ISSN (ONLINE):2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.25, February 2017

Detecting the Spreading Patterns of Mobile Phone Virus from Proximity Based Model Networks

Saraswathy.A[1], Reshma.K[2], Rama Vaishali.P[3]
[1, 2, 3] UG Scholar,
Department of Computer Science and Engineering.
Rajalakshmi Engineering College, Chennai India.

***Abstract*—as malware attacks become more frequently in mobile networks, it defects our systems. In terms of operating system, the mobile devices are heterogeneous. The malware infects the targeted system in any way i.e. local or global connectivity. The Next Generation of Anti-Virus (Advanced version of Pretty Good Privacy (PGP)) is a technique which relies on machine learning to analyze new viruses in an automated way. It uses some changing analysis approach to detect viruses, instead of relying on analysis based on previously captured samples of viruses. As a result, they are so much effective in detecting new viruses. Several documents can be send through mobile to mobile networks. The attackers inject the document then send to several users. In existing, once the file is download then only the users knows the system is affected through this document. But we proposed using some alerts to overcome this existing technique. While during download the file the alert will display three options (allow, deny and scan). Scan is used to scan the document. If malware is present it shows to the user. After scanning we can clear the virus then allow or deny the download of the document by the user choices. The main objective of the project is network-based detection systems detect worms primarily by monitoring, collecting, and analyzing the scan traffic (messages to identify vulnerable computers) generated by worm attacks.**

Keywords—heterogeneous mobile networks; network based detection; pretty good privacy; vulnerable.

INTRODUCTION

Computer viruses and worms have been studied for a longtime both by research and application communities. Viruses spread mainly through the exchange of floppy disks. At that time, only a small number of computer viruses existed, and virus infection was usually restricted to a local area. As computer networks and the Internet became more popular from the late 1980s, viruses and worms quickly evolved the ability to spread by various means such as file downloading, email, exploiting security holes in software, etc. Currently, mobile malware can propagate through two different dominant approaches. Via MMS, a malware may send a copy of itself to all devices whose numbers are found in the address book of the infected handset. The other approach is to use the short-range wireless media such as Bluetooth to infect the devices in proximity as proximity malware has investigated the proximity malware propagation features, and finds that it spreads slowly because of the human mobility, which offers ample opportunities to Deploy the defense system.

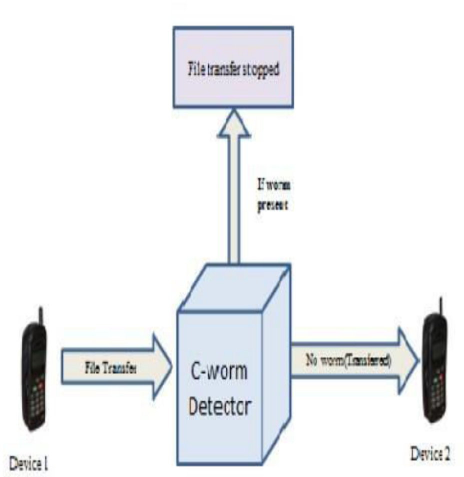


Fig.1. Illustration of existing Malware spreading model.

Challenges on existing system is that, Typically we cannot rely on centralized algorithms to distribute the signatures because the service infrastructure is not always available .Devices in general have limited resources, i.e., CPU, storage, and power.

Although their storage and CPU capacity has been increasing rapidly, it is still very resource-limited compared with Devices. If worm were found out and cleared user might not know about the source node which sent the worm file. Source node which sent the worm file.

However, the approach for efficiently deploying such a system is still an ongoing research problem.

RELATED WORK

First, the bulk of simulation studies involve topological worms and mobile viruses which refers to the existing literature for models of worms that spread via email, P2P and IM networks. The traditional models of malicious agent propagation do not capture several unique properties of a mobile enterprise network. Service interactions among the nodes at different time-scales create different vulnerable service topologies, rather than an average degree of connectivity among the nodes, as assumed by many epidemiological models. Mobile users with laptops, PDAs and cell phones not only contribute to these time-varying service topologies, but also introduce new vulnerabilities as well, e.g., Amber-type viruses that can spread via SMS/MMS messages and Bluetooth connections. Further, today's enterprise networks consist of diverse network segments with different levels of bandwidth, services and latencies. All of these factors affect the growth rate of an epidemic. Our proximity-based network modeling framework captures these factors by using traffic traces collected from enterprise networks. Using this framework, an enterprise can perform a realistic vulnerability assessment of its popular services, such as SMS, Bluetooth, email, P2P and IM. These services are often targeted by virus writers and increasingly, new malware are designed by the malware writers to exploit multiple of these services simultaneously. As a result, the current count of known mobile malware stands at 100, up from only 10 in previous years combined. Maintaining the Integrity of the Specifications.

The majority of the published studies on modeling and containment of malware have focused on scanning and email worms due to their prevalence and several successful large scale attacks on the Internet. On the contrary, there appears to be very little published work on mobile viruses and worms. This is the primary motivation of our work. We study the mobile viruses discovered to date and the various target discovery, infection and replication mechanisms they deploy to spread to other devices. We then model the propagation of a mobile virus similar to Commwarrior that targets Bluetooth and SMS/MMS services, using an agent-based malware simulator that we develop for studying mobile worms. This paper makes two primary contributions. First, it provides a detailed analysis of vulnerability from mobile viruses. Second, it proposes a new emulation framework for detecting the spreading patterns of mobile phone virus from proximity based model networks. The paper is organized as follows. In Section III, we provide an overview of mobile viruses and vulnerabilities. System descriptions related to our paper that is bloom filter, REA monitoring, network monitoring are discussed in Sections IV respectively. Next, we discuss briefly our proposed proximity -based model networks framework in Section V. Section VI presents the conclusion and future works of our simulation studies. Finally, we make reference in Section VII.

BACKGROUND STUDY

Study on mobile malware

The earliest versions of malicious codes are harmless and they didn‘t spread from device to device. The recent malicious malwares are capable of spreading to nearby devices via Bluetooth and pose serious threats on enterprise networks. The main difference between Blue tooth worm and the Internet worm is the mechanism adapted by the worms to infect the devices. The Bluetooth worm uses proximity scanning process to infect the nearby devices. Note that proximity scanning requires physical proximity (e.g., up to 10 meters for Bluetooth Class-2 devices) between an infected device and a target device, whereas SMS or MMS requires only a network connection between an infected device and the service gateway for sending messages and malicious payload to other devices. Similar to email viruses, the mobile viruses use social engineering techniques to entice unsuspecting users to click on infected audio, video or picture attachments. Caber & Commwarrior are the most popular worms that affects Bluetooth enabled devices.

•**Caber** replicates over Bluetooth connection and install its payload as Symbian System Installation file (SSI).It drains the power of infected phones and starts scanning for next Bluetooth devices for infection.

•**Commwarrior** spreads through the messages sent, in which the payload is attached. When it reaches a MMS enabled phones it randomly choose a phone number from the device address book and resets the infected device on the 1st hour of 14th of any month. After infecting a phone, it searches for nearby Bluetooth enabled devices for sending infected files.

•**Skulls** (2005) is a Trojan that propagates by sending both SMS and MMS messages, and overwrites many default phone applications such as the address book, and e-mail viewer and to-do lists. Many variants of Skulls have been observed in the wild.

•The **Red browser Trojan** (2006) is the first malware targeting J2ME (Java 2 Mobile Edition) phones and represents a major evolution in mobile viruses. Instead of focusing on high-end smart phones running on Symbian or Pocket PC, it works on many low-end phones with J2ME support. Red browser pretends to be a WAP browser offering free WAP browsing and SMS messages — the purpose is to use a social engineering technique to fool the user into sending SMS messages. However, it actually sends a flood of SMS messages to a specific number and therefore, can cause financial damage to the user.

•Multiplex (2005) is a Trojan that propagates by sending both SMS and MMS messages, and overwrites many default phone applications such as the address book, and e-mail viewer and to-do lists. Many variants of Skulls have been observed in the wild.

Malware Detection Methods

There are many types of techniques are available to detect mobile malwares. There are some common methods to find the malwares

•**Signature-based Detection,** in this the code is matched with all the malware codes in the database of the antivirus. Most of the commercial antivirus companies used a signature based detection methods to identify malwares. Signature is the binary of pattern of the machine code of a particular virus. It checks the content of the file dictionary of malware signatures. This method needs the huge database to store the malware signatures. It fails to identify an unknown malwares because a new malware may not contain a known signature of malwares.

•**Heuristic-based Detection,** in this case if a large fraction of the code matches with a malware code in database.

•**Behavioral-based Detection,** if a program has started to behave maliciously i.e. (started to copy itself uncontrollably, or deleting files without users permission etc Signature based detection method fails to detect a new malwares. Due to that problem, Behavior based detection mechanism was proposed to identify the new malwares with the help of machine learning algorithm named Support Vector Machine (SVM). Behavior based detection monitors the behavior of an application and compares the malicious and/or normal behavior profiles to detect the malwares

•**Sandbox detection,** Run it in a virtual environment to check how it functions (this method is slow though). In this “sandbox,” the execution is observed and malicious behavior can be detected. Once the malicious behavior is detected, signatures can be created so the next time that type of attack is seen it can be blocked by a more traditional Intrusion Prevention System.

•**Data-mining techniques**: - Data-Mining, Machine learning algorithms are employed to classify whether a program is malicious or not (behaviour detection). The goal of the data mining process is extracting to extract information from a data set and transform it into an understandable structure for further use. Data mining methods are used to detect the patterns in large amount of data and to detect the future instances in similar data with the help of these patterns. Naive bayes algorithm is used to classify the malicious code (MC) and begin code (BC) and detect new malwares.

SYSTEM DESCRIPTION

Bloom filter

Network Intrusion Detection System (NIDS) is needed to protect the end user machines from threats. An effective NIDS is therefore a network security system capable of protecting the end user machines well before a threat affects. NIDS requires a space efficient data base for detection of threats in high speed conditions. It acts as hardware antivirus device and connected with the CPU to remove the malicious input data. It consists of a set of hash functions, a hash function buffer to store hash results temporarily, a look up array to signify hash values and a decision component made of an AND to test the membership of testing string as shown in Fig.2.

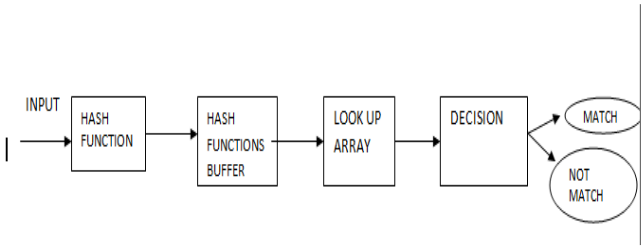


Fig.2. Block Diagram

```
Data    :  x is the object key to insert into the Bloom filter. Function: insert(x) for j: 1 . . . k do
Loop all hash functions k i? He (x); if Bi == 0 then
Bloom filter had zero bit at position i Bi? 1;
End
```

Algorithm 1: Pseudo code for Bloom filter insertion

```
Data: x is the object key for which membership is tested. Function: is member(x) returns true or false to the membership
Test m? 1; j? 1;
While m == 1 and j? K do i? He (x); if Bi == 0 then m? 0;
End j? j + 1; end
Return m;
```

Algorithm 2: Pseudo code for Bloom member test

Counting bloom filter

When a counter changes from 0 to 1, the corresponding bit in the bit-vector is cleared. It is important to note that the counters are changed only during addition and deletion of strings in a Bloom filter. For applications like network intrusion detection, these updates are relatively less frequent than the actual query process itself. Architecture of counting bloom filter is shown in fig. 3

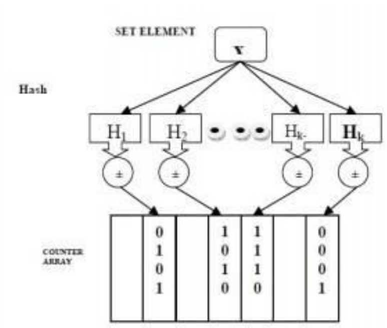


Fig.3: Counting Bloom Filter

Boyer–Moore String Search Algorithm

The Boyer–Moore string search algorithm is an efficient string searching algorithm that is the standard benchmark for practical string search literature. The algorithm scans the characters of the pattern from right to left beginning with the rightmost one. In case of a mismatch (or a complete match of the whole pattern) it uses two recomputed functions to shift the window to the right. These two shift functions are called the good-suffix shift (also called matching shift and the bad- character shift (also called the occurrence shift).

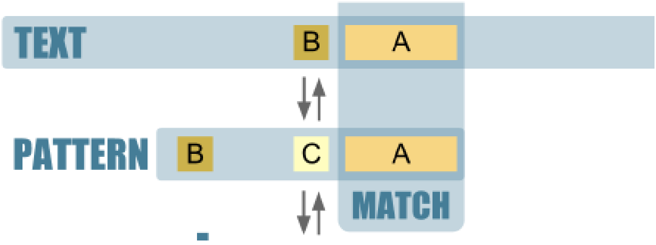


Fig.4: Working of Boyer Moore Algorithm

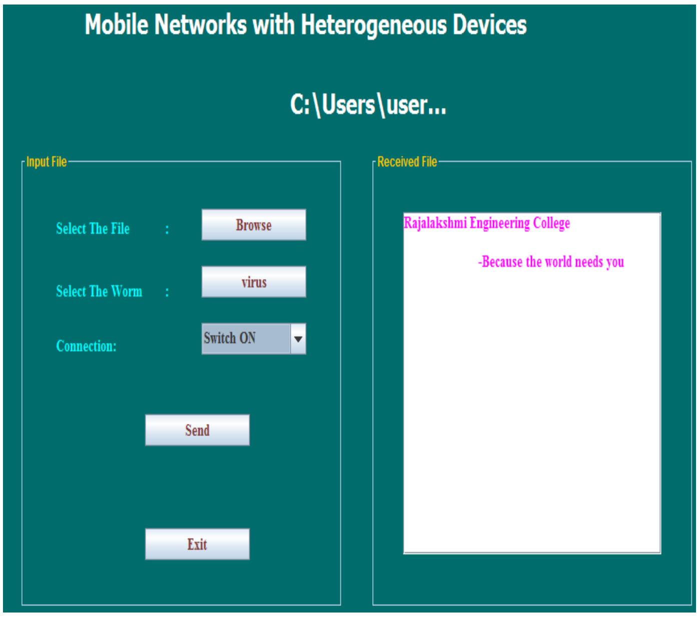
Algorithm Batch (T, P):

```
Input: String T (text with n characters and P (pattern) with m characters
Output: Starting index of first substring of T matching p, or an indication that P is not a substring of T
i <-- m - 1
j <-- m - 1
Repeat
If P[j] = T[i] then if j = 0 then
Return i {a match!}
Else {check next character} i <-- i - 1
j <-- j - 1
Else {P[j] <> T[i] move the pattern}
i <-- i + m - j - 1 {reset i to where it began in most-recent test}
i <-- i + max (j - last (T[i]), match (j)) {shift P relative to T}
{Note that even if j-last (T[i]) is negative, we will still perform appositive shift, because match (j) is always at least 1.}
j <-- m-1
Until i > n - 1
Return "There is no substring of T matching P."
```

Algorithm 3: Pseudo code for Boyer-Moore Algorithm

PROPOSED WORK

In the existing system, only once the file is downloaded, the users knows that the system is affected by virus through this document which was transferred. But in the proposed system we use some alerts to overcome this existing technique. During the download of the file the scanner alert will display three options (i.e. allow, deny and scan). Scan is used to scan the document beforehand in the sender's system itself before sending the document. If malware is found, it is then alerted to the user for the father actions that is supposed to be taken on the infected file. The virus detected file can then be either denied or then accepted without the malicious code. The type of virus is then detected using the bloom filter in case of any virus signature is found and is then removed using VERV algorithm. We cannot rely on existing algorithms to detect the virus because of the evolving nature of virus and are thus they are not effective enough in near-duplicate detection. There are several applications for scanning the files once the user has downloaded the file, but the virus must have already started to spread once the infected file is already downloaded into the device. If worms were found out and cleared, the user might not know about the source node which sent the worm file.



Mobile Networks with Heterogeneous Devices

C:\Users\user...

Select The File :

Select The Worm : virus

Connection: Switch ON

Received File:

Rajalakshmi Engineering College

-Because the world needs you

Fig. 5. Screenshot of sending virus file

The file transfers from sender to receiver by using Socket connection (port number).Thread is used to make the request and response simultaneously and to avoid the queue in processing request while handling multiple user. While during download the file the alert will display three options (i.e. allow, deny and scan).During Scan the file is first passed through bloom filter for content analysis to check the whether the file is infected or not infected. The bloom filter algorithm is chosen as it consists of a set of hash functions, a hash function buffer to store hash results temporarily, a look up array to signify hash values.

Bloom filter matches those collection of data in file with the virus pattern. If no match found file is downloaded in user's device or else if the file is infected, the Counting bloom filter update count in database. After update, the file is sent for scanning .In this step the Scanning is done using Boyer Moore Algorithm. This algorithm preprocesses the string being searched for (the pattern), in the file but not the string being searched in (the text). It is thus well-suited for applications in which the pattern either is much shorter than the text or does persist across multiple searches .Moore Algorithm first checks the virus update count and remove those virus one by one .In this stage, alert option displays whether to start download and now when the user starts downloading a normal file is downloaded.

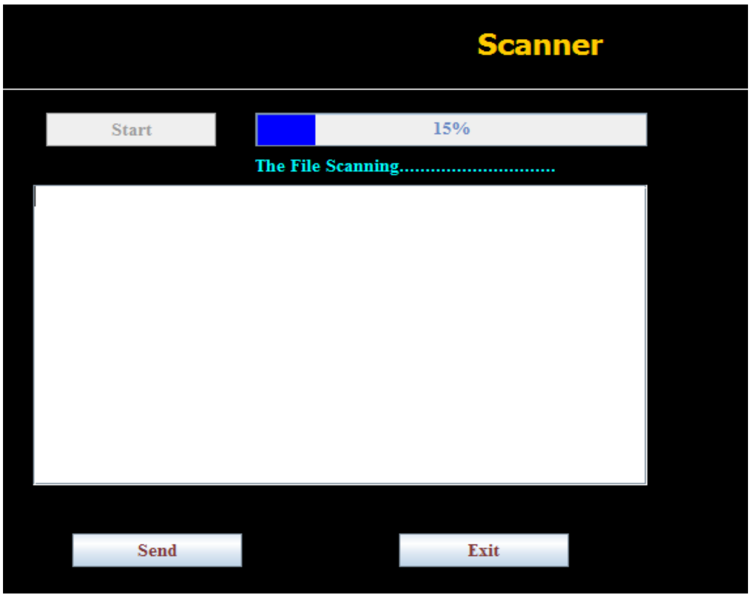
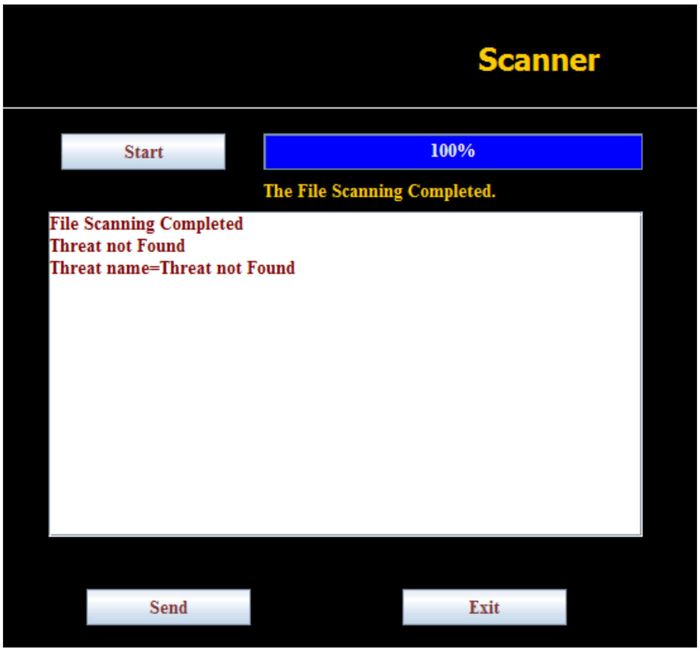


Fig.6: Screenshot of File Scanning



```
File Scanning Completed
Threat not Found
Threat name=Threat not Found
```

Fig.7. Screenshot after Removing Thread from File

CONCLUSION

In this paper, we have proposed a novel model to stop the propagation of modern malware. This model is able to address two critical processes unsolved in previous models: the scanning of the infected document is not done in the user’s device while leads to faster propagation of virus into the system without the user's knowledge and the virus detection scanner result also displays the source from which the virus had entered the mobile device. For the user can immediately stop the transfer of files with the infected node. For the future work, there are also some problems needed to be solved, such as the independent assumption of virus based on the near duplication signatures of the virus in the infected files or devices. The common files which is been used by the user is keenly observed and the virus signature created is similar to the file that is accessed by the user. In near future the independent assumption of virus based on near-duplication signature can be detected.

REFERENCES

[1] Gregory R Ganger, Gregg Economic, and Stanley MBielski. Self-securing network interfaces.

[2] A. Gupta and R. Sekar. An approach for detecting.

[3] Matthew M Williamson. Design, implementation and test of virus.

[4] Cynthia Wong, Chennai Wang, Dawn Song, Stanley M [5] Matthew M Williamson. Throttling viruses.

[5] Cliff Changchun Zou, Weibo Gong, and Don Towsley.

[6] Shigang Chen and Yong Tang. Slowing down internet worms.