structural relationships between the identifiers, which sequential numbering and hashing cannot do. For example, IP addresses can anonymized in a prefix-preserving way such that any two IP addresses sharing a prefix will share a prefix of the same length in their
anonymized form. Prefix-preserving mapping can be similarly applied on the directory components of file names.

1.1.5 Trace transformation

Trace transformation is also carried out with anonymization. Trace transformation aims at the transforming the data present from one form to a more coherent form after anonymization process is done. This involves the calculation and correcting of the checksums and lengths of header fields. Trace transformation can be implemented as required by writing short policy scripts. This policy script should adhere to certain constraints so as to make it amenable for verification. The trace transformation is in no way related to the strength of the privacy factor of the system. It is just used to bring about more reality into the anonymized traces and improved cohesion in it.

2 OUR NEW ALGORITHM

We will now discuss the algorithm that we have developed for the effective anonymization and trace transformation of the vital, sensitive internet packet traces

2.1.1 ALGORITHM

STEP 1: Open tcpdump for reading data in it

STEP 2: Call the reconstruction procedure to reorganized packets and to generate semantically meaningful data entities.

STEP 3: Identify the part of the trace to be anonymized and select type respectively

STEP 4: If it is IP anonymization use prefix preserving mapping scheme

STEP 5: If it is the anonymization of payload then use MD5 hashing to digest the payload

STEP 6: Use REWRITE FUNCTION to write the new data

STEP 7: Call the TRANSFORMATION procedure to correct checksum and header length fields

STEP 8: Use DEFFERED write to update newer values

STEP 9: Reconstruct the trace into packets with the composer

STEP 10: Validate output packets

STEP 11: If ok then exit with success

STEP 12: If failed then goto STEP 2

From [1] we get a great insight into the drawbacks of the current proposals. Thus we will now shift our focus on to developing a robust scheme for including FTP (file transfer protocol) traces into the anonymization process. This will improve the efficiency of the system and make the contributed traces immune against any possible data leak and subsequent attacks.

Our objectives are: 1) ensure that the anonymization hides the identity of clients, non-public FTP servers, and non-public files, as well as confidential authentication information; and 2) the anonymization keeps the original request/reply sequence and other nonsensitive information intact. In some ways, these goals and the resulting traces are quite modest. But we believe that the path to site’s becoming open to releasing traces with packet contents is one that must be tread patiently,
The algorithm described here will perform anonymization of the Internet packet traces making them absolutely safe and hence openly available for research activities. Trace transformation is also done together in order to bring in integrity of the trace and thus preserving its original protocol dynamics.

3 SAMPLE INPUT (Before Anonymization)

848278028:829593  848278028:893670
848278028:895350  172.16.3.5:8082
172.16.3.3:8080  2  8  4294967295
4294967295  835418853  170 844
35  GET google.com/svcelogo.gif  HTTP/1.0

- 848278028:829593 is the time at which the client made the request
- 848278028:893670 is the time at which the first byte of the server response was seen
- 848278028:895350 is the time at which the last byte of the server response was seen
- 172.16.3.5:8082 is the client IP address and the client port number
- 172.16.3.3:8080 is the server IP address and the server port number
- 2 is the decimal representation of the client headers bitfield
8 is the decimal representation of the server headers bitfield
the first 4294967295 is the if-modified-since client header value
the second 4294967295 is the expires server header value
835418853 is the last-modified server header value
170 is the length of the HTTP response header
844 is the length of the response data
31 is the length of the request URL
"GET google.com/svce.gif HTTP/1.0" is the request URL.

4 SAMPLE OUTPUT (After Anonymization)

848278028:829593 848278028:893670
848278028:895350 23.240.8.98:1462
207.36.205.194:80 2 8 4294967295
4294967295 835418853 170 844
37 GET 9168504434183313441..gif HTTP/1.0

848278028:829593 is the time at which the client made the request
848278028:893670 is the time at which the first byte of the server response was seen
848278028:895350 is the time at which the last byte of the server response was seen
23.240.8.98:1462 is the anonymized client IP address and the client port number
207.36.205.194:80 is the anonymized server IP address and the server port number
2 is the decimal representation of the client headers bitfield
8 is the decimal representation of the server headers bitfield
the first 4294967295 is the if-modified-since client header value
the second 4294967295 is the expires server header value
835418853 is the last-modified server header value
170 is the length of the HTTP response header
844 is the length of the response data
37 is the length of the anonymized request URL
"GET 9168504434183313441..gif HTTP/1.0" is the anonymized request URL.

5 Experimental Implementation

Processing of traces from GOOGLE web server
The traces required for processing were collected from trace donating web servers. Some web servers contribute sanitized Traces for research activities. The original trace was collected by tcpdump recording a retrieval of the www.google.com homepage. The tcpdump output (with wrapped packet summary lines and TCP payloads) of the original trace is in the APPENDIX. The algorithm does the following

1. Replaces the data entity with itsMD5 hash value
2. Rewrites the Content-length field to reflect the length of the MD5 hash value.
3. Adds the header: "X-Actual-Data-Length, gap, content-length to record the original Content-length field and how many bytes are actually transferred.
4. The algorithm replaces every occurrence of “Google” in the data entity with “Goooogle”, instead of replacing the whole data entity with its MD5 hash value.

5.1 Analysis of the results

1. Data Entity replaced by “867119294265e3f445708c3fcbf214f”, the MD5 hash value.
2. Modification in header is as: “X-Actual-Data-Length: 2709; gap=0, content-length=2709”
3. There are four occurrences of “Google” in the original message, thus the Content-length increases from 2709 to 2717.

5.2 Execution time of our system

<table>
<thead>
<tr>
<th>System Configuration</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP analyzer</td>
<td>2358 seconds</td>
</tr>
<tr>
<td>HTTP analyzer + anonymizer</td>
<td>18162 seconds</td>
</tr>
<tr>
<td>HTTP analyzer + dummy rewriter</td>
<td>3456 seconds</td>
</tr>
</tbody>
</table>

6 System Analysis

6.1 Strengths

- Meaning intact
- Sensitive information is filtered in
- Original Dynamics are preserved

6.2 Weakness

- Not standardized
- Technology in nascent stage
- Requires higher Processing Time
- Storage is difficult

7 Further Scope and Development

The first and foremost aim of this paper is to make an effort to bring about the availability of the research material to the scientist. With these material the research activity will boom and there is huge scope for the development of many useful techniques in security and traffic aspects. The paper supports a noble cause of contributing a research to aid researchers. As we can generate stronger methods of anonymization many will come forward confidently to contribute their tcpdump, thus promoting research.
Bibliography


7. Traffic Measurements for Link Dimensioning* A Case Study Remco van de Meent Aiko Pras Michel Mandjes Hans van den Berg Lambert Nieuwenhuis 13th August 2003 University of Twente CWI

8. Observed Structure of Addresses in IP Traffic Eddie Kohler Jinyang Li, Vern Paxson, Scott Shenker ICSI Center for Internet Research MIT Lab for Computer Science