Fall 2015

Real-Time Detection System for Suspicious URLs

Krishna Prasad Chouty
Governors State University

Anup Chandra Thogiti
Governors State University

Kranthi Sudha Vudatha
Governors State University

REAL-TIMEDETECTION SYSTEM FOR SUSPICIOUS URL’S

By

Krishna Prasad, Chouty
B.Tech, JAWAHARLAL NEHURU TECHNOLOGICAL UNIVERSITY, 2012
AnupChandra, Thogiti
B.Tech, JAWAHARLAL NEHURU TECHNOLOGICAL UNIVERSITY, 2011
Kranthi Sudha, Vudatha
B.Tech, JAWAHARLAL NEHURU TECHNOLOGICAL UNIVERSITY, 2008

GRADUATECAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science

Governors State University
University Park, IL 60484

2015

Table of Contents

REAL-TIME DETECTION SYSTEM FOR SUSPICIOUS URL’S:

1 Project Description

1.1 Project Abstract

Twitter is prone to malicious tweets containing URLs for spam, phishing, and malware distribution. Conventional Twitter spam detection schemes utilize account features such as the ratio of tweets containing URLs and the account creation date, or relation features in the Twitter graph. These detection schemes are ineffective against feature fabrications or consume much time and resources. Conventional suspicious URL detection schemes utilize several features including lexical features of URLs, URL redirection, HTML content, and dynamic behavior. However, evading techniques such as time-based evasion and crawler evasion exist. In this paper, we propose WARNINGBIRD, a suspicious Real-Time URL detection system for Twitter. Our system investigates correlations of URL redirect chains extracted from several tweets. Because attackers have limited resources and usually reuse them, their URL redirect chains frequently share the same URLs. We develop methods to discover correlated URL redirect chains using the frequently shared URLs and to determine their suspiciousness. We collect numerous tweets from the Twitter public timeline and build a statistical classifier using them. Evaluation results show that our classifier accurately and efficiently detects suspicious URLs

1.2 Competitive Information

In the existing system attackers use shortened malicious URLs that redirect Twitter users to external attack servers. To cope with malicious tweets, several Twitter spam detection schemes have been proposed. These schemes can be classified into account feature-based, relation feature-based, and message feature based schemes. Account feature-based schemes use the distinguishing features of spam accounts such as the ratio of tweets containing URLs, the account creation date, and the number of followers and friends. However, malicious users can easily fabricate these account features. The relation feature-based schemes rely on more robust features that malicious users cannot easily fabricate such as the distance and connectivity apparent in the Twitter graph. Extracting these relation features from a Twitter graph, however, requires a significant amount of time and resources as a Twitter graph is tremendous in size. The message feature-based scheme focused on the lexical features of messages. However, spammers can easily change the shape of their messages. A number of suspicious URL detection schemes have also been introduced.

1.3 Relationship to Other Applications/Projects

To adapt to noxious tweets, a few Twitter spam identification plans have been proposed. These plans can be characterized into record highlight based, connection highlight based, and message, for example, the proportion of tweets containing URLs, the record creation date, and the quantity of adherents highlight based plans. Record highlight based plans utilize the recognizing elements of spam records and companions.

1.4 Assumptions and Dependencies

The connection highlight construct plans depend in light of more powerful elements that malevolent clients can't without much of a stretch manufacture, for example, the separation and network obvious in the Twitter chart. Removing these connection highlights from a Twitter diagram, on the other hand, requires a lot of time and assets as a Twitter chart is gigantic in size. The message highlight construct plan centered with respect to the lexical elements of messages. Be that as it may, spammers can without much of a stretch change the state of their messages. Various suspicious URL recognition plans have additionally been presented.

1.5 Definitions and Acronyms

Definition

Phishing: Phishing email will typically direct the user to visit a website where they are asked to update personal information, such as a password, credit card, social security, or bank account numbers, that the legitimate organization already has. The website, however, is bogus and will capture and steal any information the user enters on the page.

Crawler: A crawler is a program that visits Web sites and reads their pages and other information in order to create entries for a search engine index Crawlers are typically programmed to visit sites that have been submitted by their owners as new or updated. Entire sites or specific pages can be selectively visited and indexed.

Acronyms

URL- Uniform Resource Locator

HTTP- Hypertext Transfer Protocol

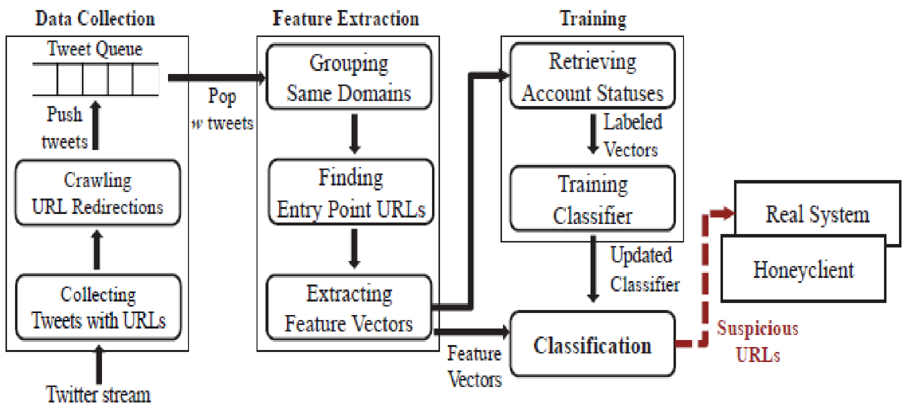
HTML- Hypertext Markup Language.

2 Technical Description

In this paper, we propose WARNINGBIRD, a suspicious URL detection system for Twitter. Instead of investigating the landing pages of individual URLs in each tweet, which may not be successfully fetched, we considered correlations of URL redirect chains extracted from a number of tweets. Because attacker's resources are generally limited and need to be reused, their URL redirect chains usually share the same URLs. We therefore created a method to detect correlated URL redirect chains using such frequently shared URLs. By analyzing the correlated URL redirect chains and their tweet context information, we discover several features that can be used to classify suspicious URLs. We collected a large number of tweets from the Twitter public timeline and trained a statistical classifier using the discovered features.

- We present a new suspicious URL detection system for Twitter that is based on the correlations of URL redirect chains, which are difficult to fabricate. The system can find correlated URL redirect chains using the frequently shared URLs and determine their suspiciousness in almost real time.
- We introduce new features of suspicious URLs: some of which are newly discovered and while others are variations of previously discovered features.
- We present the results of investigations conducted on suspicious URLs that have been widely distributed through Twitter over several months.

2.1 Project/Application Architecture



System overview

2.2 Project/Application Information flows

Module Description:

1. **Data collection**
2. **Feature extraction**
3. **Training**
4. **Classification**

Data collection: The data collection component has two subcomponents: the collection of tweets with URLs and crawling for URL redirections. To collect tweets with URLs and their context information from the Twitter public timeline, this component uses Twitter Streaming APIs. Whenever this component obtains a tweet with a URL, it executes a crawling thread that follows all redirections of the URL and looks up the corresponding IP addresses. The crawling thread appends these retrieved URL and IP chains to the tweet information and pushes it into a tweet queue. As we have seen, our crawler cannot reach malicious landing.
URLs when they use conditional redirections to evade crawlers. However, because our detection system does not rely on the features of landing URLs, it works independently of such crawler evasions.

Feature extraction: The feature extraction component has three subcomponents: grouping of identical domains, finding entry point URLs, and extracting feature vectors.
This component monitors the tweet queue to determine whether a sufficient number of tweets have been collected. Specifically, our system uses a tweet window instead of individual tweets. When more than *w* tweets are collected (*w* is 10,000 in the current implementation), it pops *w* tweets from the tweet queue. First, for all URLs in the *w* tweets, this component checks whether they share the same IP addresses. If several URLs share at least one IP address, it replaces their domain names with a list of domains with which they are grouped.

Training: The training component has two subcomponents: retrieval of account statuses and training of the classifier. Because we use an offline supervised learning algorithm, the feature vectors for training are relatively older than feature vectors for classification. To label the training vectors, we use the Twitter account status; URLs from suspended accounts are considered malicious whereas URLs from active accounts are considered benign. We periodically update our classifier using labeled training vectors.
Classification: The classification component executes our classifier using input feature vectors to classify suspicious URLs. When the classifier returns a number of malicious feature vectors, this component flags the corresponding URLs and their tweet information as suspicious.
These URLs, detected as suspicious, will be delivered to security experts or more sophisticated dynamic analysis environments for an in-depth investigation.

2.3 Interactions with other Projects (if Any)

We compared the efficiency of WARNINGBIRD with that of Twitter’s detection system. For the comparison, we sampled 14,905 accounts detected by our online WARNINGBIRD system between September1,2011 and October 22, 2011. To compare their efficiencies, we measured the time difference between WARNINGBIRD’s detection and Twitter’s suspension of the accounts. We monitored the WARNINGBIRD to obtain newly detected suspicious accounts and then checked the status of each account every 15 s, for one day, until it was suspended. Among the sampled accounts, 5,380 accounts were suspended within a day; 37.3% of them were suspended within a minute, another 44.3% of them were suspended within four hours, and the remaining 18.4% of them were suspended within a day.

The average time difference was 13.5 min, which shows that our detection system is more efficient than that of Twitter. We also checked the status of the sampled accounts on October 28, 2011 to verify the accuracy of our system. Among the 14,905 accounts, Twitter had suspended 9,250 accounts. We then randomly selected 500 accounts from the remaining 5,655 active accounts to manually check how suspect they were. Among the 500 accounts, 320 accounts were suspicious. Therefore, the detection accuracy of our system given the sample data is about 86.3%.

2.4 Interactions with other Applications

Although WARNINGBIRD is suitable for detecting frequent suspicious URLs distributed by bot accounts (which are common on Twitter[30]),we need to consider more advanced attacks using compromised accounts . We can classify compromised Twitter accounts into two types: (i) accounts authorizing malicious applications and (ii) accounts stolen by attackers. Twitter users may accidently (or intentionally) authorize malicious applications luring them with interesting advertisements, such as enticements to increase the number of their followers or notify them regarding their unfollowers. User accounts may also be stolen by attackers guessing or stealing their passwords. In such cases, five account similarity-based features, i.e., the number of source applications and the similarities in the account creation dates, the number of followers, the number of friends, and the follower-friend ratio, are no longer effective.

2.5 Capabilities

- We present a new suspicious URL detection system for Twitter that is based on the correlations of URL redirect chains, which are difficult to fabricate. The system can find correlated URL redirect chains using the frequently shared URLs and determine their suspiciousness in almost real time.
- We introduce new features of suspicious URLs: some of which are newly discovered and while others are variations of previously discovered features.
- We present the results of investigations conducted on suspicious URLs that have been widely distributed through Twitter over several months

2.6 Risk Assessment and Management

Multiple redirections: Web pages can embed several external pages and different content. Therefore, some pages can cause multiple redirections. Because our system currently only considers HTTP redirection and does not consider page-level redirection, it cannot catch multiple redirections. Therefore, we need customized browsers to catch and address multiple redirections.

Dynamic redirection: Currently, WARNINGBIRD uses a static crawler written in Python. Because it can only handle HTTP redirections, it is ineffective on pages that have embedded dynamic redirections such as JavaScript or Flash redirection. Therefore, WARNINGBIRD will designate pages with embedded dynamic redirection as entry point URLs. This determination causes inaccuracy in some of the feature values, including the redirect chain lengths, positions of the entry point URLs, and the number of different landing URLs. Therefore, in the future we will use customized Web browsers to fully retrieve redirect chains.

Coverage and scalability: Currently, our system only monitors one percent of the samples from the Twitter public timeline, because our accounts only have the Spritzer access role. The current implementation, however, cannot handle100%oftheTwitterpublictimeline.Therefore,we need to extend WARNINGBIRD to a distributed detection system, for instance, Monarch [19], to handle the entire Twitter public timeline.

3 Project Requirements

3.1 Identification of Requirements

We performed a simple investigation on three days' worth of tweet samples culled from July 23 to 25, 2011. We extracted frequent URL redirect chains from the sample data and ranked them according to their frequency after removing white listed domain names. Many suspicious sites, such as jbfollowme.com, which attempts to attract Justin Bieber's fans, proved to be highly ranked.

We consider blackraybansunglasses.com, which is a suspicious site associated with spam tweets. We first encountered this site in April 2011 and it was active until August 2011. We used a one percent of a sample of tweets collected on July 11, 2011, to conduct an in-depth analysis of the site . blackraybansunglasses.com has a page, redirect.php, which conditionally redirects users to random spam pages. It uses a number of different Twitter accounts and shortened URLs to distribute its URL to other Twitter users. According to our dataset, it uses 6,585 different Twitter accounts and shortened URLs, and occupies about 2.83% of the sampled 232,333 tweets with URLs. When a user clicks on one of the shortened URLs, such as bit.ly/raCz5i distributed by zarzuelavbafpv0, heorshewillberedirectedtoaprivate redirection site, such as beginnersatlanta.tk, which seems to be managed by the operator of blackraybansunglasses.com. The user will then be repeatedly redirected to bestfreevideoonline.info and blackraybansunglasses.com. The redirection site blackraybansunglasses.com evaluates whether its visitors are normal browsers or crawlers using several methods, including cookie and user-agent checking. When it is sure that a current visitor is a normal browser, it redirects the visitor to forexstrategysite.com, which then finally redirects him or her to random spam pages. When blackraybansunglasses.com determines that a current visitor is not a normal browser, it simply redirects the visitor to google.com to avoid investigation. Therefore, crawlers may not be able to see forexstrategysite.com or the further spam pages. Another interesting point about blackraybansunglasses.com is that it uses the Twitter Web interface. Conventional Twitter spam detection schemes usually assumed that many spammers would use Twitter APIs to distribute their spam tweets. Advanced Twitter spammers, however, no longer rely on Twitter APIs, because they know that using APIs will distinguish their tweets from normal tweets. For instance, tweetattacks.com sells a Twitter spam program that uses the Web interface to deceive spam receivers and to circumvent API limits

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

We have provided our customers to use this project on real time basis by providing online detection.

The online version of WARNINGBIRD uses a sliding window technique for achieving good latency and detection coverage. A small window gives immediate results; however, it cannot catch suspicious URLs that repeat after long-time intervals. A large window has good detection coverage; however, its latency is bad. A sliding window is a well-known technique for taking advantage of both small and large windows. Let w denote the window size and s denote the sliding size($s \leq w$).Whenever a sliding window system receives s new items, it processes the previous $w - s$ items and the s new items at the same time. Therefore, the latency of this method depends on s and its detection coverage depends on w. Currently, we have set w at 10,000 and s at 2,000. About every 12 min, the online version of WARNINGBIRD returns suspicious URLs that have appeared

In the previous hour—near real time detection. Because our system can process 10,000 collected tweets in less than one minute (Fig. 12), we can detect suspicious URLs with only one-minute time lags. In addition, we could set s at 200 to detect suspicious URLs about every 1.2 min. However, because we do not want to make our system heavily burdened, we have not use such parameter.

3.3 Security and Fraud Prevention

Feature evasion methods: Attackers can fabricate the features of their attacks to evade our detection system. For instance, they can use short redirect chains, change the position of their entry point URLs, reuse initial and landingURLs, or use a small number of different domain names and IP addresses. These modifications, paradoxically, would allow conventional detection systems to detect their malicious URLs. Attackers may also be able to reduce the frequency of their tweets to bypass our detection system. However, this would also reduce the number of visitors to their malicious pages. Features derived from tweet information, however, are relatively weak at protecting against forgery. Attackers could use a large number of source applications and Twitter accounts, use similar tweet texts, and carefully adjust the numbers of followers and friends of their accounts to increase the standard deviation values. In addition, they could increase the standard deviation of their account creation date if they own or have compromised older accounts. Although these features are weak, attackers have to consume their resources and time to fabricate these features. Therefore, using these features is still meaningful. The strongest evasion method is definitely to increase the number of redirect servers. This method, however, would require a lot of resource and large financial investment on the part of the attackers.

Adaptation to the other services: Although WARNINGBIRD is designed for Twitter, with some simple modifications it can also be applied to other services that can monitor a continuous URL stream. For example, we can consider an e-mail service that continuously processes a large number of e-mails for its users. Its operators can collect and investigate e-mails containing

URLs. When a proper number of such e-mails are collected, the URL based features can be extracted, such as the length of the URL redirect chain, the frequency of entry point URLs, and the number of different initial and landing URLs. The operators can also extract other features from e-mail context information such as the number of senders and receivers, the number of mail servers and relay servers, and similarities in e-mail messages. Web forum services are also similar; as their operators can collect all posts and comments of users containing URLs and can extract URL-based features as well as other features including user IDs, IP addresses, and message similarities. We can modify WARNINGBIRD to use the above features for detecting suspicious URLs on those systems. A similar method can also be applied to other social networking services such as Facebook and Google+.

3.4 Release and Transition Plan

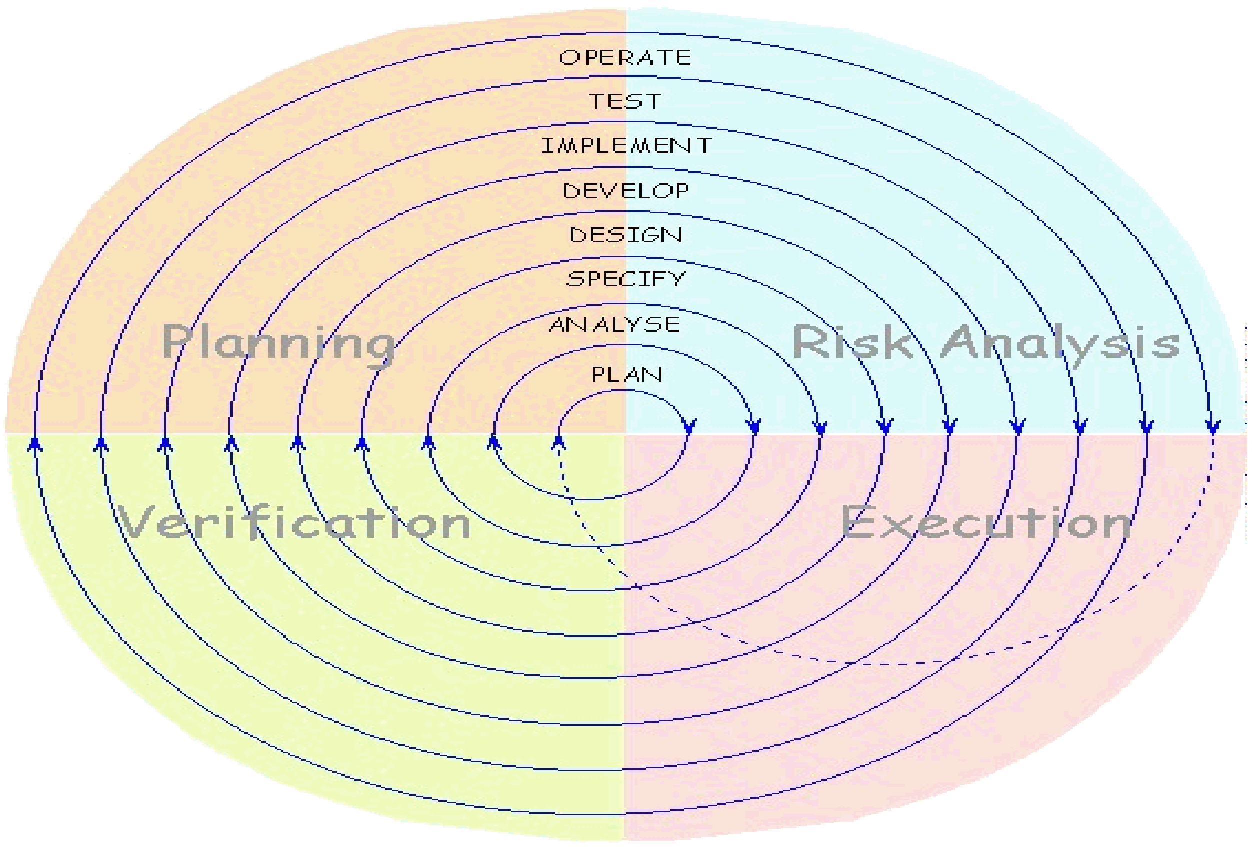
In this paper, we propose WARNINGBIRD, a suspicious URL detection system for Twitter. Instead of investigating the landing pages of individual URLs in each tweet, which may not be successfully fetched, we considered correlations of URL redirect chains extracted from a number of tweets. Because attacker's resources are generally limited and need to be reused, their URL redirect chains usually share the same URLs. We therefore created a method to detect correlated URL redirect chains using such frequently shared URLs. By analyzing the correlated URL redirect chains and their tweet context information, we discover several features that can be used to classify suspicious URLs. We collected a large number of tweets from the Twitter public timeline and trained a statistical classifier using the discovered features. The trained classifier is shown to be accurate and has low false positives and negatives.

A number of suspicious URL detection schemes have also been introduced. They use static or dynamic crawlers, and they may be executed in virtual machine honeypots, such as Capture-HPC , HoneyMonkey , and Wepawet, to investigate newly observed URLs. These schemes classify URLs according to several features including lexical features of URLs, DNS information, URL redirections, and the HTML content of the landing pages. Nevertheless, malicious servers can bypass an investigation by selectively providing benign pages to crawlers. For instance, because static crawlers usually cannot handle JavaScript or Flash, malicious servers can use them to deliver.

4 Project Design Description

Our system consists of two Intel Quad Core Xeon E5530 2.40GHz CPUs and 24 GiB of main memory. To collect the tweets, we used Twitter Streaming APIs [31]. Our accounts have a Spritzer access role, and thus we can collect about one percent of all tweets from the Twitter public timeline as samples. From April 8 to December 8, 2011 (245 days in total), we collected 59,056,761 samples of tweets with URLs. We observed about 240,000 tweets daily on average. Our system visited all the URLs in the tweets to collect the URL redirect chains. In addition, starting on July 23, our system collected the IP addresses of all URLs for the domain grouping. From the collected tweets, we found 13,261,069 unique Twitter accounts. Among them, 1,339,496 accounts (10.1%) were suspended as of January 15, 2012.

Twitter announced that it had started to wrap URLs with lengths longer than 19 characters using its URL shortening service t.co [33] from August 15, 2011 and that it started to wrap all URLs regardless of their length from October 10, 2011 [34]. We noticed that this additional layer of URL redirections affects our classification results; therefore, from August 15, 2011, we decided to remove the first t.co URLs in redirect chains.



SPIRALMODEL

Labeling is essential for classification. Unfortunately, we were unable to find a suitable source for labeling our datasets, as many of the URLs in our datasets have not been listed on a public URL blacklist, such as the Google Safe Browsing API. Therefore, instead of URL blacklists, we used Twitter account status information to label our datasets. That is, if some accounts had posted the same URLs andTwitter suspended the account slater,we regarded theURLs as malicious. Otherwise,we regarded them as benign. Our treatment of URLs is acceptable as Thomas et al. have recently confirmed that most suspended accounts are spam accounts.

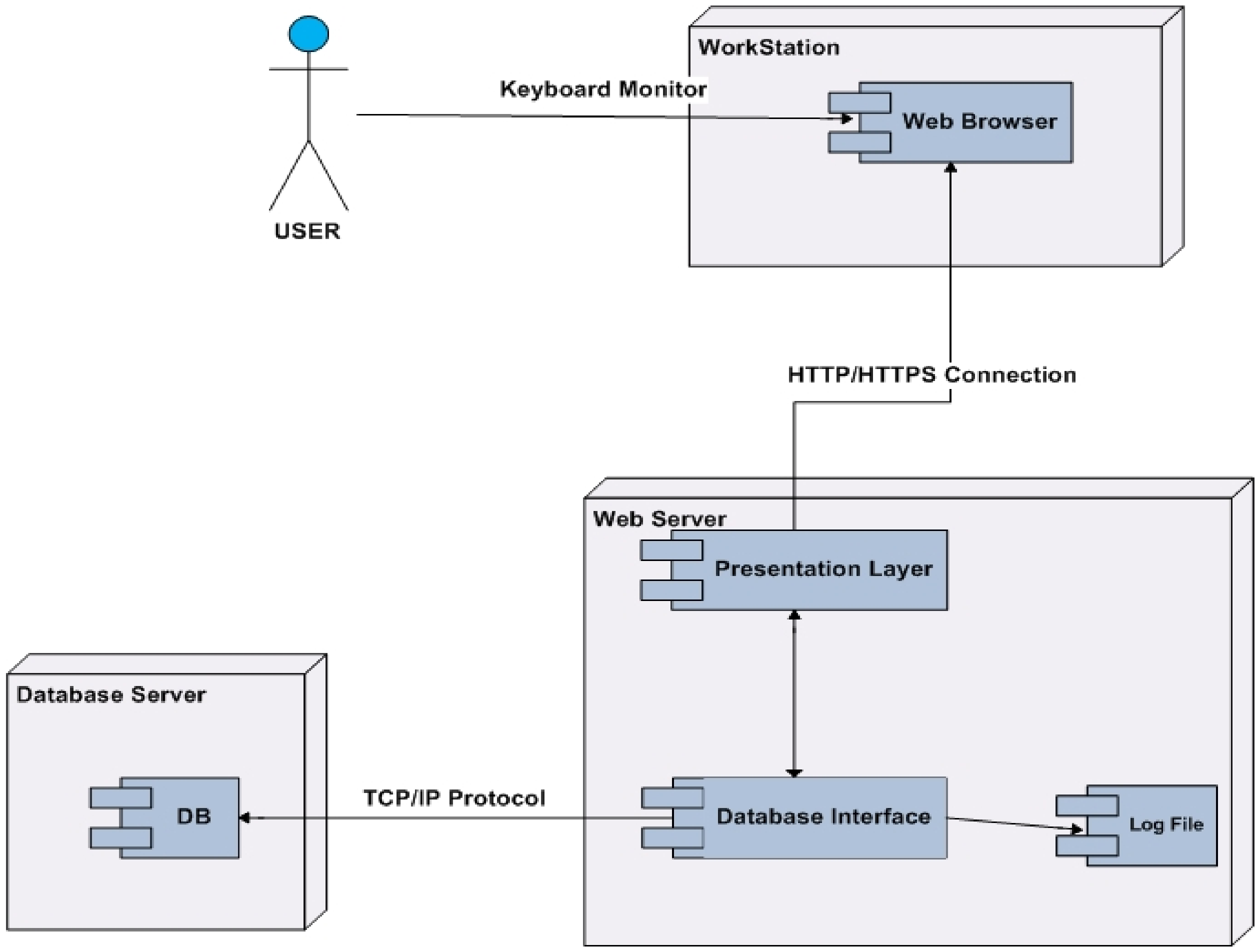
5 Project Internal/external Interface Impacts and Specification

Since we rely on the results of Twitter's spam account detection system to label the collected datasets, one can argue that it just mimics the Twitter's detection system at most. However, most of our features are independent of the Twitter's rules that focus on the suspicious characteristics of individual accounts, such as aggressive following, many tweets with (blacklisted) URLs, a small number of followers compared to the number of followings, and frequently blocked or reported by other users. Twitter can know whether an account violates the rules or not only after the account have performed a series of activities. However, unlike the rules, we focus on the characteristics of URL redirect chains and the similarity of a group of users who uploaded the same URL redirect chains; our system can immediately check them. We also verified that our system can detect suspicious accounts that Twitter cannot detect even several days later. Therefore, we can say that our system is not a simple mimic of the Twitter's detection system. Because the Twitter's detection system had a time delay for suspicious account detection, we checked the status information of accounts at least one month later from their posting of tweets.

Since Twitter is an evolving system, the features of accounts and URLs on the system could change with time. To know how they had been changed during our data collection periods, we checked the F-scores of our features in each month between May 2011 and November2011,and compared six features that had high F-scores in some of the months. The F-scores of the similarity of account creation dates and the relative number of initial URLs had not much changed during the months. This is because the differences between the average feature values of them had not much changed (Fig. 11). On the other hand, the F-scores of the relative number of source applications and the frequency of entry point URLs had increased during the months owing to the reduced number of malicious applications and the reduced frequency of benign URLs. We think the reasons why they reduced are Twitter's efforts to reduce the number of malicious applications and less sampled tweets containing the same benign URLs due to the continuous growth of the number of tweet sit implies that attackers had changed the characteristics of their accounts to avoid detection. two possible explanations are i) attackers really had reduced the lengths of redirect chains because too long chains could be treated as malicious, or ii) they had applied dynamic redirections to prevent simple static.

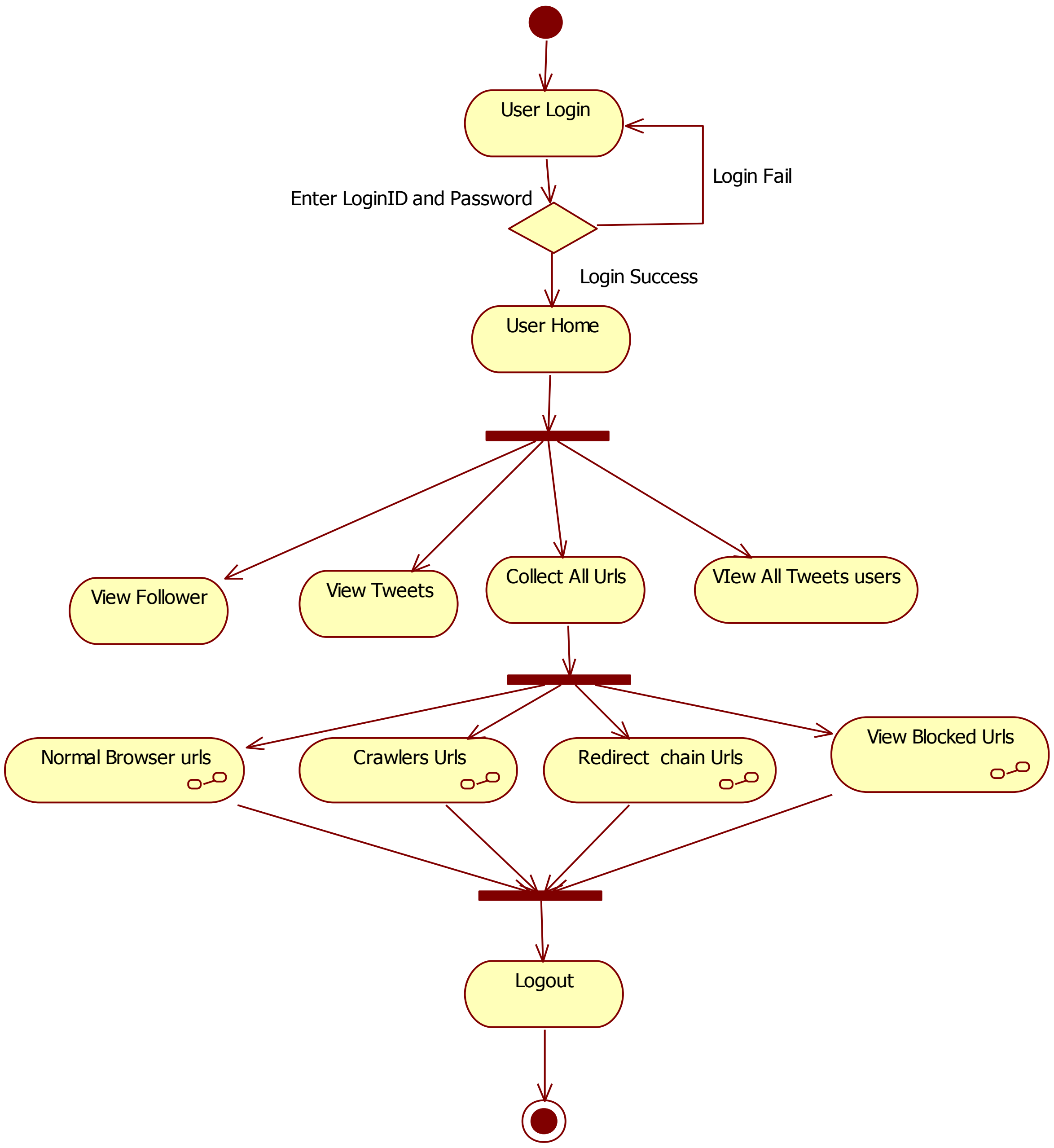
6 Project Design Units Imp

6.1 *Functional Area/Design UnitA*



DEPLOYMENT DIAGRAM

6.1.1 Functional Overview



. **ACTIVITY DIAGRAM**

6.1.2 Impacts

The features derived from there late tweet context information are variations of previously discovered features. However, unlike previous studies that have focused on the differences between malicious and benign accounts, we focused on the similarity of the features of accounts distributing the same entry point URLs. Preparing a large number of dissimilar Twitter accounts for distributing spam URLs becomes a burden to attackers; therefore, similarity checking is effective.

Relative number of different initial URLs: The initial URL is the beginning URL that redirects visitors to the current entry point URL. Attackers usually use a large number of different initial URLs to make their malicious tweets, which redirect visitors to the same malicious URL, look different. The number of different initial .URLs cannot exceed the number of times that their entry point URLs appear. Therefore, if the number of different initial.Content may change prior to final publication.URLs redirecting visitors to an entry point URL that appears n times is i, this feature can be computed as i n.

6.1.3 Requirements

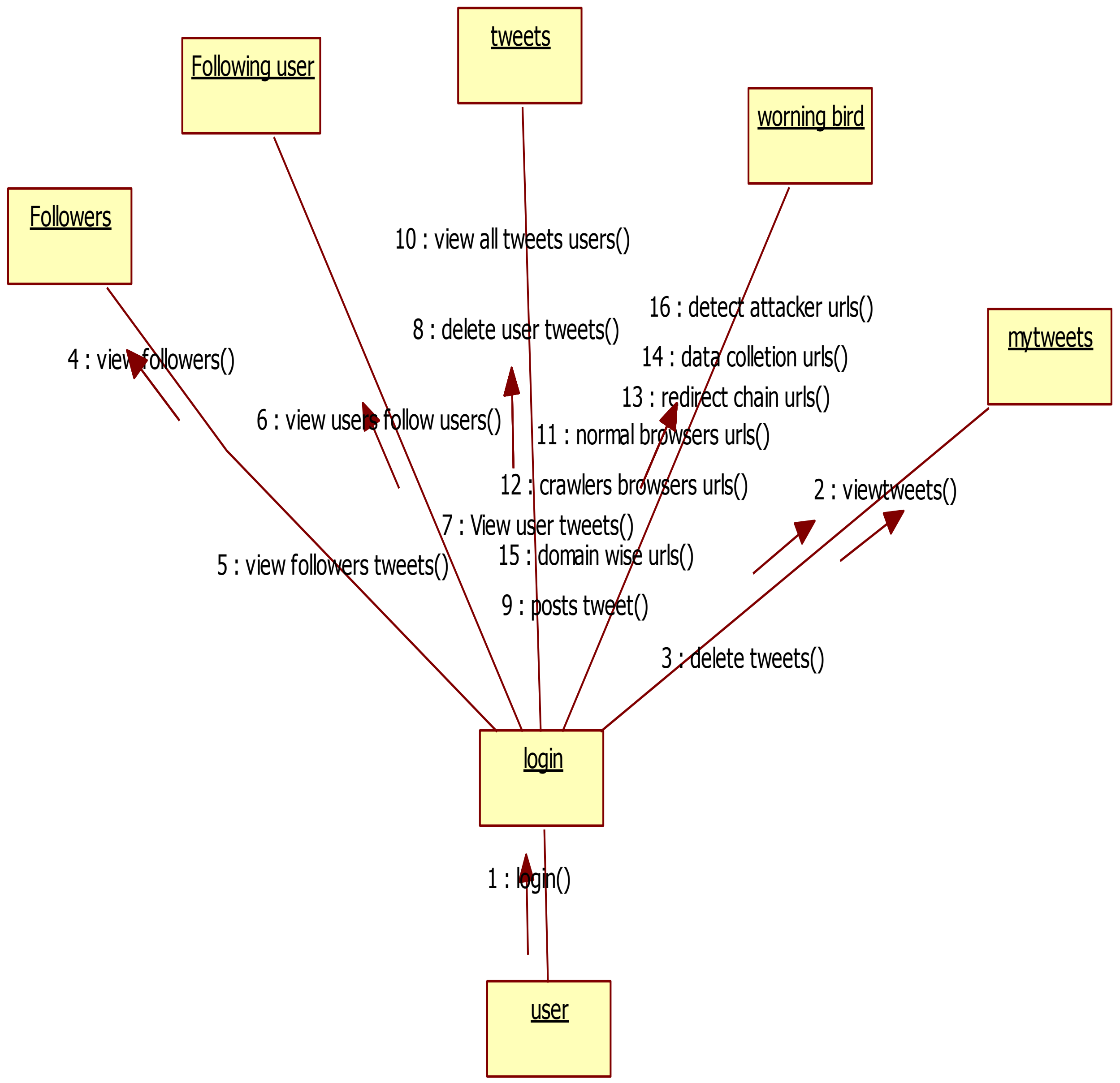
HARDWARE CONFIGURATION:-

Processor	-	Pentium –IV
RAM	-	512MB
Hard disk	-	80GB

SOFTWARE CONFIGURATION

Operating System	: Windows 2007
Programming Language	: JAVA
Frontend	: JSP, Servlets
Backend	: oracle11g
IDE	: my eclipse 8.6

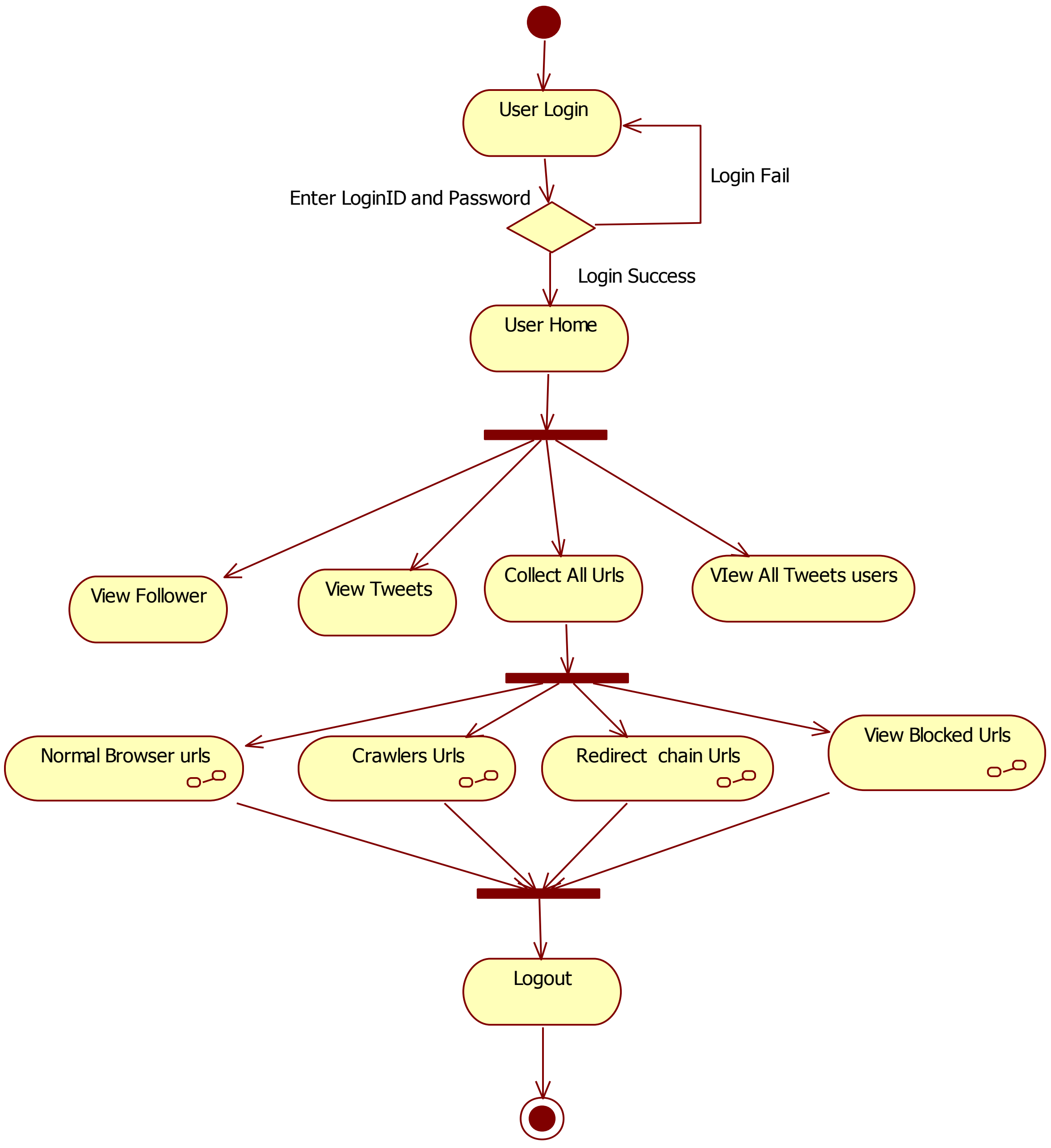
6.2 Functional Area/Design Unit B



tweets
Following user
worning bird
Followers
mytweets
10 : view all tweets users()
16 : detect attacker urls()
8 : delete user tweets()
14 : data colletion urls()
4 : view followers()
13 : redirect chain urls()
6 : view users follow users()
11 : normal browsers urls()
12 : crawlers browsers urls()
2 : viewtweets()
7 : View user tweets()
15 : domain wise urls()
5 : view followers tweets()
9 : posts tweet()
3 : delete tweets()
login
1 : login()
user

COLLABORATION DIAGRAM

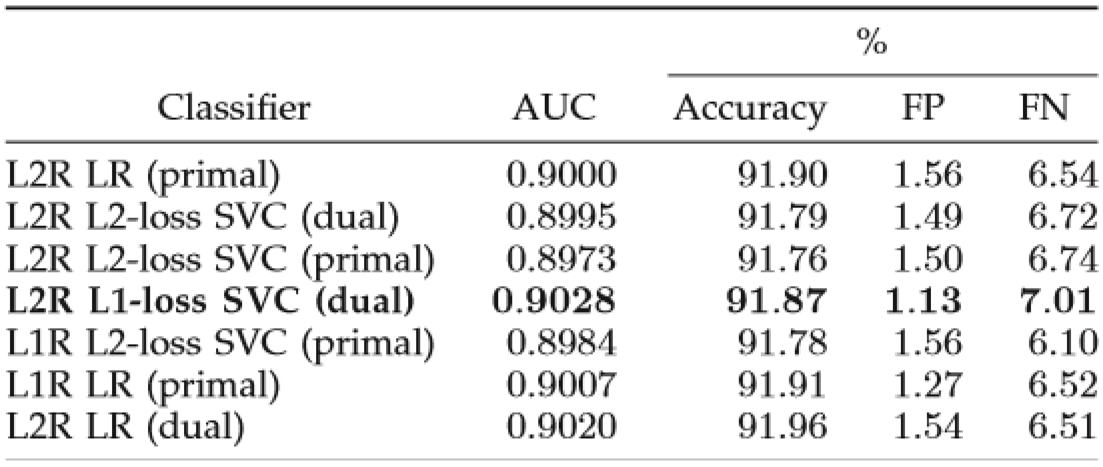
6.2.1 Functional Overview



STATECHART DIARAM

We used sample tweets collected between September 2011 and October 2011 to train the classification models and sample tweets collected during August 2011 and during November 2011 for testing the classifier using older and newer datasets, respectively. From the training dataset, we found 183,846 entry point URLs that appeared more than once in every 10,000 consecutive sample tweets. Among them, 156,896 entry point URLs were benign and 26,950 entry point URLs were malicious. We used the LIBLINEAR library to implement our classifier. We compared seven classification algorithms, and selected an L2-regularized L1-loss support vector classification (SVC) algorithm, since it shows the highest AUC and the lowest FP with the training dataset, experimentally. Table shows the results here,LRisan abbreviation of logistic regression, SVC is support vector classification, AUC is area under the ROC curve, FP is false positive, FN is false negative, L1R and L2R are L1and L2-regularized, and primal and dual represent functions that determine termination of training. Standard deviations of the AUC were 0.0029–0.0032, those of the accuracy were 0.17%–0.20%, those of the FP were 0.05%– 0.09%, and those of the FN were 0.18%–0.19%

Comparing classifiers within a 10- fold cross validation



		%		
Classifier	AUC	Accuracy	FP	FN
L2R LR (primal)	0.9000	91.90	1.56	6.54
L2R L2-loss SVC (dual)	0.8995	91.79	1.49	6.72
L2R L2-loss SVC (primal)	0.8973	91.76	1.50	6.74
L2R L1-loss SVC (dual)	**0.9028**	**91.87**	**1.13**	**7.01**
L1R L2-loss SVC (primal)	0.8984	91.78	1.56	6.10
L1R LR (primal)	0.9007	91.91	1.27	6.52
L2R LR (dual)	0.9020	91.96	1.54	6.51

6.2.2 Impacts

- We present a new suspicious URL detection system for Twitter that is based on the correlations of URL redirect chains, which are difficult to fabricate. The system can find correlated URL redirect chains using the frequently shared URLs and determine their suspiciousness in almost real time.
- We introduce new features of suspicious URLs: some of which are newly discovered and while others are variations of previously discovered features.
- We present the results of investigations conducted on suspicious URLs that have been widely distributed through Twitter over several months.

6.2.3 Requirements

URL redirect chain length: Attackers usually use long URL redirect chains to make 1investigations more difficult and avoid a dismantling gof their servers.Therefore,when an entry point URL is malicious, its chain length l may be longer than those of benign URLs. Frequency of entry point URL: The number of occurrences of the current entry point URL within a tweet window is important. Frequently appearing URLs that are not whitelisted are usually deemed suspicious. Suspicious entry point URLs are not usually located at the end of a redirect chain since they have to conditionally redirect visitors to different landing URLs. Their positions are relative to the lengths of their redirect chains. Therefore, if the position of an entry point of a redirect chain of length l is p, this feature can be computed as p/l.

7 Open Issues

Specialized – Relating to an innovative issue in the task.

II. Business process – Relating to the venture's outline.

III. Change administration – Relating to school, understudies, or ecological changes.

IV. Asset – Relating to hardware, material, or individuals issues.

V. Outsider – Relating to issues with outside plannin

8 Acknowledgements

We kept all our efforts for the success of our project. However, it would not be possible to get this success without the kind support and help of our beloved Professor and Guide Dr. Park and many other people. We would like to extend our heartfelt gratitude to all of them.

We are highly indebted for their guidance, constant supervision and also providing us necessary information regarding the project. Thank you for your support in helping us to completing the project.

9 References

[1] S. Lee and J. Kim, “WarningBird: Detecting suspicious URLs in Twitter stream,” in Proc. NDSS, 2012.

[2] H. Kwak, C. Lee, H. Park, and S. Moon, “What is Twitter, a social network or a news media?” in Proc. WWW, 2010.

[3] D.Antoniades,I.Polakis,G.Kontaxis,E.Athanasopoulos,S.Ioannidis, E. P. Markatos, and T. Karagiannis, “we.b: The web of short URLs,” in Proc. WWW, 2011.

[4] D. K. McGrath and M. Gupta, “Behind phishing: An examination of phisher modi operandi,” in Proc. USENIX LEET, 2008.

[5] Z.Chu,S.Gianvecchio,H.Wang,andS.Jajodia,“Whoistweeting on Twitter: Human, bot, or cyborg?” in Proc. ACSAC, 2010.

[6] G. Stringhini, C. Kruegel, and G. Vigna, “Detecting spammers on social networks,” in Proc. ACSAC, 2010.

[7] C. Grier, K. Thomas, V. Paxson, and M. Zhang, “@spam: The underground on 140 characters or less,” in Proc. ACM CCS, 2010.

[8] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, “Phi.sh/$oCiaL: the phishing landscape through short URLs,” in Proc. CEAS, 2011.

[9] F. Klien and M. Strohmaier, “Short links under attack: geographical analysis of spam in a URL shortener network,” in Proc. ACM HT, 2012.

[10] K. Lee, J. Caverlee, and S. Webb, “Uncovering social spammers: Social honeypots + machine learning,” in Proc. ACM SIGIR, 2010.

[11] A. Wang, “Don’t follow me: Spam detecting in Twitter,” in Proc. SECRYPT, 2010.

[12] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, “Detecting spammers on Twitter,” in Proc. CEAS, 2010.

[13] J.Song,S.Lee,andJ.Kim,“SpamfilteringinTwitterusingsenderreceiver relationship,” in Proc. RAID, 2011. [14] C. Yang, R. Harkreader, and G. Gu, “Die free or live hard? empiricalevaluationandnewdesignforfightingevolvingTwitter spammers,” in Proc. RAID, 2011.

[15] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary, “Towards online spam filtering in social networks,” in Proc. NDSS, 2012.

[16] J.Ma,L.K.Saul,S.Savage,andG.M.Voelker,“Beyondblacklists: Learning to detect malicious web sites from suspicious URLs,” in Proc. ACM KDD, 2009.

[17] ——, “Identifying suspicious URLs: An application of large-scale online learning,” in Proc. ICML, 2009. [18] D. Canali, M. Cova, G. Vigna, and C. Kruegel, “Prophiler: A fast filterforthelarge-scaledetectionofmaliciouswebpages,”inProc. WWW, 2011.

[19] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, “Design and evaluation of a real-time URL spam filtering service,” in Proc. IEEE S&P, 2011.

[20] C. Whittaker, B. Ryner, and M. Nazif, “Large-scale automatic classification of phising pages,” in Proc. NDSS, 2010.

[21] Capture-HPC, https://projects.honeynet.org/capture-hpc

[22] Y.-M. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, andS.King,“AutomatedwebpatrolwithStriderHoneyMonkeys: Finding web sites that exploit browser vulnerabilities,” in Proc. NDSS, 2006.

[23] M. Cova, C. Kruegel, and G. Vigna, “Detection and analysis of drive-by-download attacks and malicious JavaScript code,” in Proc. WWW, 2010.

[24] P. Eckersley, “How unique is your web browser?” in Proc. PET, 2010.

[25] A. Kapravelos, M. Cova, C. Kruegel, and G. Vigna, “Escape from monkey island: Evading high-interaction honeyclients,” in Proc. DIMVA, 2011.

Module Description:

1. **Data collection**
2. **Feature extraction**
3. **Training**
4. **Classification**

Data collection: The data collection component has two subcomponents: the collection of tweets with URLs and crawling for URL redirections. To collect tweets with URLs and their context information from the Twitter public timeline, this component uses Twitter Streaming APIs. Whenever this component obtains a tweet with a URL, it executes a crawling thread that follows all redirections of the URL and looks up the corresponding IP addresses. The crawling thread appends these retrieved URL and IP chains to the tweet information and pushes it into a tweet queue. As we have seen, our crawler cannot reach malicious landing.

URLs when they use conditional redirections to evade crawlers. However, because our detection system does not rely on the features of landing URLs, it works independently of such crawler evasions.

Feature extraction: The feature extraction component has three subcomponents: grouping of identical domains, finding entry point URLs, and extracting feature vectors.

This component monitors the tweet queue to determine whether a sufficient number of tweets have been collected. Specifically, our system uses a tweet window instead of individual tweets. When more than *w* tweets are collected (*w* is 10,000 in the current implementation), it pops *w* tweets from the tweet queue. First, for all URLs in the *w* tweets, this component checks whether they share the same IP addresses. If several URLs share at least one IP address, it replaces their domain names with a list of domains with which they are grouped.

Training: The training component has two subcomponents: retrieval of account statuses and training of the classifier. Because we use an offline supervised learning algorithm, the feature vectors for training are relatively older than feature vectors for classification. To label the training vectors, we use the Twitter account status; URLs from suspended accounts are considered malicious whereas URLs from active accounts are considered benign. We periodically update our classifier using labeled training vectors.

Classification: The classification component executes our classifier using input feature vectors to classify suspicious URLs. When the classifier returns a number of malicious feature vectors, this component flags the corresponding URLs and their tweet information as suspicious.
These URLs, detected as suspicious, will be delivered to security experts or more sophisticated dynamic analysis environments for an in-depth investigation.