MET: An Experimental System for Malicious Email Tracking

Manasi Bhattacharyya
Department of Computer Science
Columbia University
mb551@cs.columbia.edu

Shlomo Hershkop
Department of Computer Science
Columbia University
shlomo@cs.columbia.edu

Eleazar Eskin
Department of Computer Science
Columbia University
eeskin@cs.columbia.edu

ABSTRACT

Despite the use of state of the art methods to protect against malicious programs, they continue to threaten and damage computer systems around the world. In this paper we present MET, the Malicious Email Tracking system, designed to automatically report statistics on the flow behavior of malicious software delivered via email attachments both at a local and global level. MET can help reduce the spread of malicious software worldwide, especially self-replicating viruses, as well as provide further insight toward minimizing damage caused by malicious programs in the future. In addition, the system can help system administrators detect all of the points of entry of a malicious email into a network. The core of MET's operation is a database of statistics about the trajectory of email attachments in and out of a network system, and the culling together of these statistics across networks to present a global view of the spread of the malicious software. From a statistical perspective sampling only a small amount of traffic (for example, .1 %) of a very large email stream is sufficient to detect suspicious or otherwise new email viruses that may be undetected by standard signature-based scanners. Therefore, relatively few MET installations would be necessary to gather sufficient data in order to provide broad protection services. Small scale simulations are presented to demonstrate MET in operation and suggests how detection of new virus propagations via flow statistics can be automated.

Keywords

email attachment, virus detection, anti-virus, email tracking

1. INTRODUCTION

Computer systems are constantly under attack by malicious software attached to email. According to NUA Research, email is responsible for the spread of 80 percent of computer virus infections [8]. Various estimates place the cost of damage to computer systems by malicious email attachments in the range of 10 to 15 billion dollars last year alone. There are many commercial systems available which attempt to detect and prevent these attacks. The most popular approach to defend against malicious software is through anti-virus scanners such as Symantec and McAfee, as well as server-based filters that filters email with executable attachments or embedded macros in documents [6, 14].

New Security Paradigms Workshop '02, September 23-26, 2002, Virginia Beach, Virginia.
Copyright 2002 ACM ISBN 1-58113-598-X/02/0009 ...$5.00

These approaches have been successful in protecting computers against known malicious programs usually by employing signature-based methods. However, they have not yet provided a means of protecting against newly launched (unknown) viruses, nor do they assist in providing information that may help trace those individuals responsible for creating viruses. Only recently have there been approaches to detect new or unknown malicious software by analyzing the payload of an attachment. The methods used include heuristics [16], neural networks [5] and data mining techniques [11, 12]. However, these methods in general do not perform well enough to detect all malicious software. Other approaches recently appearing include email client wrappers that aim to detect violations of behavior based upon policy rules that specify legitimate behavior [1]. This approach relies on software being deployed at the client-side system receiving the attachment.

In recent years, not only have computer viruses increased dramatically in number and begun to appear in new and more complex forms, but the increased inter-connectivity of computers has exacerbated the problem by providing the means of fast viral propagation [15].

Since malicious software can not always be detected in advance by inspecting payload, we can reduce the damage caused by malicious software by monitoring its behavior in spreading among nodes in networks. After some (particularly unlucky) initial victim networks are infected, others may be forewarned by an impending threat, who then may take preventive action against that threat. Currently monitoring systems exist through organizations such as WildList [3], and the Trend Micro World Virus Tracking Center [7].

WildList is an organization consisting of 65 virus information professionals, who report all computer programs that they have received and positively identified as malicious. This list does not include those cases where an attachment is considered suspicious but not yet classified as malicious, or include any viruses not specifically reported by these 65 participants. This leaves computer systems vulnerable to attack from unreported viral incidents [3]. Since the process of reporting is not automated, malicious software (especially the self-replicating variety) can spread much faster than the warnings generated by WildList.

Trend depends on a proprietary virus scanner (Housecall) [2] which integrates with the Trend Virus Control System (their management solution for network administrators) to report information about actual virus infections. It attempts to predict virus outbreaks and prevent them pro-actively with the use of a dynamic map to analyze worldwide virus trends in real time [7]. However, since HouseCall is not widely used, Trend's data is incomplete. Furthermore, if Trend's database is not updated at the time that a virus infects a system, then the virus remains unreported.

In this paper, we present the Malicious Email Tracking (MET) system which addresses these problems. The key dia erence between MET and previous monitoring systems such as Trend is that MET extracts and logs a unique identifier froln Kattachments passing through a mail server. If an attachment is discovered to be malicious after the fact, the statistics on its behaviors will have been recorded and available for further analysis and reporting functions.

MET provides three major capabilities. The first capability is the ability to track the global spread of malicious software through email. We employ the epidemiological framework froln ephart et al., B 3 [4] to quantify and describe the spread of malicious emailsP y monitoring the spread of malicious software via email, we can assess the general threat that a specific malicious program is causing while it is occurring. This information can also help apprehend those responsible for creating the virus (up to the point of a well engineered spoof of I addresses at the launch point). The second capability is determining all of the points of entry via email of malicious software into a network. This can help the system administrators contain the damage caused by the software. The third capability is to reduce the spread of self-replicating viruses through email. otential self-replicating viruses can be detected by their traffic patterns at some mail servers and this information can be quickly propagated to other mail servers to block these viruses from being delivered to users.

This paper presents the concepts behind MET and demonstrates, via simulations of generated email for a small LAN, how it may operate in practice. Detailed studies concerning the analysis of email flows for detecting likely virus propagations (versus false alarms) is beyond the scope of this paper since the data needed to analyze these flows requires deployment of the MET system. However, the deployment is planned shortly and a detailed analysis of the actual flows will be the the subject matter of a future report.

2. MALICIOUS EMAIL TRACKING

MET has two primary components, the MET client and the MET server. The MET client component runs on mail servers, monitors and logs email traffic, and generates reports sent to the MET server. The MET server runs at a central location and receives these reports in order to generate statistics and alerts about malicious programs which are distributed back to the MET clients. The MET server can be operated by a trusted third party oa ering a service to networks running the MET client.

The MET system requires trusted communication between the MET client and the MET server. Standard techniques can be used to authenticate and secure the communications between the MET client and MET server.

MET provides an efficient system of storing and transferring the data it gathers, while also taking into account security and privacy issues. We now detail the choices that we made in designing the proof of concept system.

2.1 Unique Identifiers for Email Attachments

The key to tracking attachments in the MET system is the assignment of a unique identifier for each email attachment. The MET client extracts an attachment from an email and computes an identifier from the payload of the attachment. This unique identifier is used to aggregate information about the same attachment propagated in dia erent emails. We presume that payload will be replicated without change during virus propagation among spreading emails and thus tracking the email attachment via this identifier is thus possible.

The name of the attachment, or the subject of the email is clearly not sufficient information for tracking because one virus may be sent under several dia erent names and subject lines since these fields are easily alterable by the malicious software. MET computes the MD5 hash [10] of every binary attachment received to create the unique identifier, using the hexadecimal representation of the binary as input to the algorithm. We note that polymorphic viruses will have dia erent identifiers for each instance of the virus. We discuss possible methods to address this problem in the conclusion section.

2.2 MET Client

The MET client consists of several components. The core of the MET client is a database, which stores information about all email attachments that pass through the mail server. The MET system contains a component to integrate with the mail sever. In our prototype implementation, we integrated the MET client with sendmail [13] using procmail [ ]. The MET client also contains a component to compute the unique identifiers for attachments. A data analysis component extracts statistics from the database to report to the MET server and a communication component handles the communication between the MET client and the MET server. The MET architecture is graphically displayed in Figure 1.

When integrated with the mail server, the MET client processes all mail that contains attachments and computes a unique identifier for each attachment. The MET client stores a record containing the identifiers for each mail that contains an attachment in a databaseP y querying the database with a list of the identifiers for known malicious programs, the administrator can determine the points of entry of malicious emails into a network, and can maintain a list of the senders and recipients of these emails. Even if a logged attachment was not initially acknowledged as malicious but only later categorized to be so, the points of entry can still be recovered since a record of all attachments is stored in the database.

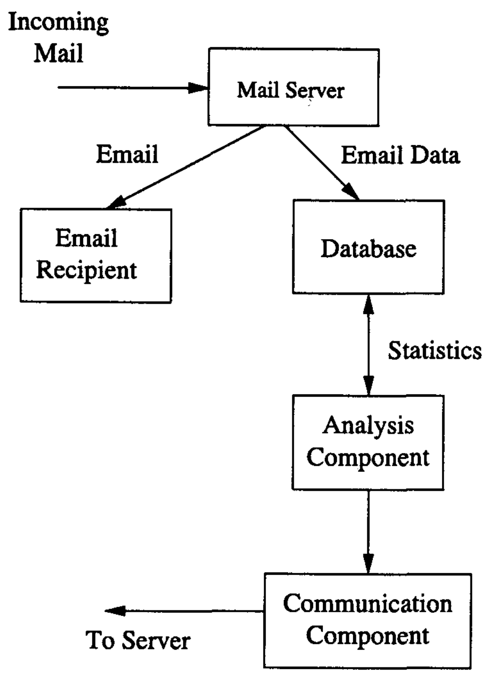

Figure 1: MET Client Architecture

While monitoring the flow of all attachments, MET allows the system administrator to distinguish between email traffic containing non-malicious email attachments and email traffic containing malicious software attachments. Malicious programs that self-replicate will likely propagate at a significantly diaerent rate than regular attachments sent within the environment in which MET is installed. These diaerences may become more apparent as all email is monitored, and (temporal) statistics are gathered carefully within that environment to establish norms for email flows.

Each MET client is required to keep the minimum amount of information concerning emails that contain attachments described in Table 1.

In addition, in the database we store the list of unique identifiers for known malicious attachments along with the names of these attachments. This list is typically obtained and updated from the MET server. Typically, an MET server would only be able to add identifiers for known malicious attachments. This will prevent a compromised MET server from sending updates that make MET clients vulnerable.

The MET system uses the information stored in the database in two ways. Since MET can determine the points of entry of a malicious attachment into a network, this can greatly assist the cleanup associated with an email virus incident and can help the system administrator to reduce and contain the associated damage.

In addition, the MET client gathers statistics about the propagation of each malicious attachment through the site which is shared with the MET server. This allows a global view of the propagation of malicious attachments and allows us to quantify the threat of these attachments as described

Email Attachment Log Record:
Unique ID of every Attachment
Time Stamp
Attachment Classification (Malicious or enign)
Sender Email
Receiver Email

Table 1: Information stored in MET Client Database for each email that contains an attachment

Record reporting malicious attachment incident
ID of reporting server
Unique ID of attachment
Date/Time of report
revalence
irth Rate

Table 2: Information store on a central repository

below. The core statistics that are reported for each malicious attachment is the prevalence of an attachment and the birth rate of an attachment. The prevalence is the number of times it was observed by the MET client and the birth rate is the average number of copies sent from the same user. oth of these statistics can be easily obtained from the database. In section 3 we show how we combine this information from multiple MET clients to quantify the threat level and various other statistics on a virus from this basic information.

Self-replicating viruses naturally have extremely high birth rates. If a MET client detects an attachment with a very high birth rate, the MET client can warn the MET server that this attachment is a potential self-replicating virus. The MET server can in turn warn other MET clients on the Internet about this attachment which can reduce the spread of these types of viruses. In section 4 we discuss the algorithms for determining when an attachment may correspond to a self-replicating virus.

2.3 MET Server

The MET server runs at a central location and communicates with the MET clients deployed at various mail servers. The MET server can typically be operated by a trusted third party and various networks can make agreements with this third party to provide the MET services.

The MET server has several functions. The MET server is responsible for propagating an updated list of unique identifiers associated with known malicious viruses to the MET clients. This propagation is automated which allows for rapid update of the MET clients immediately when a new malicious virus is discovered. In this model, the responsibility of updating the list is centralized and updates are not dependent on the responsiveness of individual system administrators.

The MET server is responsible for aggregating statistics that MET clients reports which allows MET to monitor viruses at a global level. The information contained in each record is shown in table 2. The fields all correspond to information that the central server needs to either query the local

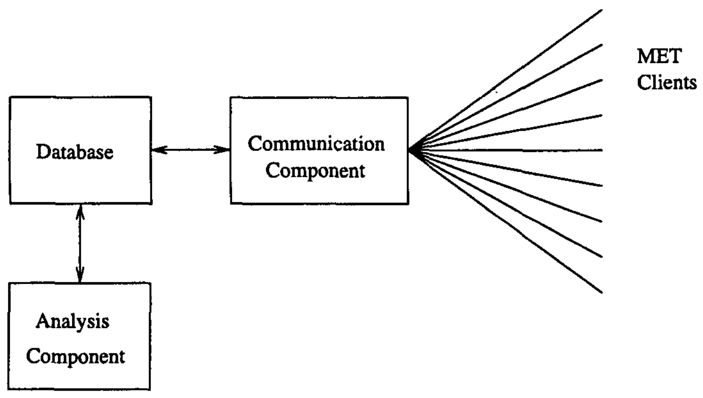

Figure 2: MET Server Architecture

host for more information, or to compute basic aggregate statistics. We point out that the type of information sent to the MET server are statistics that protect the privacy of individual users who may have sent or received the malicious attachment. There is essentially no information in the records which can identify an individual user account.

The core component of the MET server is a database of these records. MySQL is used in the MET prototype. The MET server also contains a data analysis component which performs the analysis over these records and a communication component which manages the communication with multiple MET clients. An architecture for the MET Server is shown in Figure 2.

When a local database reports an incident of a received malicious email attachment, it reports a unique incident identification number, the unique identifier of the attachment, the date and time of the attack, the prevalence and the birth rate. The prevalence is the number of users in the network that received this attachment, and the birth rate is the rate at which the virus is replicating on the local level.

In addition, the MET server can warn all MET clients about potential self-replicating virus threats when they are detected by a MET client. This process is fully described in section 4.

The communication between the MET server and the MET client consists of messages passed on a secured channel using encryption and authentication mechanisms.

3. DERIVED STATISTICAL INFORMATION ABOUT MALICIOUS EMAILS

A great deal of information can be derived from the statistics obtained from the MET clients that is reported to the MET server. We use the framework presented in ephart et al., 1 3 [4] to quantify the flows of malicious attachments. We can both quantify the flows of malicious attachments through a network and the global flows of the malicious attachments through the Internet. We compute the following metrics for each malicious attachment (others are certainly possible):

- Virus Incident: the fraction of the total number of machines within an organization infected by a particular virus, due to a single initial infection from outside the organization. Since each attachment is saved in the local repository with a Unique ID and malicious or benign classification, this value is simply the number of times each malicious unique ID appears in the local repository.
- irth rate: the rate at which a virus attempts to replicate from one machine to another. This value is calculated by determining the total number of email addresses an attachment is sent to per minute. If this value is set to a specific threshold, it can be used to determine whether or not a program is a self-replicating virus. Obviously, any time quanta can be implemented, and is best determined by observing local email behavior. (We presume that a malicious payload will not have access to these statistics in order to make its spread behavior appear normal within the environment.)
- Lifespan: the length of time a virus is active. This value is calculated by subtracting the first time a virus is seen from its last occurrence in the local repository. This values reports the amount of time a virus was free to cause damage to a network before it was detected.
- Incident rate: the rate at which virus incidents occur in a given population per unit time, normalized to the number of machines (computers) in the population. This is calculated by the central server based on the virus incident values reported by the local server.
- Death rate: the rate at which a virus is detected. This is calculated by the central repository by taking the average lifespan of the virus.
- revalence: a measure of the total number of local hosts infected by a particular virus. This value is calculated by the central repository by summing over the number of local hosts reporting the same virus.
- Threat: the measure of how much of a possible danger a virus may be. One straightforward way to measure threat is to calculate the incident rate of a virus added to the prevalence of a virus divided by the total number of participating local hosts and the total number of viruses.
- Spread: a measure of the global birth rate of a virus. This is calculated by taking the average of the birth rates reported by the participating hosts.

These metrics are directly implemented by computing SQL aggregates over the databases (both local and central).

Each time a MET client determines that an attachment is a virus, it sends a report to the MET server and the MET server updates its statistics for that virus.

3.1 Sampling to Estimate Global Malicious Email Prevalence

If a MET server obtains reports from all mail servers, then exact metrics for all malicious email attachments can certainly be gathered. However, in practice, a MET server will only obtain reports from a fraction of mail servers. This is because only a small portion of mail servers will deploy MET clients and each MET client does not necessarily report to the same MET server.

Even though the MET server only obtains reports on a fraction of the total malicious email attachments propagating through the Internet, the MET server can still compute accurate statistics for the malicious email attachments that it observes. We can view the limited set of mail servers as a representative sample of the Internet and extrapolate the statistics for malicious viruses accordingly. Since the Internet is extremely large, even with a small fraction ($\approx .1\%$) of the mail servers, we can accurately compute the statistics. Even at the local level, simple thresholds on email attachment propagation rates may detect new viruses. We demonstrate this with a simulation in the following sections.

4. SELF-REPLICATING MALICIOUS PROGRAMS

The third capability of the MET system is to reduce the spread of self-replicating viruses. This is capability is implemented in both the MET client and MET server. The basic idea is that if a MET client detects an attachment that seems to be self-replicating, it warns the MET server which in turn warns other MET clients. These clients then instruct their mail server not to deliver mails containing this attachment. Although the network that initially detected the self-replicating virus is likely infected by the virus, the warning it generates can both prevent other networks from being infected and significantly reduce the spread of the virus.

The MET prototype has built-in heuristics to determine whether or not an attachment is a self-replicating malicious program. The intuition behind these heuristics was derived by observing the behavior of a set of well-known self-replicating malicious programs. These heuristics are just a first approximation of detecting self-replicating viruses and potential improvements are discussed in the conclusion.

Many self-replicating viruses have a similar method of propagation – they send themselves to email addresses found on the victim's computer. This kind of behavior would manifest itself in an extremely high birth rate for the attachment. While in some cases a large birthrate for an attachment would be normal such as in a broadcast message, what is characteristic of self-replicating viruses is that the message comes from multiple users. In fact the number of users that send the message depends on the number of users who open the attachment.

Our heuristic for detecting self-replicating viruses is to classify an attachment as self-replicating if its birth rate is greater than some threshold t and the attachment is sent from at least l users. If an email flow record is above the threshold t, the MET client notifies the MET server with the unique ID of the attachment. The MET server propagates the unique ID to the MET clients which instruct the mail server to block all emails that contain an attachment with this unique ID. In practice, these mails can be queued until a system administrator can determine whether or not they are malicious.

The core capability provided by MET allows deeper analysis of local flows in order to set appropriate thresholds. Such studies will be reported in a future paper.

5. SIMULATIONS

A problem with analyzing simulated emails flows is that it is very difficult to generate data that is representative of email flows across multiple mail servers. Even if we were able to obtain large amounts of logs from these servers, since we are only interested in the emails that contain attachments, only a small portion of the logs would be applicable. However, typical email logs do not contain information on whether or not the message contains an attachment. Furthermore, it is impossible to determine which set of emails correspond to the same attachment.

Only with deployment of a system like MET, can we obtain real data to perform a performance analysis. A deployment of MET is planned for the near future and a future report will focus on an in depth analysis of email flows.

In the absence of real data, we generate a set of synthetic data to perform the analysis. We test MET with a simple threshold logic (bounds on birthrate and numbers of users) to demonstrate how it may operate on a local level.

5.1 Generated Data Sets

The simulated data was generated for 80 dia erent hosts, 500 dia erent email addresses, and 100 dia erent attachments. Ten of the attachments were classified as malicious.

We generate our synthetic data as follows. We generate a set of email records by picking a random sender and destination address. The attachment that is sent is a randomly selected attachment drawn from the set of attachments that the sender has previously received. If the sender address received no email attachments, then the attachment is picked at random from the global pool of 100 dia erent attachments.

We now show the results of this data, the various statistics computed and how self-replicating viruses were detected or escaped detection.

5.2 Metrics Calculation

Table 3 displays a portion of the generated email log. Using this log, for the virus with ID *Fxw4foiv8fugOwJIHqFenAki1Ui1E* which appeared two times in our dataset and spread to two other hosts, we can compute its metrics. In this case, the revalence for this virus would be 2, and the irthrate would be 0.06.

The metrics calculated for the malicious attachments labeled M in Table 4 are sent to the MET Server for further analysis. The attachments classified as benign ( ) are also further analyzed to detect unknown self-replicating viruses. If the attachment has a virus incident above threshold t, the senders, recipients and time stamps associated with that attachment are further evaluated. If the threshold t was set to 10, the data in table 4 requires that the logs corresponding to both benign attachments, *Zi5XtPiyk...* and *EpC0Gwnyi...*, be evaluated to check for self-replicating viruses.

md5sum	Sender Address	Recipient Address	date-time
Zi5Xt iykp...	toohot@pb.com	monica@columbia.edu	11:34:00, 1/17/02
EpC0Gwnyii...	bob@ccny.edu	helana@gls.com	11:34:00, 1/17/02
Qiqw7xyg0...	elvis@columbia.edu	allen@microsoft.com	11:34:00, 1/17/02
Qiqw7xyg0...	recruiting@db.com	boston@yahoogroups.com	11:34:00, 1/17/02
EpC0Gwnyii...	elvis@columbia.edu	ejcab@exchange.ml.com	11:34:00, 1/17/02
EpC0Gwnyii...	toohot@pb.com	monica@columbia.edu	11:34:00, 1/17/02
EpC0Gwnyii...	johedoe@sell-your-soul.com	bob@ccny.edu	11:34:00, 1/17/02
Fxw4foiv8f...	monica@columbia.edu	recruiting@db.com	11:34:00, 1/17/02
Qiqw7xyg0...	crown432@aol.com	bob@ccny.edu	11:34:00, 1/17/02
EpC0Gwnyii...	notice@freelotto.com	susan@verizon.com	11:34:00, 1/17/02
Fxw4foiv8f...	alewis@msn.net	kathylee@voicestream.com	11:34:00, 1/17/02

Table 3: Sample Synthetic Data

MD5	Virus Incident	Lifespan	irthrate	Class
Zi5Xt iyk...	11	35	0.31	
7Y5 mdEN2...	1	0	-1.00	
THySziDD1...	5	34	0.15	M
TnAcVv64j...	2	12	0.17	
YxO.3XSXf...	1	0	-1.00	M
EpC0Gwnyi...	32	45	0.71	
DCt37a1...	1	0	-1.00	
L CTlsMZe...	1	0	-1.00	M
LWL6q q l...	1	0	-1.00	
Fxw4foiv8...	2	33	0.06	M
ruuznFcUo...	4	36	0.11	
5j3eUV w...	2	4	0.50	M
EEHODdLIO...	1	0	-1.00	
O ZWOQd0...	1	0	-1.00	

Table 4: Metrics Calculated and Stored By Client

MD5	Incident Rate	Death Rate	revalence	Threat	Spread
THySziDD1...	5.5	31	2	0.1375	0.14
YxO.3XSXf...	1	0	1	0.0125	-1.00
L CTlsMZe...	1	0	1	0.0125	-1.00
Fxw4foiv8...	2.5	30.5	2	0.0625	0.065
HeHSXvmU ...	6.66	27	3	0.25	0.25
gsJHH.s.p...	2	33	1	0.0125	0.06
DHGYYAH4 ...	26	40	1	0.325	0.65
OiltoOr 3...	23	47	1	0.2875	0.4

Table 5: Metrics Calculated and MET Server Based on Table 6

The analysis suggests that both attachments are self replicating viruses. A report is then sent to the central server. The central server would then subsequently make the final decision as to whether or not the attachment is actually a self-replicating virus based on the number of other reports of the virus it has received from other clients.

The simulation shown here reports to the central server that *Zi5XtPiyk...* is a self-replicating virus.

Furthermore, the report in Table 6 can then be used to determine the death rate, and incident rate of this virus. All this information is sent back to the clients and can be used to prevent future infections.

Given further information from the host, MET is capable of providing even more insightful information. If sites send the cost of repair after attack, the total damage cost caused by a virus can also be calculated by MET. In addition, if email and I addresses are sent from clients to servers, the original sender I address can be tracked perhaps providing insight to the identity of the virus originator, or the unlucky initial victims at the launch point.

Host ID	MD5	Virus Incident	Lifespan	irthrate
137.23.1235	THySziDD1...	5	34	0.15
137.23.1235	YxO.3XSXf...	1	0	-1.00
137.23.1235	L CTlsMZe...	1	0	-1.00
137.23.1235	Fxw4foiv8...	2	33	0.06
137.23.1235	HeHSXvmU ...	7	24	0.25
765.12.4674	THySziDD1...	6	28	0.13
765.12.4674	HeHSXvmU ...	5	25	0.20
765.12.4674	gsJHH.s.p...	2	33	0.06
76.23.8 76	DHGYYAH4 ...	26	40	0.65
76.23.8 76	Fxw4foiv8...	3	28	0.07
76.23.8 76	HeHSXvmU ...	8	32	0.30
76.23.8 76	OiltoOr 3...	23	47	0.4

Table 6: Data Stored By MET Server

6. CONCLUSIONS

Even with the use of state of the art anti-virus software, malicious programs continue to cause damage to computer systems worldwide. Although complete eradication of malicious programs seems to be an impossible task, the more information we have on the propagation of these programs, the more effectively we can limit their damage. The Malicious Email Tracking system was designed to gather this information in conjunction with any anti-virus scanners, and across hosts while maintaining privacy and security polices. As patterns of viral propagation evolve and viruses mutate in an attempt to bypass new anti-virus software, MET will be able to monitor these changes and assess the need for improved software, minimizing repair costs.

As email behavior is observed and statistics are collected, basic metrics are calculated via SQL commands in a COTS DBMS and distributed. However, as the number of participants increases, the amount of data obtained increases. As a result there is a greater potential to calculate additional metrics and run further tests to learn new patterns of propagation. Additional fields over which to calculate more metrics can also easily be incorporated into the system. This data can be used to train new anti-virus programs and can further prevent the spread of new and unknown malicious programs.

The MET system relys on the users trust of the mail provider. The MET system collects and stores information that is already collected and stored by the mail server. Because of this, the MET system does not compromise the privacy of the users mail any more than the logs of the mail server. Since the MET server is allowed to make only aggregate queries, it is difficult to obtain information about individual users. The set of allowable querries can be restricted to make it very difficult to obtain information about individual users. Furthermore, the MET server is used only by a trusted party.

The topology and framework of MET allows for the tracking of any email attachment traveling through the Internet, not simply viruses. Simple modifications to the type of data saved and the metrics calculated can make MET a tool used for various research purposes.

The MET system we presented is an experimental prototype. There are many directions for future research to improve various aspects of the system. For example, instead of using unique identifiers based on MD5 hashes, we can use a different type of identifier that is robust for polymorphic viruses. In general polymorphic viruses make relatively minor changes between generations of the virus. An effective identifier would map all instances of a polymorphic virus to the same identifier. This problem is non-trivial and is worth further research. In addition, an identifier that covers several similar binaries would allow us to save space and allow us to track similar infections and thus give the system more flexibility.

Another future direction is to incorporate a more sophisticated model of self-replicating viruses. We can approach the problem from a probabilistic framework and compute a probability for each attachment of how likely it is self-replicating. One advantage to this approach is that we can quantify the risk associated with allowing the delivery of the attachment. This can allow administrators of individual MET clients to base their decision on whether they should allow the emails to be delivered based on this risk assessment.

The simulations presented in this paper were based on synthetic data. A more robust analysis of the performance of the system would require more realistic data for testing. We are in process of collecting email data from users on a single domain. All the email data collected is first hashed, so only statistical data is donated. The framework for this is under the EMT (Email Mining Toolkit) project currently underway.

In addition we are planning on deploying the MET system in the near future and presenting an in depth performance analysis of MET and analysis of email flows in a future report.

7. ADDITIONAL AUTHORS

Additional authors: Salvatore J. Stolfo (Department of Computer Science, Columbia University, email:`sal@cs.columbia.edu`).

8. REFERENCES

[1] B. Balzer. Assuring the safety of opening email attachments. In *Proceedings of DARPA Information Survivabilty Conference and Exposition II (DISCEX II)*, 2001.

[2] HouseCall. Free online virus scan. Online Publication, 2002. http://housecall.antivirus.com.

[3] W. O. International. Pc viruses in the wild. http://www.bocklabs.wisc.edu/ janda/wildlist.html.

[4] D. M. C. Jeffrey O. Kephart and S. R. White. Computers and epidemiology. *IEEE Spectrum.* http://www.research.ibm.com/antivirus/ SciPapers/Kephart/Spectrum/Spectrum.html.

[5] J. O. Kephart. A biologically inspired immune system for computers. *Artificial Life IV, Proceedings of the Fourth International Workshop on Synthesis and Simulatoin of Living Systems, Rodney A. Brooks and Pattie Maes, eds.*, pages 130–193, 1994.

[6] MacAfee. Macafee home page. Online Publication, 2002. http://www.mcafee.com.

[7] T. Microsystems. Trend world virus tracking center map. http://wtc.trendmicro.com/wtc/.

[8] Postini. Postini press release. Online Publication, 2000. http://www.postini.com/company/pr/pr100200.html.

[9] Procmail. Procmail home page. Online Publication, 2002. http://www.procmail.org.

[10] R. L. Rivest. The md5 message digest algorithm. http://www.ietf.org/rfc/rfc1321.txt.

[11] M. G. Schultz, E. Eskin, S. J. Stolfo, E. Zadok, and M. Bhattacharyya. Mef: Malicious email filter - a unix mail filter that detects malicious windows executables. http://www.cs.columbia.edu/ids/mef/rel_papers.html.

[12] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo. Data mining methods for detection of new malicious executables. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2001.

[13] Sendmail. Sendmail home page. Online Publication, 2002. http://www.sendmail.org.

[14] Symantec. Symantec worldwide home page. Online Publication, 2002. http://www.symantec.com/product/.

[15] C. Wang, J. Knight, and M. Elder. On computer viral infection and the effect of immunization. In *Proceedings of the 16th ACM Annual Computer Applications Conference*, 2000.

[16] S. R. White. Open problems in computer virus research. *Online publication.* http://www.research.ibm.com/antivirus/SciPapers/White/Problems/Problems.html.