Asian Journal of Information Technology 16 (7): 592-598, 2017
ISSN: 1682-3915

Securing Social Sites-Mitigating Clickjacking Attacks

Anjana Joyce Vinod and N. Harini
Department of Computer Science and Engineering, Amrita School of Engineering,
Amrita Vishwa Vidyapeetham University, Coimbatore, India

Abstract: Clickjacking attack is an emerging threat on the web. Although, tools are available for identifying clickjack attacks on web pages, complete information on attacks are not available and more over the procedure adopted by the tool to identify and handle attacks is not made visible for users. Hence, arises the need for developing an application which can aid users in collecting dataset, allow users to modify/streng then, the procedures for identifying and preventing clickjacking attack. In clickjacking attack the attacker presents a sensitive user interface to the user by making the user interface transparent and thus, the user is tricked to perform an action which is out of context. So, to mitigate the clickjacking attack a schema is introduced that consist of two phases: signature detection and in context defence. Signature detection approach determines whether the attack is detected by comparing them against a database of signatures or their pattern from the known malicious sites. By using a scoring mechanism signature detection is done. This method uses static features to identify potential malicious pages. Scoring algorithm works based on the concept of standard score which measure how standard deviation of the observed attribute is away from the mean. Using each instance, two types of scores are calculated, Foreign content score and the script content score and based on the score the web pages are classified as malicious or not. A threshold value is chosen and the group score greater than the threshold value is considered as a malicious page. In context checks whether an attack has occurred by comparing the referred bitmap and the screenshot of the current browser. Thus, a hybrid schema is introduced to mitigate the clickjacking attack. The focus of the researcher is to create an attack dataset that could be used to train the system to prevent the clickjacking attack.

Key words: Clickjacking, signature detection, scoring mechanism, hybrid schema, screenshot, approach

INTRODUCTION

Social networking platform helps us to connect with people who share their interests, real-life connection, activities and backgrounds. Social networking has become a part of humans as it provides communication with people around the world and to assist in online networking. These sites are generally communities created to support a common theme. The growth of social networking sites such as Facebook, Twitter, LinkedIn and Myspace, helps individuals to meet new people and friends by their own and to express their ideas and thoughts to the world.

The real problem that has to be taken into considered is that, most of the social media users are unaware about the various forms of attack that may occur in the social media. SNS are used by all age group people and their unawareness helps hackers to find their victims easily. The information shared in social media may contain sensitive and private information which can be easily compromised by the hackers. Due to the low awareness of how security attacks are performed the user are easily trapped into attacks like clickjacking and phishing attack. In clickjacking attack the attacker tricks the web user to click on another page other than what they want. Another name for clickjacking attack is UI readdressing. Through clickjacking the attacker can get confidential information of the user (such as passwords, pin numbers) or even get control of the user's computer (mainly web camera and microphone). This attack is vulnerable to various browsers and platforms, so, it is considered as a browser security issue. In clickjack attack the attacker embed the script or code to the user's web page without their knowledge and make the user to perform unintended actions. The attacker load their page as a transparent layer to the user, the user thinks that they are clicking on the visible button but they are actually clicking on the hidden button. The hidden page may be an authentic page so, the attackers can trick users into performing actions which the users never intended. The user will be genuinely authenticated to the hidden page so, it is difficult to trace the attack.

Corresponding Author: Anjana Joyce Vinod, Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Coimbatore, India

Literature review: Most of the research address the issue of content filtering in social networks (Rachna and Harini, 2016; Surya and Harini, 2016). However, the schemes would fail if the site is subject attacks. Research focusing on providing recommended systems based on likes/tweet collected from user are also available in the literature. Very few works addressing attack and impacts on social networks really exist. Attacks can be classified into two based on the nature, namely active attacks and passive attacks. In active attack attacker modify or delete the message during the transmission of the message. In passive attack the attacker just eavesdrop the message and do not make any modification to the message during transmission of messages. The various social media attacks are phishing attacks, spoofing attacks, DOS attacks. By phishing attack attacker can get confidential information about the user and the confidentiality of the user is lost. Spoofing attack is used to masquerade the person or address as another by falsifying the data with the purpose of unauthorized access. DOS attacks are designed to cause an interruption or suspension of services of a specific host/server by flooding it with large quantities of useless traffic or external communication requests. Other attacks in the social media are like jacking, clickjacking, bad SEO, rouge application, spammed tweets and all these attacks are active attacks.

Existing defence mechanisms: Several anti-clickjacking mechanisms have been proposed and some of them have been deployed by browsers frame busting is a technique to prevent a page from being embedded in another page. A small piece of code written in JavaScript is embedded in the page inorder to protect from the clickjacking attack (Rydstedt *et al*., 2010). When an attacker tries to embed the page with malicious script the frame busting code redirects the browser to original site, thus, the user see the protected page rather than the attacker page (Balduzzi *et al*., 2010; Sinha *et al*., 2014). The disadvantage of using the frame busting technique is that it's incompatible with the third party widgets. It does not work on the Facebook like buttons. JavaScript frame busting is unreliable can bypass frame busting using navigating browsing history.

The HTTP response header indicates whether a browser can display a page in a frame or not. Using X-frame the page cannot be embedded in other pages (Shahriar *et al*., 2013; Sinha *et al*., 2014). There are three attributes used to specify the X-frame-options and they are deny, sameorigin, allow-from. Deny prohibits the page from being displayed. SAME ORIGIN allows the page to display if it is from the same origin. ALLOW-FROM allows the pages that originate from the pre-defined origins to be displayed in a frame. In user confirmation method the user is asked to verify their clicks (Rehman *et al*., 2013). To prevent the out-of-context clicks a confirmation prompt is provided to the user so that, the target element can be checked. The disadvantage of using this method is that, it degrades the user experience while on single-click button. This method is vulnerable to the double-click attacks as the attacker can make the victim to click on both the target element and the confirmation pop ups.

The target element can be saved from the attack by randomizing the UI layout (Wondracek *et al*., 2010). By randomly placing the pay button on online sites the attacker cannot decoy the button as the random position confuses the exact location of the button. This method is not robust as the attacker can make the victim to keep on click the button until the button's location is obtained.

In context checks whether an attack has occurred by comparing the referred bitmap and the screenshot of the current browser. In this method, it tries to protect all the integrities such as pointer, temporal, visual. Visual integrity is preserved by checking the target visual integrity and comparing the OS level screenshot and bitmap of sensitive elements rendered at the time of user's action (Huang *et al*., 2012). The demerit for the system is that needs to screenshot the browser interface and then compare with the referred element everytime.

Signature detection approach determines whether the attack is detected by comparing them against a database of signatures or their pattern from the known malicious sites (Nei *et al*., 2014). There is no need to take the screenshot of the browser and compare it with the referred element, so the performance of the browser increases. Malicious and sensitive sites can be detected by comparing the websites that being redirected to the database. The demerit for this method is that all the browsers have to support, so that, the malicious website database can be obtained.

Clickjuggler is an automated tool used to check whether the defence mechanism for clickjacking is properly done. To check for the clickjacking vulnerabilities, clickjuggler performs actual attacks on web applications. Clickjuggler prepares attacker's pages to perform clickjacking for each button, link and form, manipulates the button and determines whether the clickjacking attack is successful or not.

MATERIALS AND METHODS

Layered architecture: The layered architecture contains three layers: application layer, information security layer and the communication layer. The application layer

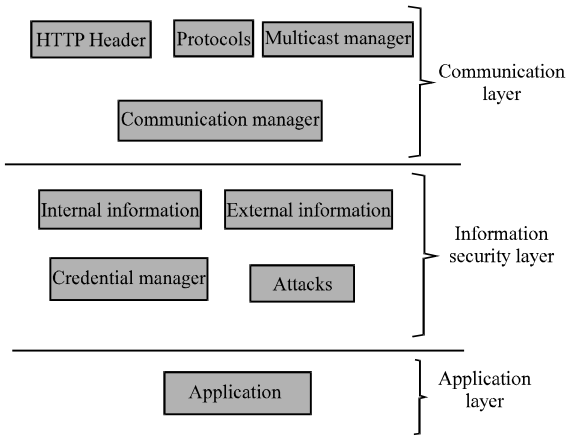

Fig. 1: Layered architecture

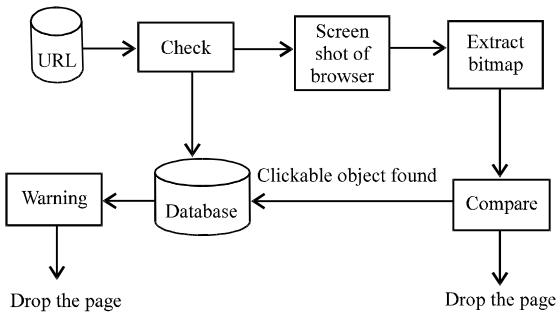

Fig. 2: Architecture diagram to mitigate the clickjacking attack

consists of any web application which the user will be using to communicate in the social media network. The information security layer consists of the attack, the credential manager, the internal information and external information. Finally, communication layer contains the protocols used for the communication, multicast manager and the communication manager (Fig. 1).

A web application can be any client-server software application which can run in a web browser. The users can use any web application according to their need. In the information security layer deals with how the information can be made secure from attacks. Attacks can be any attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset. Clickjacking attack makes the user to perform actions without their knowledge. Credential manager store credentials such as user names and passwords which are used to log on to websites or other computers on a network. Attacks mainly focus on the credential manager to get the sensitive information. The external information and the internal information are also taken by the attacker. So, security in the information layer is an important factor that has to be considered. Multicast manager provides a monitoring tool and help to verify the configurations and analyses the traffic profiles on the network.

Proposed architecture: The proposed system consists of two phases: signature detection and in context defence. In signature detection approach, determines whether the attack is detected by comparing them against a database of signatures or their pattern from the known malicious sites. In context defence checks whether an attack has occurred by comparing the referred bitmap and the screenshot of the current browser.

The architecture diagram works as follows in Fig. 2. When a URL is given it checks whether it is present in the database which contains a set of URLs that has clickable object in it. If the entered URL is present in the database a warning will be generated and the pages will be dropped. If the URL is not present in the database the screenshot of the browser is taken and bitmap extraction is done. Firstly, a screenshot of browser window based on elements position and dimensions is taken in order to do the comparison. On the other hand, reference bitmap, the position and dimensions of the sensitive element should be looks like when rendered in isolation is determined by browser. After that, the referred bitmap and screenshot of browser is compared to determine whether the sensitive element in referred bit-map is same with what user see. If a clickable object is found that URL is added into the database and the page will be dropped. If clickable objects are not found, then the page will be loaded.

Signature detection: Signature detection is done based on a scoring mechanism that uses static features to identify potential malicious pages. This mechanism is intended as a filter that allows us to reduce the number suspicious web pages requiring more expensive analysis by other mechanisms that require loading and interpretation of the web pages to determine whether they are malicious or benign. The scoring mechanism uses candidate static features of malicious web pages that are evaluate using a features election algorithm. This identifies the most appropriate set of features that can be used to efficiently distinguish between benign and malicious web pages. These features are used to construct a scoring algorithm that allows us to calculate a scorefor a web pages potential maliciousness.

The advantages of using the scoring mechanism are as follows. The first reason is that scoring mechanism scores malicious pages and acts as a filter not as a classifier. Secondly, the use of static features can be obtained without rendering web pages. But the run time features that are extracted by rendering full webpages

have more value than the partial rendered web pages. So, the static features are good for detecting malicious webpages. Scoring algorithm can reduce the number of false negative rate without the help of third party.

Feature selection: The step for feature selection is to identify the potential malicious features in the web page so, that, we can distinguish the pages as benign or malicious. The features can be classified into two groups as Foreign content and script content based on analysing the web pages. Foreign contents are malicious contents that can loaded from outside along with suspicious web pages. Malicious HTML tags such as frame tags, iframe tags, anchor tags, external links, applets and object tags are used to load web pages with Foreign content. Iframes are commonly used method to load outside malicious webpage. Third party contents such as advertising and site hit counters leads to load malicious Foreign content in the web page. The malicious contents of the malicious web page are commonly seen in the script content. The action that has to be performed by the attacker will be written in the script. The main purpose of the script code is to deliver and hide the malicious code by obfuscation. To identify some of potential malicious features from scripts which could distinguish between benign web pages and malicious web pages, the features such as number of scripts, word count, line count and character count in the web page are used.

Scoring mechanism: Scoring algorithm works based on the concept of standard score which measure how many standard deviations a value of observed attribute is far from the mean. Each instance has two types of scores based on two groups of contents of web pages. Foreign content score, script content score. A group score of instance x is calculated as follows:

$$GSg \in G(X) = \sum \frac{XA - \mu a}{\delta a} \quad (1)$$

Where:

- g = An attribute group which can be foreign content group, script content group
- a = An attribute of g
- X_a = Value of attribute a of instance X_a is a standard deviation of attribute a which is estimated during training a set of benign instances
- μa = Mean of attribute a which is estimated during training a set of benign instances

The instance X has greater score in each group, then it is classified as potential malicious class. If Tg (is chosen as a threshold for content group g in order to identify potential malicious instances, the rule of classification is as follows:

$$x = \left\{ \begin{array}{c} \text{potentially malicious if} \\ \forall g \varepsilon G : Gsg(x) > Tg \text{otherwise, x is benign} \end{array} \right\}$$

Pages will be classified as malicious or non-malicious by calculating the score using the Eq. 1 and comparing that score with the threshold value. If the group score is greater than the threshold value then the page is considered to as a malicious page.

In context defense: In this method, we are mainly focusing on how to preserve the display integrity. The original page is loaded with the attacker's page and the user is forced to click on a page which is barely visible. By using code injection method the attacker can embedded their page in the user's page.

To enforce the display integrity we take the OS level screenshot of the page and then bitmap value of the sensitive element is calculated. The comparison of the bitmap is done to check whether the clickjacking attack is present or not. If there is any difference in bitmap value then there is a chance of clickjacking attack in that page and the page will be dropped. Comparison of the bitmap is done by comparing what the user sees and at the time of the user performing some action.

The OS level screenshot is taken using the OS API, so that, the elements can be easily inspected using the APIs. After taking the screenshot of the page the sensitive area is cropped based on the positions and dimensions. Then, the bitmap value of the cropped region is extracted. Pixel wise comparison is done to compare the bitmap values. The reference bitmap which are rendered in isolation is used to compare with the bitmap value of cropped screenshot. If there is no change in the bitmap value then there is no transparent button hidden in the page. The mismatch of bitmap values indicates the presence of transparent button present in the page. Thus, clickjackingattack is detected in the page.

RESULTS AND DISCUSSION

To evaluate the scoring mechanism the various features to identify the web as malicious is taken. The features used are the number of iframe tag, number of frame tag, number of external links, number of anchor tags, number of applet tag, number of object tag, number of script tag, number of script count, number of word count and character count in the script. Using the regular expression and pattern matching the counts are taken. Regular expression for each tag is evaluated and count is incremented according to the pattern match occurs (Table 1-4).

Table 1: Value of Foreign content group of genuine page

Script	Iframe	Anchor	Link	Applet	Object	Frame
12	0	31	41	0	0	0
11	0	251	260	0	0	0
12	0	47	117	0	0	0
3	0	547	561	0	0	0
34	1	59	76	0	0	0
0	0	0	0	0	0	0
11	0	215	224	0	0	0
30	0	240	266	0	0	0
11	0	256	265	0	0	0
11	0	256	265	0	0	0
13	0	240	252	0	0	0
5	0	411	433	0	0	0
44	0	41	57	0	0	0
12	0	31	41	0	0	0
11	0	250	259	0	0	0
12	0	31	41	0	0	0
12	0	31	41	0	0	0
12	0	31	41	0	0	0
9	0	48	66	0	0	0
11	0	235	244	0	0	0
14	0	20	127	0	0	0
10	0	106	133	0	0	0
7	0	16	20	0	0	0
20	0	112	121	0	0	0
30	0	139	353	0	0	0
13	0	142	174	0	0	0
22	0	69	86	0	0	0
8	0	107	122	0	0	0

Table 2: Value of Foreign content of malicious page

Script	Iframe	Anchor	Link	Applet	Object	Frame
3	0	0	2	0	0	0
0	0	0	0	0	0	0
25	0	6	12	0	0	0
0	0	0	0	0	0	0
51	0	221	235	0	0	0
2	0	0	0	0	0	0
0	0	0	1	0	0	0
7	0	9	14	0	0	0
22	0	13	20	0	0	0
2	0	0	2	0	0	0
0	0	1	3	0	0	0
16	0	140	162	0	0	0
4	0	0	5	0	0	0
4	0	0	5	0	0	0
3	0	0	0	0	0	0
0	0	0	12	0	0	0
20	0	1	22	0	0	0
11	0	9	15	0	0	0
1	0	0	1	0	0	0
1	0	0	1	0	0	0
2	0	0	1	0	0	0
6	0	10	54	0	0	0
7	0	19	24	0	0	0
4	0	0	5	0	0	0
4	0	0	5	0	0	0
0	0	0	0	0	0	0
7	0	3	52	0	0	0
15	0	12	18	0	0	0

Feature selection: Page source of the web page is taken from the URL by writing a web crawler program. Each page source is saved to a particular location for further analysis. The dataset was taken from the PhishTank website to get the list of malicious pages and the genuine pages are chosen randomly. About 100 pages were taken for analyses and the score was calculated to set the threshold value. Collected pages were divided into two set one for testing the data and the other set to training the data. For all the web page the features are evaluated and the count is taken. The count of each attribute is written to an excel sheet inorder to analyse the mean and

Table 3: Value of script content group of genuine page

Linecount	Wordcount	Charactercount
14	16143	128877
66	382	3275
8	2347	30867
2	308	2865
267	1259	11596
0	0	0
66	347	3240
32	489	5171
66	374	3241
64	390	3383
2	358	3211
83	751	65280
14	16147	128904
66	374	3239
62	1658	20725
101	3319	41801
14	16136	128826
14	16143	128876
14	16130	128784
3	3178	29332
66	374	3240
13	2604	23143
7	251	2297
7	19	295

Table 4: Value of script content group of malicious page

Linecount	Wordcount	Charactercount
3	14	191
0	0	0
16	264	3475
0	0	0
503	2033	18608
4	61	421
4	61	421
0	0	0
4	237	2571
13	39	557
18	66	414
0	0	0
646	1664	19176
4	51	21249
9	34	262
0	0	0
492	1777	15482
232	766	6630
1	10	110
1	10	110
12	73	523
11	54	609
25	144	1357
4	51	21249

Table 5: Score calculation

Score of script tag	Score of iframe tag	Score of anchor	Score of link tag	Score of linecount	Score of wordcount	Score of charactercount	Total score of the page
-0.400854879	-0.25275764	-0.812662193	-0.894954065	-0.457927281	2.910784240	2.856053524	2.947681706
-0.476400926	-0.25275764	0.840262041	0.623905982	-0.206584463	-0.469695581	-0.510801706	-0.452075292
-0.400854879	-0.25275764	-0.692449521	-0.367861081	-0.486928375	-0.048237459	0.228822420	-2.020266535
-1.080769301	-0.25275764	3.064196464	2.711471617	-0.515929470	-0.485570399	-0.521792061	2.918849211
1.261158151	3.87561713	-0.602290017	-0.652213875	0.764952201	-0.281596085	-0.287751099	4.077876405
-0.476400926	-0.25275764	0.569783530	0.374230358	-0.206584463	-0.471414453	-0.511739907	-0.974883501
0.958973964	-0.25275764	0.757615829	0.665518586	-0.216251494	-0.467982708	-0.507906685	0.937209851
2.016618619	-0.25275764	-0.737529273	-0.783987121	-0.457927281	2.911642177	2.856777279	5.552836760
-0.400854879	-0.25275764	-0.812662193	-0.894954065	-0.206584463	-0.471414453	-0.511766713	3.550994405
-0.476400926	-0.25275764	0.832748749	0.616970548	-0.225918526	-0.196016956	-0.043041449	0.255583800
-0.400854879	-0.25275764	-0.812662193	-0.894954065	-0.037411412	0.160241020	0.521916440	-1.716482729
-0.400854879	-0.25275764	-0.812662193	-0.894954065	-0.457927281	2.909282852	2.854686431	2.944813225
-0.400854879	-0.25275764	-0.812662193	-0.894954065	-0.457927281	2.908424916	2.852515165	2.941784024
-0.62749302	-0.25275764	-0.684936229	-0.721568215	-0.457927281	-0.203523898	-0.253493356	-3.201699638
-0.476400926	-0.25275764	0.720049369	0.512939038	-0.511095954	0.129998772	0.187675601	0.310408260
-0.249762785	-0.25275764	-0.895308404	-0.298506741	0.206584463	-0.471414453	-0.511739907	-2.886074393
-0.551946973	-0.25275764	-0.249165295	-0.256894137	-0.462760797	0.006884937	0.021774843	-1.744865061
-0.778585113	-0.25275764	-0.925361572	-1.040598180	-0.491761891	-0.497795989	-0.537017725	-4.523878109
0.203513496	-0.25275764	-0.204085543	-0.340119345	-0.491761891	-0.547556284	-0.590682827	-2.223450035
0.958973964	-0.25275764	-0.001226660	1.268901344	-0.491761891	-0.546483864	-0.590200324	0.345444930
-0.325308832	-0.25275764	0.021313216	0.027458657	-0.443426734	-0.512380903	-0.523427212	-2.008529447
0.354605589	-0.25275764	-0.527157098	-0.582859535	0.267100080	-0.291033382	-0.246845531	-1.278947518
-0.703039066	-0.25275764	-0.241652003	-0.333183911	-0.390258061	-0.523319589	-0.563180132	-3.007390402
-0.174216739	-0.25275764	-0.497103930	-0.610601271	-0.472427828	-0.368247634	-0.397225763	-2.772580804

Table 6: Score of genuine pages

Score of script tag	Score of iframe tag	Score of anchor tag	Score of link tag	Score of linecount	Score of wordcount	Score of charactercount	Total score of the page
-0.461045096	0	-0.360942729	-0.424227060	-0.406301974	-0.2276262450	-0.306190121	-2.186333225
1.848326755	0	-0.236000775	-0.232252002	-0.320093477	-0.1391695660	-0.175336982	0.745473954
4.577584397	0	4.241085912	4.048791805	2.909409459	0.4867498970	0.427647351	16.691268820
-0566016544	0	0.360942729	-0.462622072	-0.399670551	-0.2109963890	-0.297025620	-2.297273904
-0.041159305	0	-0.173529798	-0.193856990	0.399670551	-0.2109963890	-0.297025620	-1.316238652
1.533412412	0	0.090235162	0.078671955	-0.399670551	-0.1487228870	-0.211357456	0.604754401
-0.566016544	0	-0.360942729	-0.424227060	-0.339987745	-0.2187805770	-0.291606610	-2.201561265
-0.356073648	0	-0.360942729	-0.366634543	3.857702928	0.3561878380	0.450279684	3.580519531
-0.461045096	0	-0.360942729	-0.462622072	-0.366513437	-0.2205497110	-0.303361079	-2.175034123
1.323469516	0	-0.340119070	0.040276943	-0.412933397	-0.1059098540	-0.185856236	0.238374017
-0.146130752	0	-0.173529798	-0.174659484	-0.333356322	5.4767680920	4.819356015	9.993304988
-0.670987991	0	-0.152706139	0.574043244	-0.339987745	-0.6887849530	0.608347407	1.232350968
-0.566016544	0	-0.360942729	-0.443424566	-0.419564820	-0.2290415520	-0.309417619	-2.433379277
-0.146130752	0	-0.360942729	-0.443424566	-0.346619168	-0.2067504690	-0.292961363	-2.216714838
-0.146130752	0	-0.152706139	-0.574043244	-0.353250591	-0.2134731770	-0.289534636	-0.581052050
-0.041159305	0	-0.034706792	-0.001881931	-0.260410671	-0.1816287720	-0.259730084	-0.710103971
-0.041159305	0	0.298471752	-0.535648233	3.347083368	6.9525793300	6.774980878	17.270660750
0.798612277	0	-0.111058821	-0.117066967	0.455782999	-0.0369136450	-0.138639132	0.850716713
-0.670987991	0	-0.360942729	-0.462622072	-0.386407705	-0.2262109380	-0.309497310	-2.416668746
-0.461045096	0	-0.360942729	-0.424227060	-0.406301974	-0.2276266245	-0.306190121	-2.186333225
-0.566016544	0	-0.360942729	-0.462622072	-0.399670551	-0.2279800720	-0.306827651	-2.324059619
-0.356073648	0	-0.360942729	-0.443424566	-0.260410671	-0.1816287720	-0.259730084	-1.862210470
0.356073648	0	-0.360942729	-0.366634543	-0.366513437	-0.2205497110	-0.303361079	-1.974075146

standard deviation. The scoring mechanism proposed will be working on the basis of how many standard deviations a value of observed attribute is far from the mean. For all the page sources the attributes are calculated and get updated in the excel sheet (Table 5 and 6).

Score calculation: The score is calculated for each of the page using Eq. 1 and the score value is also updated to the excel sheet so, that, analyses of the score and threshold value can be calculated from the score. The threshold value was taken by averaging the score of malicious and non-malicious pages. By comparing the values, average threshold value obtained was 1.5. So, the pages having score values more than 1.5 is considered to be malicious page and the <1.5 is considered as non-malicious page. By using the scoring algorithm 66.67% of URLS were correctly classified and 33.33% of URLs were incorrectly classified (Table 7).

Table 7: Threshold value calculation

Total score of malicious page	Total score of genuine page
-2.186333225	2.947681706
0.745473954	-0.452075292
16.69126882	-2.020266535
-2.297273904	2.918849211
-1.316238652	4.077876405
0.604754401	-0.974883501
-2.201561265	0.937209851
3.580519531	5.552836760
-2.175034123	-3.550994405
0.238374017	0.255583800
9.993304988	-1.716482729
1.232350968	2.944813225
-2.433379277	2.941784024
-2.216714838	-3.201699638
-0.58105205	0.310408260
-0.710103971	-2.886074393
17.27066075	-1.744865061
0.850716713	-4.523878109
-2.416668746	-2.223450035
-2.186333225	0.345444930
-2.324059619	-2.008529447
-1.86221047	-1.278947518
-1.974075146	-3.007390402
1.057668941	-0.276393430

CONCLUSION

A new scheme was introduced to mitigate the clickjacking attack by combining the signature detection and in context mechanism. Scoring method is used for signature detection. Based on the score we can classify whether a given URL is malicious or not. Display integrity is maintained by comparing the bitmap values of the screenshot. If clickjacking attack is identified in the page then the page will be dropped. Focus of our researcher was to create an attack dataset that consist of clickjacked URLs.

REFERENCES

Balduzzi, M., M. Egele, E. Kirda, D. Balzarotti and C. Kruegel, 2010. A solution for the automated detection of Clickjacking attacks. Proceedings of the 5th ACM Symposium on Information Computer and Communications Security, April 13-16, 2010, ACM, Beijing, China, ISBN:978-1-60558-936-7, pp: 135-144.

Huang, L.S., A. Moshchuk, H.J. Wang, S. Schecter and C. Jackson, 2012. Clickjacking: Attacks and defences. Proceedings of the 21st Symposium on USENIX Security, August 8-10, 2012, USENIX, Bellevue, Washington, USA., pp: 413-428.

Nei, L.C., L.Y. Cherng and M.M. Singh, 2014. A case study on Clickjacking attack and location leakage. Intl. J. Sci. Eng. Res., 5: 190-201.

Rachna, K.S. and N. Harini, 2016. A system to screen posts that minimize user frustration. Intl. J. Appl. Eng. Res., 11: 3944-3949.

Rehman, U.U., W.A. Khan, N.A. Saqib and M. Kaleem, 2013. On detection and prevention of Clickjacking attack for OSNS. Proceedings of the 11th International Conference on Frontiers of Information Technology (FIT'13), December 16-18, 2013, IEEE, Islamabad, Pakistan, ISBN:978-1-4799-2293-2, pp: 160-165.

Rydstedt, G., E. Bursztein, D. Boneh and C. Jackson, 2010. Busting frame busting: A study of Clickjacking vulnerabilities at popular sites. Proceedings of the IEEE Conference on Web 2.0 Security and Privacy (W2SP'10), July 20, 2010, IEEE, Oakland, California, USA., pp: 1-13.

Shahriar, H., V.K. Devendran and H. Haddad, 2013. ProClick: A framework for testing Clickjacking attacks in web applications. Proceedings of the 6th International Conference on Security of Information and Networks, November 26-28, 2013, ACM, Aksaray, Turkey, ISBN:978-1-4503-2498-4, pp: 144-151.

Sinha, R., D. Uppal, D. Singh and R. Rathi, 2014. Clickjacking: Existing defences and some novel approaches. Proceedings of the 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT'14), July 12-13, 2014, IEEE, Ajmer, India, ISBN:978-1-4799-3141-5, pp: 396-401.

Surya, P.K. and N. Harini, 2016. CHATGAURD: A system that ensures safe posting in social networking sites. Intl. J. Eng. Technol., 8: 587-595.

Wondracek, G., T. Holz, E. Kirda and C. Kruegel, 2010. A practical attack to de-anonymize social network users. Proceedings of the IEEE Symposium on Security and Privacy, May 16-19, 2010, Oakland, CA, USA., pp: 223-238.