ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Analyze and Find Out the Behavior of Malwares in Delay Tolerant Networks

G.Sathish Kumar, Dr.G.Gunasekaran

Research Scholar, Sathyabama University, Chennai, India

Principal, Meenakshi College of Engineering, Chennai, India

ABSTRACT: "Potentially Unwanted Programs (PUP)" is a term which explores the possibility to attack the privacy and security issues of a system. The unwanted programs like virus, Spy ware, adware, Trojan, worms. These kinds of programs might be compromising the discretion, honesty and the presence of the system otherwise they can get hold of the responsive in sequence about the system without the knowledge of the system user. Initially all the system users concentrated on the well-known area of spreading infectionary things called *virus*. And the most recent proportion the researchers analyze the greatest defense breaker called spywares. Spyware may be competent of imprisoning keystrokes, captivating screenshots, economy authentication recommendations, amassing individual email addresses and web form information, and thus may get hold of behavioral and special in sequence concerning users. A new system is required to analyzing the deficiency cause of spywares and breaks it from an affected system. In the end Seismic released their anti-Spyware software to counter all problems that they themselves had created and earned more money than what had been earned previously by spreading the Spyware. In the closing stages Seismic at large their anti-Spyware software to oppose all troubles that they themselves had shaped and be paid more money than what had been earned beforehand by dispersal the Spyware.

KEYWORDS: Malwares, Spyware, DTN-Delay Tolerant Network, Malware Characterization, Proximity Malware.

I. INTRODUCTION

The reputation of mobile shopper electronics [8], similar to laptop computers, PDAs, and additional lately as well as highly, smart phones [9], revitalizes the Delay Tolerant Network (DTN) replica [3][4] as an substitute to the conventional communications mock-up. The prevalent implementation of these strategies, attached with physically powerful financial inducements, encourages a group of malware that purposely objectives DTNs [2]. We describe this division of malware proximity malware. An near the beginning example of proximity malware is the Symbianbased Cabir worm [14], which broadcasted as a Symbian Software setting up Script (.sis) box up through the Bluetooth link sandwiched between two spatially neighbouring strategies.

Afterwards instances is the iOS-based Ikee worm, which browbeaten the non-payment SSH password on jailbroken iPhones to proliferate through IP-based Wi-Fi [3][4] connections. Preceding investigators enumerate the hazard of nearness malware assault and make obvious the likelihood of initiation such a show aggression, which is long-established by fresh tales on take controlling hotel Wi-Fi hotspots for force by malware attacks [3][4]. With the acceptance of new-fangled petite variety announcement knowledge such as NFC and Wi-Fi Direct that make possible unprompted bulk data transfer flanked by spatially adjacent mobile devices [8], the hazard of closeness malware [1] is flattering supplementary pragmatic and germane than increasingly. Immediacy malware pedestal on the DTN representation [15] conveys only one of its kind refuge confronts that are not at hand in the communications replica.

In the communications mock-up, the cellular transporter mildly checks networks for malformations [15]; furthermore, the reserve shortage of creature nodes restrictions the speed of malware proliferation. For case in point, the putting in place wrap up in Cabir and the SSH meeting in Ikee, which were used for malware proliferation, cannot be become aware of the cellular transporter. On the other hand, such middle keeping an eye on and supply restrictions are not present in the DTN replica. Closeness malware makes use of the opportunistic connections and dispersed natural world of DTNs for proliferation.

DOI: 10.15680/IJIRCCE.2016. 0404281

A. ACTIVITY DIAGRAM

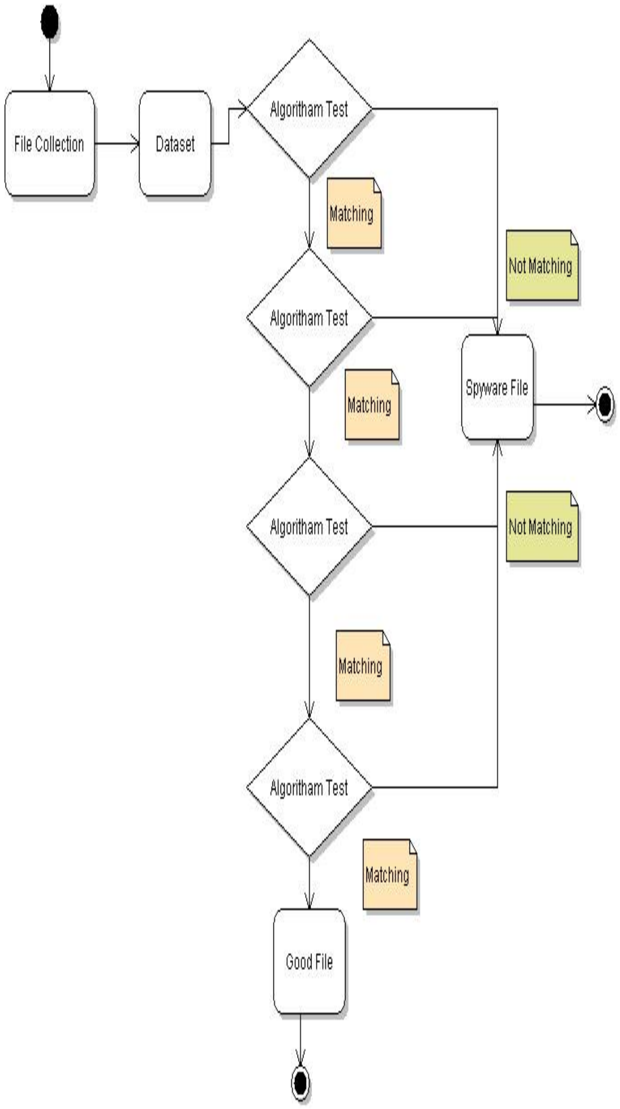


Fig.1: Activity Diagram

An advance planning to fight opposed to proximity malware is to find it [5]. In this paper, we think of a common behavioral characterization of proximity malware. Behavioral characterization, in terms of system call and program flow, has been previously introduced as an energetic replacements to model comparing for malware identification. In our replica, malware affected nodes behaviors are analyzed by different parties at the time of their many probabilistic findings [7]: Lonely analyses may be not in perfect criteria, but abnormal behaviors of affected nodes are fortifiable in the long run. For example, a single suspicious Bluetooth connection or SSH session request during one encounter does not confirm a Cabir or Ikee infection, but repetitive suspicious requests spanning multiple encounters is a strong indication for malware infection. The imperfection of a single, local observation was previously in the context of distributed IDS against slowly propagating worms.

II. DATASET GENERATION

Two ARFF databases based on frequency and common features were generated. All input attributes in the data set are represented by Booleans. These ranges are represented by either 1 or 0. The classification process is as follows: A Naive Bayes classifier [10] is a probabilistic classifier based on Bayes theorem with independence assumptions, i.e., the different features in the data set are assumed not to be dependent of each other. This of course, is seldom true for real-life applications. Nevertheless, the algorithm has shown good performance for a wide variety of complex problems. J48 is a decision tree-based learning algorithm. During classification, it adopts a top-down approach and traverses a tree for classification of any instance. Moreover, Random Forest is an ensemble learner. In this ensemble, a collection of decision trees are generated to obtain a model that may give better predictions than a single decision tree.

Dataset Generation - Rule and Practical Implementation:

```
protected String getRelationNameToUse()
{
String result;
result = getRelationName();
if (result.length() == 0)
result = defaultRelationName();
return result;
}
*. checks, whether the given option is in the blacklist of options not to be output by makeOptionString
*. @param option      the option to check
*. @return true if the option is on the blacklist
*. @#makeOptionString(DataGenerator)
*. Removes all the options from the options array that are blacklisted
@param options the options to remove from the blacklist
@return the processed options array
protected String[] removeBlacklist(String[] options)
{
    Enumeration     enm;
    Hashtable       pool;
    Option          option;
    // retrieve options that are on blacklist
    enm  = listOptions();
    pool = new Hashtable();
    while (enm.hasMoreElements()) {
      option = (Option) enm.nextElement();
      if (isOnBlacklist(option.name()))
        pool.put(option.name(), option);
    }
    // remove options
    enm = pool.keys();
    while (enm.hasMoreElements()) {
      option = (Option) pool.get(enm.nextElement());
      try {
        if (option.numArguments() == 0)
          Utils.getFlag(option.name(), options);
        else
          Utils.getOption(option.name(), options);
      }
      catch (Exception e) {
        e.printStackTrace();
      }
    }
    return options;
  }
```

A. Bayes Classifier

A Bayes classifier is a straightforward probabilistic classifier based on be relevant Bayes theorem (from Bayesian statistics [13]) with strapping (naive) sovereignty suppositions. An additional expressive expression for the fundamental likelihood replica would be "Independent Feature Model".

In uncomplicated conditions, a naive Bayes classifier assumes that the occurrence (or nonexistence) of a meticulous characteristic of a division is unconnected to the occurrence (or nonexistence) of any supplementary attribute. For illustration, a fruit may be measured to be an apple if it is red, encompassing, and about 4" in diameter. Constant if these features depend on each other or upon the subsistence of the other features, a naive Bayes classifier [11][12] mull over all of these belongings to autonomously supply to the likelihood that this fruit is an apple.

Depending on the accurate environment of the likelihood replica, naive Bayes classifiers can be taught very professionally in administer erudition surroundings. In numerous sensible submissions, parameter judgment for naive Bayes replicas employs the process of highest probability; in other words, one can work with the naive Bayes model without considering in Bayesian likelihood or by means of any Bayesian techniques [11].

Bayes in Implementation:

```
public void setOptions(String[] options) throws Exception {
    String       tmpStr;
    Vector       list;
    super.setOptions(options);
    list = new Vector();
    list.add("-N");
    list.add("" + getNumAttributes());
    list.add("-M");
    list.add("" + getNumExamples());
    list.add("-S");
    list.add("" + getSeed());
    list.add("-A");
    tmpStr = Utils.getOption('A', options);
    if (tmpStr.length() != 0)
list.add(tmpStr);
else
list.add("" + defaultNumArcs());
list.add("-C");
tmpStr = Utils.getOption('C', options);
if (tmpStr.length() != 0)
list.add(tmpStr);
else
list.add("" + defaultCardinality());
setGeneratorOptions(list);
}
protected static String makeOptionString(DataGenerator generator)
{
StringBuffer result;
Enumeration enm;
Option option;
result = new StringBuffer();
result.append("\nData Generator options:\n\n");
enm = generator.listOptions();
while (enm.hasMoreElements()) {
option = (Option) enm.nextElement();
// skip option if on blacklist
if (isOnBlacklist(option.name()))
continue;
result.append(option.synopsis() + "\n" + option.description() + "\n");
}
return result.toString();
}
```

In malice of their naive intend and it seems that over cut down suppositions, naive Bayes classifiers encompass employment moderately well in many multifaceted real world circumstances. In 2004, psychoanalysis of the Bayesian classification predicament has shown that there are some notional reasons for it become visible that difficult to deal with effectiveness of naive Bayes classifiers. At a languish, a all-inclusive association with supplementary cataloguing schemes in 2006 give you an ideas about that Bayes arrangement is outperformed by supplementary up to date draw near, such as accidental forests.

An improvement of the naive Bayes classifier is that it necessitates a minute quantity of preparation information to approximation the strictures (means and variances of the variables) essential for organization. For the reason that self-governing variables are unspecified, only the variances of the variables for each class require to be strong-minded and not the complete covariance surrounding substance.

B. Effects of Malware

- Malware causes your connection to slow down
- Malware badly written code can cause your computer to crash.
- Malware can cause your computer to display error messages continually.
- Malware could cause your computer to be incapable of shutting down or restarting as it keeps certain processes active.
- Malware could be used for identity theft to gather personal information or data from your computer.
- Malware can hijack your browser to redirect you to sites for its purposes.
- Malware can infect your computer and use it as a server to broadcast various files or attacks.
- Malware can send spam through and to your inbox.
- Malware could send emails you did not write getting you or your company in trouble.
- Malware can infect your computer giving an attacker control of your system and your resources, like your connection.
- Malware can cause new and unexpected toolbars to appear.

C. Malware and its Variations:

Malware is a pervasive problem in distributed computer and network systems. Malware includes computer viruses, worms, trojan horses, ransomware, spyware, adware, scareware, and other malicious programs [14].

A malware can be classified into various numbers of variations. They are as follows:

The first variation of malware is an Overwriting Malware. It usually starts with spreading the malware into the system and affects the targeting portion of the respective system and spread the malwares within it.

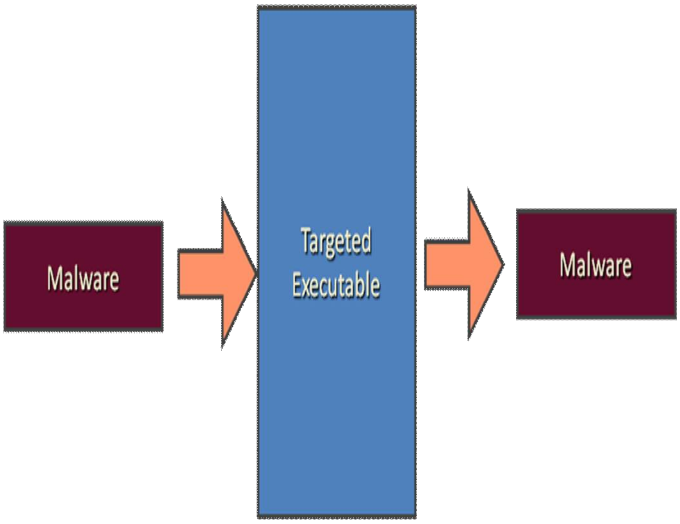


Fig.1.A. Overwriting Malware

The second variation of the malware is a Pre-pending malware. The following system model clearly shows the variation:

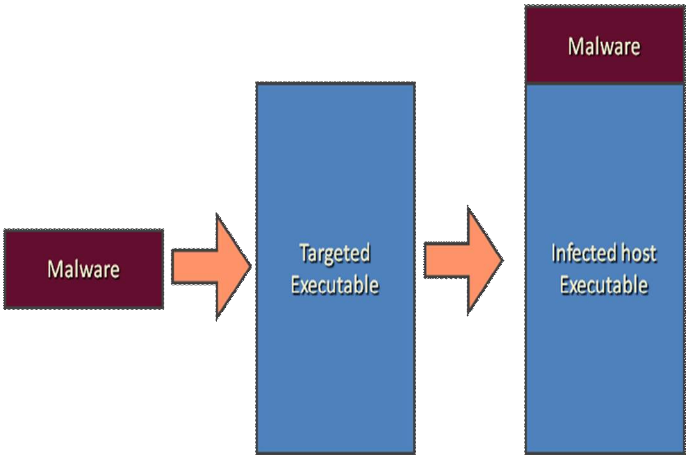


Fig.2.B. Pre-Pending Malware

The third variation of the malware is an Appending malware. The following system model clearly shows the variation:

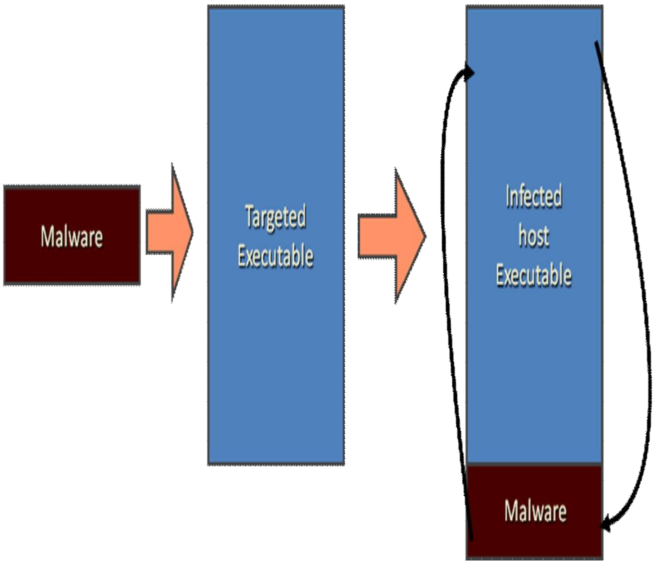


Fig.2.C. Appending Malware

The fourth variation of the malware is a Cavity malware. The following system model clearly shows the variation:

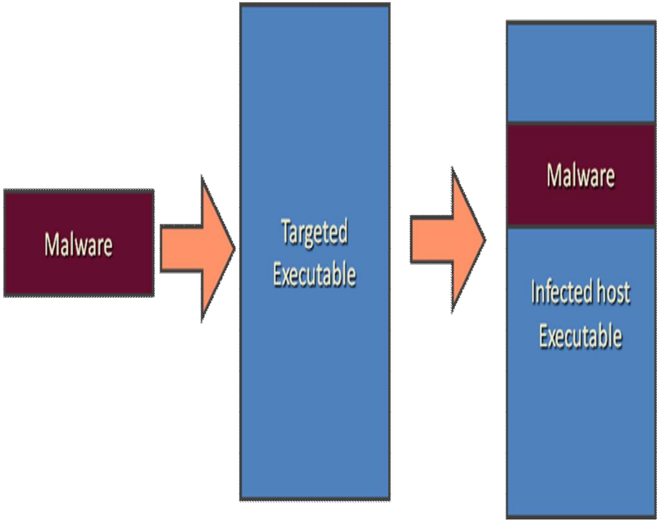


Fig.2.D. Cavity Malware

The fifth variation of the malware is a Multi-Cavity malware. The following system model clearly shows the variation:

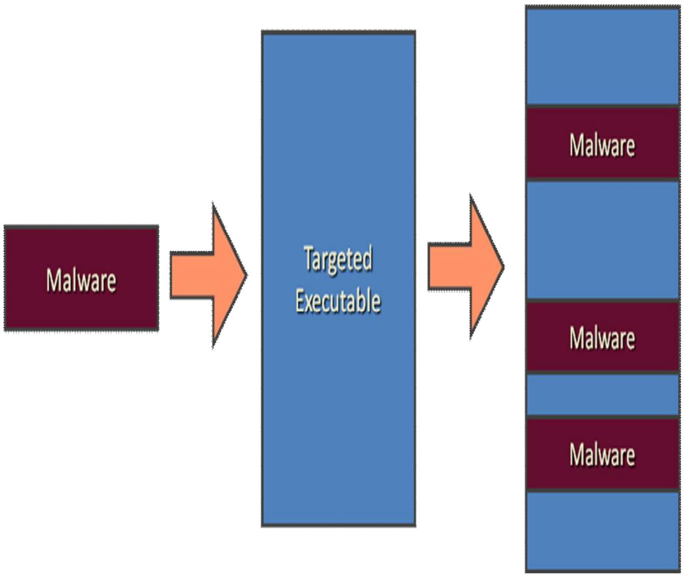


Fig.2.E. Multi-Cavity Malware

D. Advanced Random-RBF Generation Algorithm:

A new algorithm is required to create a random set of centers for each distinct class, which is called *"Advanced Random-RBF Generation Algorithm"*. This algorithm is purely based on the RandomRBF generation technique. Advanced Random-RBF data is generated by first creating a random set of centers for each class. Each center is randomly assigned a weight, a central point per attribute, and a standard deviation. To generate new instances, a center is chosen at random taking the weights of each center into consideration. Attribute values are randomly generated and offset from the center, where the overall vector has been scaled so that its length equals a value sampled randomly from the Gaussian distribution of the center [6]. The particular center chosen determines the class of the instance. Advanced Random-RBF data contains only numeric attributes as it is non-trivial to include nominal values.

Algorithm Implementations:

```
public AdvancedRandomRBF()
{
super();
setNumAttributes(defaultNumAttributes());
setNumClasses(defaultNumClasses());
setNumCentroids(defaultNumCentroids());
}
public Enumeration listOptions()
{
Vector result;

result = enumToVector(super.listOptions());
result.addElement(new Option("\tThe number of attributes (default " + defaultNumAttributes() + ").", "a", 1, "-a
<num>"));

result.addElement(new Option("\tThe number of classes (default " + defaultNumClasses() + ")","c", 1, "-c <num>"));
result.add(new Option("\tThe number of centroids to use. (default " + defaultNumCentroids() + ")","C", 1, "-C <num>"));
return result.elements();
}

* Generates a comment string that documentats the data generator.
* By default this string is added at the end of theproduces output as ARFF file type.
* @return string contains info about the generated rules
* @throws Exception if the generating of the documentaion fails

public String generateStart () {
StringBuffer result= new StringBuffer();
int i;
result.append("%\n");
result.append("% centroids:\n");
for (i = 0; i < getNumCentroids(); i++)
result.append("% " + i + ".: " + Utils.arrayToString(m_centroids[i]) + "\n");
result.append("%\n");
result.append("% centroidClasses: " + Utils.arrayToString(m_centroidClasses) + "\n");
result.append("%\n");
result.append("% centroidWeights: " + Utils.arrayToString(m_centroidWeights) + "\n");
result.append("%\n");
result.append("% centroidStdDevs: " + Utils.arrayToString(m_centroidStdDevs) + "\n");
result.append("%\n");
return result.toString();
}
```

III. MAJOR CONTRIBUTIONS

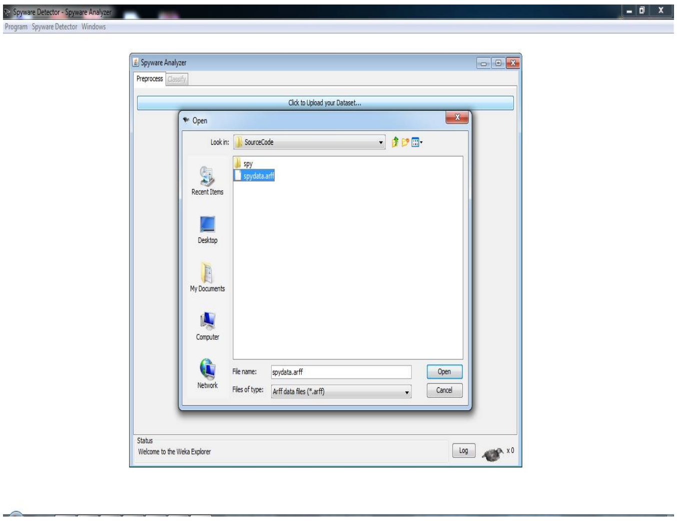
The major contribution of the paper is fully described and explained in the following sentences with details subscriptions. An also the implementation rules with clear outcome scenarios are also explained in the following. Just look the following way to clear the questionnaires and try to implement the concept with advanced proportions.

1. We in attendance an all-purpose behavioral classification of nearness malware, which imprisons the practical other than defective natural history in noticed closeness malware.

2. Beneath the behavioral malware description, and by means of a straightforward cut off malware repression approach, we put together the malware uncovering development as a disseminated pronouncement difficulty. We examine the hazard connected with the pronouncement, and intend a straightforward, yet effectual, approach, give the impression of being to the front, which logically replicates personality nodes fundamental hazard preferences adjacent to malware contagion. Seem to be ahead extending the naive Bayesian replica, and take in hands the Delay Tolerant Network exact, malware connected, “Insufficient Evidence Versus Evidence Collection Risk” difficulty.

3. We believe the benefits of distribution appraisals in the middle of nodes, and tackle confronts resulting from the DTN replica: liars (i.e., bad-mouthing and false praising malicious nodes) and traitors (i.e., good nodes that have turned rogue due to malware infections). We in attendance two substitute modus operands, unbending pass through a filters and adaptive come across in advance, that unsurprisingly lengthen seem to be further on to fuse substantiation presented by others, although be full of the downbeat cause of copied confirmation. A pleasant possession of the planned confirmation consolidation techniques is that the consequences will not get worse smooth if liars are the preponderance in the district. Authentic get in touch with traces are used to confirm the efficiency of the techniques.

E. EXPERIMENTAL OUTCOMES

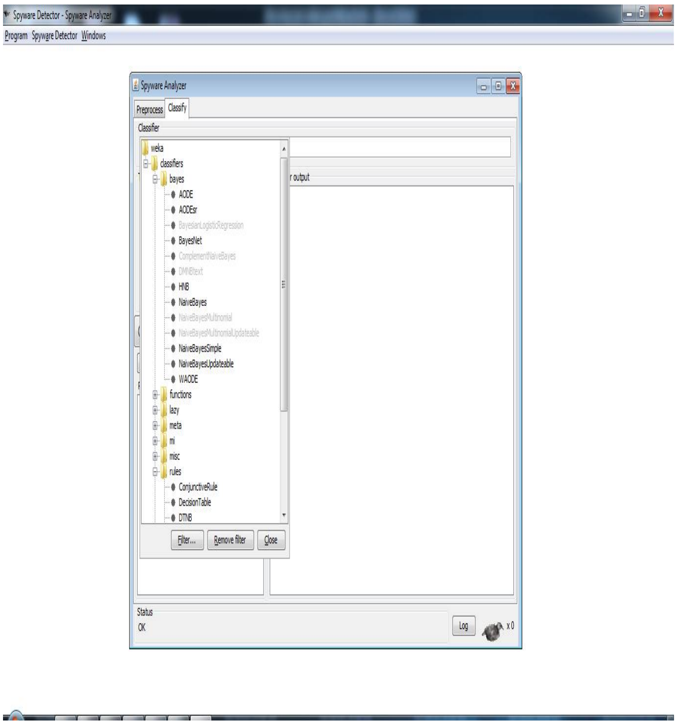


File name: spydata.arff

Files of type: Arff data files (*.arff)

Welcome to the Weka Explorer

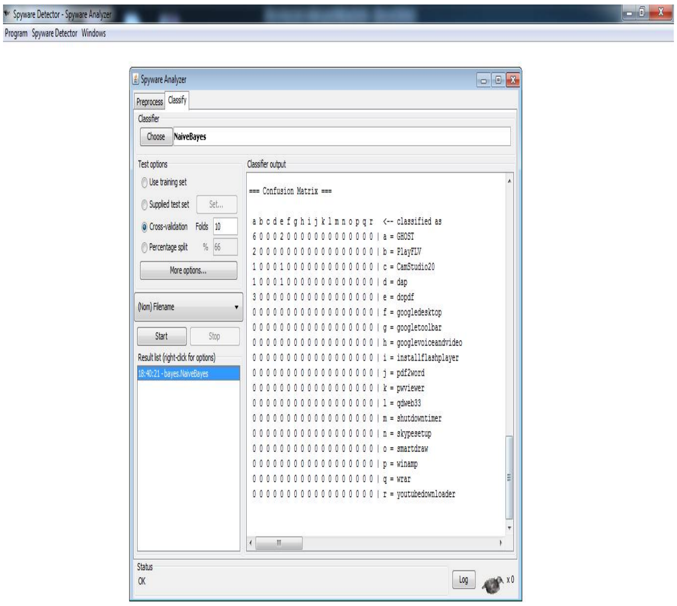
Fig.2. Loading Dataset



OK

Fig.3. Classification in Progress

DOI: 10.15680/IJIRCCE.2016. 0404281


Spyware Detector - Spyware Analyzer
Program Spyware Detector Windows
Spyware Analyzer
Preprocess Classify

Classifier: Choose NaiveBayes

Test options
○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66
More options...
(Nom) Filename
Start Stop
Result list (right-click for options)
18:40:21 - bayes.NaiveBayes

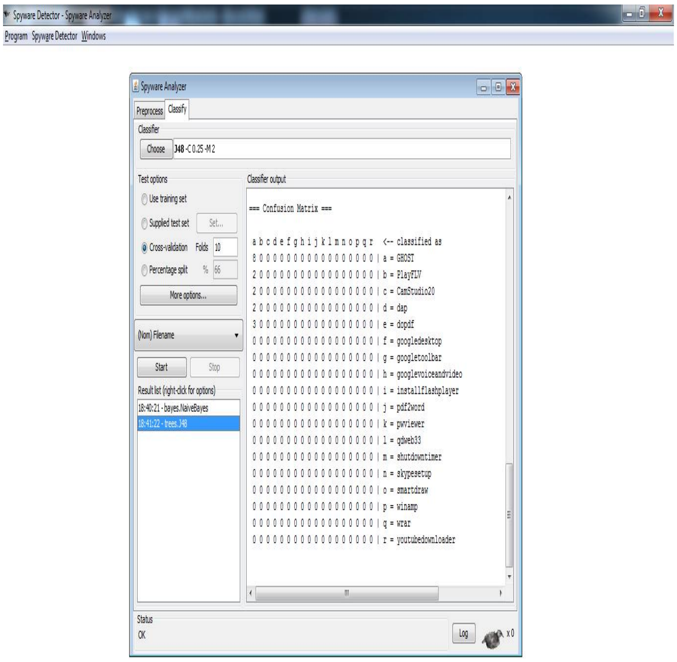
Classifier output

```
=== Confusion Matrix ===

 a b c d e f g h i j k l m n o p q r   <-- classified as
 6 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 | a = GHOST
 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | b = PlayFLV
 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 | c = CamStudio20
 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 | d = dap
 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | e = dopdf
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | f = googledesktop
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | g = googletoolbar
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | h = googlevoiceandvideo
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | i = installflashplayer
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | j = pdf2word
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | k = pwviewer
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | l = qdweb33
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | m = shutdowntimer
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | n = skypesetup
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | o = smartdraw
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | p = winamp
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | q = wrar
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | r = youtubedownloader
```

Status
OK

Fig.4. Naïve Bayes Rule


Spyware Detector - Spyware Analyzer
Program Spyware Detector Windows
Spyware Analyzer
Preprocess Classify

Classifier: Choose J48 -C 0.25 -M 2

Test options
○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66
More options...
(Nom) Filename
Start Stop
Result list (right-click for options)
18:40:21 - bayes.NaiveBayes
18:41:22 - trees.J48

Classifier output

```
=== Confusion Matrix ===

 a b c d e f g h i j k l m n o p q r   <-- classified as
 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | a = GHOST
 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | b = PlayFLV
 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | c = CamStudio20
 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | d = dap
 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | e = dopdf
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | f = googledesktop
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | g = googletoolbar
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | h = googlevoiceandvideo
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | i = installflashplayer
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | j = pdf2word
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | k = pwviewer
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | l = qdweb33
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | m = shutdowntimer
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | n = skypesetup
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | o = smartdraw
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | p = winamp
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | q = wrar
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | r = youtubedownloader
```

Status
OK

Fig.5. Tree Rule Composition

IV. Conclusion

Behavioral description of malware is an effectual substitute to prototype corresponding in detects malware, in particular at what time commerce with polymorphic or obfuscated malware. Naive Bayesian replica has been productively functional in non Delay Tolerant Network surroundings, such as riddles email spams and become aware of botnets. A wide-ranging behavioral classification of Delay Tolerant Network is proposed to pedestal nearness malware. We nearby appears to the front, the length of with inflexible riddles and adaptive give the impression of being to the front, to deal with two only one of its kinds demanding in expanding Bayesian sieves to Delay Tolerant Networks- "Insufficient Evidence Versus Evidence Collection Risk" in addition to "Filtering False Evidence Sequentially and Distributedly". In scene, additional room of the behavioral description of closeness malware to explanation for deliberate malware uncovering dodging with game speculation is a difficult yet appealing expectations vocation.

References

[1] P. Akritidis, W. Chin, V. Lam, S. Sidiroglou, and K. Anagnostakis, "Proximity Breeds Danger: Emerging Threats in Metro-Area Wireless Networks," Proc. 16th USENIX Security Symp., 2007.
[2] A. Lee, "FBI Warns: New Malware Threat Targets Travelers, Infects via Hotel Wi-Fi," http://goo.gl/D8vNU, 2012.
[3] NFC Forum. about NFC, http://goo.gl/zSJqb, 2013.
[4] Wi-Fi Alliance. Wi-Fi Direct, http://goo.gl/fZuyE. 2013.
[5] C. Kolbitsch, P. Comparetti, C. Kruegel, E. Kirda, X. Zhou, and X.Wang, "Effective and Efficient Malware Detection at the End Host," Proc. 18th Conf. USENIX Security Symp., 2009.
[6] U. Bayer, P. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "Scalable, Behavior-Based Malware Clustering," Proc. 16th Ann. Network and Distributed System Security Symp. (NDSS), 2009.
[7] D. Dash, B. Kveton, J. Agosta, E. Schooler, J. Chandrashekar, A. Bachrach, and A. Newman, "When Gossip is Good: Distributed Probabilistic Inference for Detection of Slow Network Intrusions," Proc. 21st Nat'l Conf. Artificial Intelligence (AAAI), 2006.

[8] G. Zyba, G. Voelker, M. Liljenstam, A. Me´hes, and P. Johansson, "Defending Mobile Phones from Proximity Malware," Proc. IEEE INFOCOM, 2009.
[9] F. Li, Y. Yang, and J. Wu, "CPMC: An Efficient Proximity Malware Coping Scheme in Smartphone-Based Mobile Networks," Proc. IEEE INFOCOM, 2010.
[10] I. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos, "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-Mail Messages," Proc. 23rd Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2000.
[11] P. Graham, "Better Bayesian Filtering," http://goo.gl/AgHkB, 2013.
[12] J. Zdziarski, Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification. No Starch Press, 2005.
[13] R. Villamarı´n-Salomo´n and J. Brustoloni, "Bayesian Bot Detection Based on DNS Traffic Similarity," Proc. ACMymp. Applied Computing (SAC), 2013.
[14] J. Agosta, C. Diuk-Wasser, J. Chandrashekar, and C. Livadas, "An Adaptive Anomaly Detector for Worm Detection," Proc. Second USENIX Workshop Tackling Computer Systems Problems with Machine Learning Techniques (SYSML), 2007.
[15] S. Marti et al., "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proc. ACM MobiCom, 2000.

BIOGRAPHY

A. G.SATHISH KUMAR received his M.E. degree in Computer Science Engineering from ANNA UNIVERSITY, Thiruchirappalli, 2010. He is a candidate for a PhD degree in the Computer Networks. at Sathyabama University, Chennai. His research areas include Network Security, Information Security, and Computer security. Now he is working as a Senior Assistant Professor at MNSK College of Engineering, Pudukkottai. His Working Experience is 4 years 10 months in Teaching Field. He was joined a research fellow from Sathyabama University at 2013 JAN. He presented and participated papers in various colleges International and national conferences.

Prof. Dr. G. GUNASEKARAN, earned his P.H.D. Degree in Jadavpur University, Kolkata in the year of 2009 and he is presently working as a Principal in Meenakshi College of Engineering, 12, Vembuliamman Koil Street, West K.K.Nagar, Chennai-78, Anna University. He has 16 years of Teaching Experience and 11 Years of Research Experience and 7 Years of Industrial Experience.

DOI: 10.15680/IJIRCCE.2016. 0404281