AN INTELLIGENT SECURITY SYSTEM TO MAKE OUT MALICIOUS WEBPAGES

K.Hemalatha[1] | S.K.B.Rathika[2]

[1](CSE, ANNA UNIVERSITY, TAMIL NADU, hemasathya9222@gmail.com)

[2](CSE, ANNA UNIVERSITY, TAMIL NADU)

Abstract— *The general medical examination is a typical type of preventive medication including visits to a general expert by well feeling adults on a regular basis. Making out the ones taking part at risk is important for early suggestions and precautions coming between groups. The big challenge of learning the design for risk of unhealthy life in future lies in the unlabeled data which is a very integral part of the dataset which consist of the person's data who is perfectly healthy and whose condition varies from healthy to ill. In this paper, they propose a graph-based, semi-supervised learning algorithm called SHG-Health (Semi-supervised Heterogeneous Graph on Health) for risk predictions of what will take place in the future to put in order a by degrees undergoing growth place, position with the greater number or part of the facts without mark, name. Here, they will focus mainly on unlabeled data so that system will work for both undiagnosed patient and the healthy one. With this system, people will be getting intimate precaution before even dealing with a disease. Hence, this system will lead to a healthy life.*

1. INTRODUCTION

Mobile device are increasingly being used to access the web. However, in spite of significant advances in processor power and bandwidth, the browsing experience on mobile devices is considerably different. These differences can largely be attributed to the dramatic reduction of screen size, which impacts the content, functionality and layout of mobile Web Pages.

Content, functionality and layout have regularly been used to perform static analysis to determine maliciousness in the desktop space. Features such as the frequency of iframes and the number of redirections have traditionalist served as strong indicators of malicious intent. Due to the significant changes made to accommodate mobile devices, such assertions may no longer be true.

For example, whereas such behavior would be flagged as suspicious in the desktop setting, any popular benign mobile Web Pages require multiple redirections before users gain access to content previous techniques also fail to consider mobile specific webpage elements such as calls to mobile APIs. For instance, links that spawn the phone's dialer (and the reputation of the number itself) can provide strong evidence of the intent of the page. New tools are there after necessary to identify malicious pages in the mobile web.

The web attracts are the challenging issues of the web community. when the user visits the malicious web site the attack is initiated through various features(lexical, domain, path, web content and hyperlink etc).To prevent the user against accessing the malicious websites, several automated analysis and detection methods have been proposed. The attackers lure the visitor to access malicious web sites and they steal crucial information from the client machine or install the spyware for further exploits. Dynamic HTML gives attackers a new and powerful technique to compromise the security of computer system. A malicious dynamic HTML code is usually such DHTML code can disguise itself easily though obfuscation of transformation, which makes the detection even harder. Detecting and preventing the user from these attackers are significant task. A huge number of attackers have been observed in last few years. Malicious attack detection and prevention system play an immense role against these attacks by provide full protection to the system. Hence efficient detection systems are essential for web security.

2. DYNAMIC APPROACHES

Identify the malicious URLs based on dynamically extracted lexical patterns from URLs. They developed a new method to mine their URL patterns, which are not assembled using any pre-defined items and thus cannot be mined using any existing frequent pattern mining methods. it can provide new flexibility and capability malicious URLs algorithmically generated by malicious programs. Hossian shahrriar and Mohammed zulkernine proposed a tool phish tester to test the trustworthiness of the websites based on the behavior of the web application. To classify the genuine and malicious behavior of the website using state traversal. In, a fast pre filtering technique combining URL structure, host-based information and page content is proposed and is demonstrated to significantly reduce the execution load of a dynamic analysis technique. The general limitation of considering only page content is the high risk of obfuscated content (e.g., multilevel obfuscation of Java script code).M. Alexander analyzes the behavior of the website through execution to detect the malicious web content. It is, however based on evidence of malicious side-effect (the attack phase). It is an anti-malware tool that uses signatures to identify malware infections on a user's PC. The web content is executed in an isolated environment before reaching the client browser. By observing the side-effects of the execution, malicious behavior is detected in advance in a safe environment. Yinxing Xue Developed a tool to detect JavaScript malware. Deterministic Finite Automaton is used to analyze the dynamic behavior of the JS malware. The experiment results justify the efficiency the efficiency of the approach.

2.1 Issues in Dynamic Approaches

The behavior model is dynamically the malicious attack in web pages. They also have some limitation. The Finite State Machine (FSM) model uses the various states of the malicious behavior and they detect the malicious website based on their state traversals. But these approaches only detect the attacks based on predefined states (behavior). This method is not capable of detecting random inputs and new behaviors. Malicious URL is detected by dynamically mining the lexical patterns of the URL. The complete pattern set algorithms and greedy selection algorithms are used for this purpose. As the size of data set increases, the algorithms running time also increases drastically. So the existing pattern selection algorithms are not delivered a desirable performance, so better pattern selection algorithm is needed. Bottracer is a tool to detect bot like malware on end systems through detecting the start-up, preparation, and attack behavior during execution. This tool implement a prototype of Bottracer based on VMware and Windows XP professional. But if a bot first detect user activities before it launches itself, the current BotTracer would fail to detect such bots. They also fail to detect time bomb bots. Spy Proxy analyzes the behavior of the website through execution. It is not able to detect the dynamically changing malicious content. However, highly interactive web pages resemble general-purpose programs whose execution paths depend on non-deterministic factors such as randomness, time, unique system properties, or user input. An attacker could use non-determinism to evade detection. For Example, a malicious script could flip a coin to decide whether to carry out an attack, this simple scheme defeats Spy proxy 50% of the time. Guanghuim Liang at developed a classification technique to detect malicious website. A dynamic analysis is used to capture API calls and other running information of the malware. Finally a similarity comparison algorithm is used to diagnose the degree of similarity between malware variants. This method is not capable of identifying anti-detection malware.

3. ANALYSIS OF CLASSIFICATION TECHNIQUES

The traditional classification method involves a manual analysis of the contents of the page. But this approach is inappropriate in case of malicious web page classification because of vast number of pages available on the internet. Meta classification algorithm is used to analyze the suspicious web pages algorithm is sole based on the content of the web pages. The disadvantage is the website owner specifies irrelevant keywords and contents in their web page to increase their hit ratio. The machine learning algorithms are used in most of malicious web pages detection methods. The machine learning algorithm has the following drawbacks. Navie bayes classification needs big data set. When we use this classification algorithm with a small data set the precession and recall is very low. In support vector machine and decision tree algorithm, it is very difficult to update the model to take new data. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision tree within an ensemble. K-nearest neighbor (K-NN) classification is an instance-based learning algorithm that has shown to be very effective for a variety of problem domains. The key element of this scheme is the availability of a similarity measure that is capable of identifying neighbors of a particular document. A major drawback of the similarity measure used in K-NN is that it uses all features in computing distance. In many data sets, only smaller number of the total features may be useful in categorizing webpage-gram and markov chain models are used to analyze the behavior of the web site. These dynamic analysis models are effective in detecting malicious websites. Major drawback of these approaches is their computational complexity. So their real time implementation is complex. Another drawback most of these detection approaches requires the malware to be executed in virtual environment but the behavior of the malwares not similar in virtual and real environments.

4. CHALLENGES IN THE DETECTION METHODS

Most of the existing methods to detect malicious web site are based on their core techniques for a well-known attack. But the attacker invents changes in the existing approach rely on the fixed set of features but the attacker makes changes in the existing features based and also introduces new features. As a result, the detection methods are not able to detect the new attacks. So the analysis and detection technique need to be improved. The various technique like signature based, features based and behavior based approaches to detect malicious website and content are facing these limitations due to sophisticated invasions. Due to the limitations the various existing features are not sufficient to detect malicious websites. For example existing approaches are not able to detect malicious websites based on the domain because the attacker frequently changes the domain. Apart from that none of the feature collection technique is able to allocate the emerging features. The existing detection methods suffer a lot from the true and false negatives. So there is need a new approach to overcome all these limitations. The performance is a major problem. Most of the detection methods affect the performance of the system. The hybrid approaches consumes more time due to their analysis and detection phases. Most of our real time applications like financial management, health care and etc, are time critical applications. So time efficiency needs to be addressed. The emerging features, limitation of the detection method and performance are the major challenges in detecting malicious web sites. Hence these issues needs to be considered while designing a new technique.

4.1 URL Detection

The coming and the rising fame of systems, Internet, intranets and conveyed frameworks, security is getting to be one of the central purposes of exploration. Web substance is experiencing a critical change. Early pages contained basic, detached substance, while present day pages are progressively dynamic, containing implanted code, for example, ActiveX parts, JavaScript, or Flash that executes in the client's program.

The malevolent site page contains potential dangers, which is an accumulation of scripts, remote substance or a misused substance, added by an interloper to a site.

Phishing is a demonstration of sending an email to a client dishonestly asserting to be a honest to goodness business foundation trying to trick or trap the client into surrendering private data that will be utilized for fraud. It is a sort of system assault where the assailant makes an imitation of a current true blue business site to betray clients to submit individual, money related, or classified information to what do they believe is their authentic business supplier's site. It is a security assault that includes getting private and arranged information by introducing oneself as a dependable and veritable element. Numerous suspicious URL discovery plans have likewise been presented. They utilize static or element crawlers and might be executed in virtual machine honey pots, similar to Capture-HPC, Honey Monkey, and Wepawet, to inspect recently watched URLs. These plans partition URLs as indicated by a few elements involving DNS data, lexical components of URLs, URL redirection, and the HTML substance of the greeting pages. Notwithstanding, malignant servers can sidestep examination by specifically giving favorable pages to crawlers.

5. SOFTWARE DESCRIPTION

5.1 Windows XP

Windows' XP offer many new, exciting features, in addition to improvements to many features with form earlier versions to windows. Windows XP Professional makes sharing a computer easier than ever by storing personalized settings and preferences for each user.

5.1.1 Windows XP Features

XP RAP project members review individual features in Windows XP, including:

- Remote Desktop and Remote Assistance
- Power management
- Windows application compatibility
- System tools: device driver rollback, last known good configuration, and system restore
- Multi-language toolkit
- Personal firewall
- Automatic unzip feature: There is no need for expander tools such as WinZip or Aladdin Expander with Windows XP. Zipped files are automatically unzipped by Windows and placed in folders.

Managing a myriad of network and Internet connections can be confusing. Empower with knowledge about managing network and Internet connections for local and remote users. Windows XP is loaded with new tools and programs that ensure the privacy and security of data, and help to operate computer at peak performance.

5.2 Dot Net

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.".NET" is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).Programmers produce software by combining their own source code with .NET Framework and other libraries.. NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

5.3 .NET Core

.NET Core is a cross-platform free and open-source managed software framework similar to .NET Framework. It consists of Core CLR, a complete cross-platform runtime implementation of CLR, the virtual machine that manages the execution of .NET programs. Core CLR comes with an improved just-in-time compiler, called RyuJIT. .NET Core also includes Core FX, which is a partial fork of FCL. While .NET Core shares a subset of .NET Framework APIs, it comes with its own API that is not part of .NET Framework. Further, .NET Core contains CoreRT. The .NET Native runtime optimized to be integrated into AOT compiled native binaries. A variant of the .NET Core library is utilized for UWP. .NET Core's command-line interface offers an execution entry point for operating systems and provides developer services like compilation and package management. .NET Core supports four cross-platform scenarios: ASP.NET Core web apps, command-line apps, libraries, and Universal Windows Platform apps. It does not implement Windows Forms or WPF which render the standard GUI for desktop software on Windows. .NET Core is also modular, meaning that instead of assemblies, developers deal with Nugget packages. Unlike .NET Framework, which is serviced using Windows Update, .NET Core relies on its package manager to receive updates.

5.4 .NET Framework

.NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library known as Framework Class Library (FCL) and provides language interoperability each language can use code written in other languages across several programming languages. Programs written for .NET Framework executes in a software environment is known as Common Language Runtime (CLR). An application virtual machine that provides services such as security, memory management, and exception handling. (As such, computer code written using .NET Framework is called "managed code".) FCL and CLR together constitute .NET Framework.

FCL provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with .NET Framework and other libraries.

.NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

5.5 Common Language Infrastructure

Common Language Infrastructure (CLI) provides a language-neutral platform for application development and execution, including functions for exception handling, garbage collection, security, and interoperability. By implementing the core aspects of .NET Framework within the scope of CLI, this functionality will not be tied to a single language but will be available across the many languages supported by the framework. Microsoft's implementation of CLI is Common Language Runtime.

It serves as the execution engine of .NET Framework. All .NET programs execute under the supervision of CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling. Computer programs to run on CLI, they need to be compiled into Common Intermediate Language (CIL) – as opposed to being compiled into machine code. Upon execution, an architecture-specific just-in-time compiler (JIT) turns the CIL code into machine code. To improve performance, however, .NET Framework comes with Native Image Generator (NGEN), which performs ahead-of-time compilation.

5.6 Objectives of Dot Net Framework

- To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.
- To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
- Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications.

5.7 SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

SQL Server supports different data types, including primary types such as Integer, Float, Decimal, Char (including character strings), Varchar (variable length character strings), binary (for unstructured blobs of data), Text (for textual data) among others. The rounding of floats to integers uses either Symmetric Arithmetic Rounding or Symmetric Round Down (fix) depending on arguments. Microsoft SQL Server also allows user-defined composite types (UDTs) to be defined and used. It also makes server statistics available as virtual tables and views (called Dynamic Management Views or DMVs). In addition to tables, a database can also contain other objects including views, stored procedures, indexes and constraints, along with a transaction log. A SQL Server database can contain a maximum of 231 objects, and can span multiple OS-level files with a maximum file size of 260 bytes (1 Exabyte). The data in the database are stored in primary data files with an .mdf extension. Secondary data files, identified with a .ndf extension, are used to allow the data of a single database to be spread across more than one file, and optionally across more than one file system. Log files are identified with the .ldf extension.

Storage space allocated to a database is divided into sequentially numbered pages, each 8 KB in size. A page is the basic unit of I/O for SQL Server operations. A page is marked with a 96-byte header which stores metadata about the page including the page number, page type, free space on the page and the ID of the object that owns it.Page type defines the data contained in the page: data stored in the database, index, allocation map which holds information about how pages are allocated to tables and indexes, change map which holds information about the changes made to other pages since last backup or logging, or contain large data types such as image or text. While page is the basic unit of an I/O operation, space is actually managed in terms of an extent which consists of 8 pages.

A database object can either span all 8 pages in an extent ("uniform extent") or share an extent with up to 7 more objects ("mixed extent"). A row in a database table cannot span more than one page, so is limited to 8 KB in size. However, if the data exceeds 8 KB and the row contains varchar or varbinary data, the data in those columns are moved to a new page (or possibly a sequence of pages, called an allocation unit) and replaced with a pointer to the data.

5.7.1 Buffer management

SQL Server buffers pages in RAM to minimize disk I/O. Any 8 KB page can be buffered in-memory, and the set of all pages currently buffered is called the buffer cache. The amount of memory available to SQL Server decides how many pages will be cached in memory. The buffer cache is managed by the Buffer Manager. Either reading from or writing to any page copies it to the buffer cache. Subsequent reads or writes are redirected to the in-memory copy, rather than the on-disc version. The page is updated on the disc by the Buffer Manager only if the in-memory cache has not been referenced for some time. While writing pages back to disc, asynchronous I/O is used whereby the I/O operation is done in a background thread so that other operations do not have to wait for the I/O operation to complete. Each page is written along with its checksum when it is written. When reading the page back, its checksum is computed again and matched with the stored version to ensure the page has not been damaged or tampered with in the meantime.

5.7.2 Concurrency and locking

SQL Server allows multiple clients to use the same database concurrently. As such, it needs to control concurrent access to shared data, to ensure data integrity.

When multiple clients update the same data, or clients attempt to read data that is in the process of being changed by another client. SQL Server provides two modes of concurrency control: pessimistic concurrency and optimistic concurrency.

When pessimistic concurrency control is being used, SQL Server controls concurrent access by using locks. Locks can be either shared or exclusive. Exclusive lock grants the user exclusive access to the data. No other user can access the data as long as the lock is held. Shared locks are used when some data is being read multiple users can read from data locked with a shared lock, but not acquire an exclusive lock. The latter would have to wait for all shared locks to be released. Locks can be applied on different levels of granularity. On entire tables, pages, or even on a per-row basis on tables. For indexes, it can either be on the entire index or on index leaves. The level of granularity to be used is defined on a per-database basis by the database administrator. While a fine grained locking system allows more users to use the table or index simultaneously, it requires more resources. So it does not automatically turn into higher performing solution. SQL Server also includes two more lightweight mutual exclusion solutions latches and spinlocks. Which are less robust than locks but are less resource intensive. SQL Server uses them for DMVs and other resources that are usually not busy. SQL Server also monitors all worker threads that acquire locks to ensure that they do not end up in deadlocks.

SQL Server also provides the optimistic concurrency control mechanism, which is similar to the multisession used in other databases. The mechanism allows a new version of a row to be created whenever the row is updated, as opposed to overwriting the row, i.e., a row is additionally identified by the ID of the transaction that created the version of the row. Both the old as well as the new versions of the row are stored and maintained, though the old versions are moved out of the database into a system database identified. When a row is in the process of being updated, any other requests are not blocked (unlike locking) but are executed on the older version of the row. If the other request is an update statement, it will result in two different versions of the rows both of them will be stored by the database, identified by their respective transaction IDs.

6. TESTING

6.1 Aim of Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

6.2 Validation Testing

Validation is the process of evaluating a software system or component during or at the end of the development cycle in order to determine whether it satisfies specified requirements. Validation is usually associated with traditional execution-based testing, that is exercising the code with test cases. Validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

7. EXISTING SYSTEM

Existing techniques using static features of desktop webpages to detect malicious behavior do not work well for mobile specific pages. Functionality and layout have regularly been used to perform static analysis to determine maliciousness in the desktop space. Features such as the frequency of iframes and the number of redirections have traditionally served as strong indicators of malicious intent. Due to the significantly changes made to accommodate mobile devices, such assertions may no longer be true. For example, whereas such behavior would be flagged as suspicious in the desktop setting, many popular benign mobile webpages require multiple redirections before users gain access to content.

Previous techniques also fail to consider mobile specific webpage elements such as calls to mobile APIs. For instance, links that spawn the phone’s dialer (and the reputation of the number itself) can provide strong evidence of the intent of the page. New tools are therefore necessary to identify malicious pages in the mobile web.

7.1 Drawback of Existing System

We treat URL reputation as a binary classification problem where positive examples are malicious URLs and negative examples are benign URLs. This learning-based approach to the problem can succeed if the distribution of feature values for malicious examples is different from benign examples, the ground-truth labels for the URLs are correct, and the training set shares the same feature distribution as the testing set. We classify sites based only on the relationship between URLs and the lexical and host-based features that characterize them, and we do not consider two other kinds of potentially useful sources of information for features: the URL’s page content, and the context of the URL (e.g., the page or email in which the URL is embedded). Although this information has the potential to improve classification accuracy, we exclude it for a variety of reasons. First, avoiding downloading page content is strictly safer for users. Second, classifying a URL with a trained model is a lightweight operation compared to first downloading the page and then using its contents for classification. Third, focusing on URL features makes the classifier applicable to any context in which URLs are found (Web pages, email, chat, calendars, games, etc.)

a. There is no datasets are encrypted for privacy protection.
b. Detects a number of malicious mobile webpages in the wild that are not detected by existing techniques such as Google Safe Browsing and Virus Total.

Total and Google Safe Browsing. Finally, we discuss the limitations of existing tools to detect mobile malicious webpages and build a browser extension based on KAYO that provides real- time feedback to mobile browser users.

8. CONCLUSION

Mobile webpages are significantly different than their desktop counterparts in content, functionality and layout. Therefore, existing techniques using static features of desktop webpages to detect malicious behavior do not work well for mobile specific pages. We designed and developed a fast and reliable static analysis technique called KAYO that detects mobile malicious webpages. KAYO makes these detections by measuring 44 mobile relevant features from webpages, out of which 11 are newly Identified mobile specific features. KAYO provides 90% accuracy in classification, and detects a number of malicious mobile webpages in the wild that are not detected by existing techniques such as Google Safe Browsing and Virus Total. Finally, we build a browser extension using KAYO that provides real-time feedback to users. We conclude that KAYO detects new mobile specific threats such as websites hosting known fraud numbers and takes the first step towards identifying new security challenges in the modern mobile web.

REFERENCES

[1] Adrian Tang, Simha Sethumadhavan, and Salvatore Stolfo(2009),'Unsupervised Anomaly-based Malware Detection using Hardware Features'.

[2] Mamoun Alazab, Sitalakshmi Venkatraman, Paul Watters and Moutaz Alazab (2011), 'Zero-day Malware Detection based on Supervised Learning Algorithms of API call Signatures'.

[3] Marco Cova, Christopher Kruegel, and Giovanni Vigna (2012), 'Detection and Analysis of Drive-by Download Attacks and Malicious JavaScript Code'.

[4] Mohammad Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas, (2000), 'An Implementation Of Intrusion Detection System Using Genetic Algorithm'.

[5] Robert Moskovitch, Yuval Elovici, and Lior Rokach, (2008), 'Detection of unknown computer worms based on behavioral classification of the host', journal of Computers and Security, Vol. 52, pp. 4544–4566.

[6] Shi-Jinn Horng , Pingzhi Fan, Yao-Ping Chou, Yen-Cheng Chang and Yi Pan, (2008), 'A feasible intrusion detector for recognizing IIS attacks based on neural networks', journal of Computers and Security, Vol. 27, pp.84-100