

|  |  |  |
| --- | --- | --- |
| MSc Advanced Software Engineering  Research Project | | |
|  | | |
| Assignment Type: | Project proposal | |
| Submission Date: | 11/30/2016 | |
|  | | |
| Student Id | | Student Name |
| 2015101 | | K.A.S.C.Kathriarachchi |

|  |
| --- |
|  |
| Project Proposal |
| Blocking brute force attacks using binomial ladder filter in .net applications |

Student Id - 2015101

Name - K A S C Kathriarachchi

Supervisor - Mr Cassim Farook

# **Project Title**

Blocking bruit force attacks using binomial ladder filter in .net applications

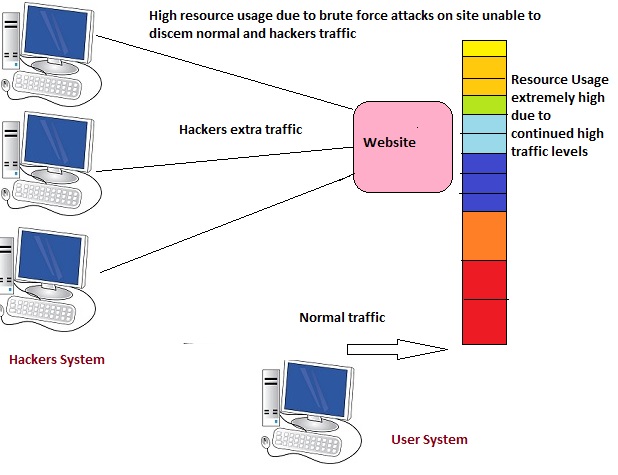
# **Project Background**

**Introduction**As we know hackers are always attempting to access our password and userid or user name. Sometimes hackers use a Brute Force attacks for hacking our information and systems. Using a Brute Force attack, hackers makes many attempts to get the user name and password until they are successful. Sometimes users use a simple password like 123456 or 098765 and the user name like admin or user. This type of password and username can be hacked easily. In a normal Brute Force attack hackers use a dictionary in which all the normal (commonly used) words and simple passwords are attempted for the user id and password to access the user account. On the other hand in a typical Brute Force attack hackers try every key combination to access the account. If the user uses a long password consisting of all possible characters like numbers, characters and special symbols then it is not easy to determine the password easily because for that a Brute Force algorithm takes a long time.

**Brute Force Attack**  
A Brute Force attack is a method or an algorithm to determine a password or user name using an automatic process that the Brute Force attack can take depending on your password length and its complexity. If you have a long and complex password then maybe a Brute Force attack cannot access your password easily. A Brute Force attack works very quickly for short passwords but if the password has a long length then it is not easy to guess the password. The main problem that occurs with a Brute Force attack is server memory problems, because the hackers attempt many times to access your account through which many HTTP requests are generated at your server so the result of this is your server goes slow and there is unnecessary traffic at our site. Sometimes the server runs out of memory. The algorithm used in a Brute Force method is a MD5 and SHA algorithm. If our website has a user authentication process, then the chances of Brute Force attack are increased at our site.

Hackers use many tools for Brute Force attacks. Using those tools there is an algorithm to guess the password automatically or create the combination depending on the given length. Sometimes we make the proxy request process but it is not very successful because we understand that every HTTP request comes from a unique IP address so it is not a good idea to block an IP address to stop a Brute Force attack. Since we know hackers are very clever, sometimes they use those types of tools that always try a different user name and different password so by blocking the IP address we cannot stop the Brute Force Attack.

**How Brute Force Attacks Work**A Brute Force Attack simply uses the cryptography algorithm. As we know hackers know that the password and user name are stored in a database. So when we attempt to login and our page request is sent from the server to the client machine hackers are more active to access the account. Hackers know that there is an encrypted key by which they can decrypt the code. So they attempt all possible combinations to unlock it. There is a computer program that runs automatically to get the password. The attackers use a high-performance computer that is very fast, in other words the computer is capable of performing a long calculation in seconds. The Brute Force attack works with a single digit, number, character and special symbol and goes high and makes the combinations and attempts to access the account until a combination does match successfully.

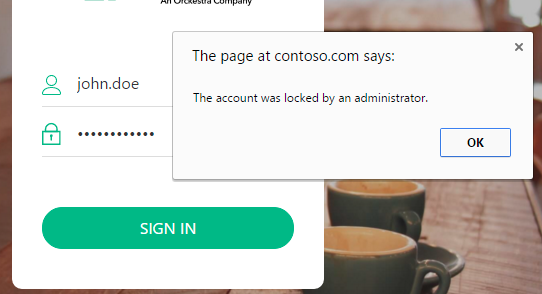


**Existing methodology to prevent Brut Force Attack**

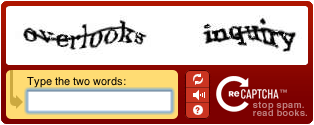
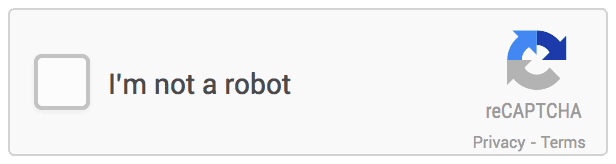
One thing is clear; a Brute Force attack is very harmful for the end users. This attack consumes memory, time and resources but if we implement some safety tips then it is not so easy for hackers to make a Brute Force attack of the user's site. Like always make a complicated and long password. Always provide a limit of login attempts for the users**.** If a user exceeds his/her attempts then block it temporarily. For example sometimes in a Gmail account when you enter the wrong password many times then Gmail sends you a CAPTCHA code and says, please fill in the given text box. Gmail does this because Gmail does not know whether the password is being submitted automatically or submitted by the user.

There is no fully secure way to block it entirely. So here I will discuss some methods for preventing a Brute Force Attack.

**Locking Account**A very famous and popular trick to prevent a Brute Force Attack is to lock the account. Locking the account means that if a user attempts a wrong password many times then the user's account will be blocked for a given time of period. The locking of the account might be for 2 hours, one day or more. The account will be opened again by the administrator. But there is the problem with account lock; sometimes the user really has forgotten his/her password. If we the lock his account then it will be problematic, so sometimes we ask them for a security answer or recovery email id. There are some drawbacks of account locking.



**CAPTCHAs Code Use**If a attacker hits your account automatically using auto-generated combinations, then a CAPTCHA code is very useful to prevent this. A CAPTCHA code is a technique by which we recognize a computer or a human. Using a CAPTCHA code we make some questions or images or numbers, the answer of which is not submitted by the computer automatically so for this we need a human for providing the right answer in the CAPTCHAs text box, if there is an incorrect answer provided then the CAPTCHA code is again send with a new code.

## **Space for Improvement**

Attackers can pick a common password and try to login to the user's account with that password. If services don't do anything to stop this attack, attackers can issue millions of guesses and compromise many accounts. Some services block user accounts after a few failed guesses, but if attackers are trying to login to all user accounts this will cause all users to be locked out. Thus, more advanced systems to prevent online-guessing attacks block IP addresses engaged in guessing, rather than the accounts targeted by guessers.

This is a implementation of an IP reputation framework. It provides two unique features not present in previous system. First, this identifies frequently-occurring passwords in failed login attempts to identify which passwords are being frequently guessed by attackers. It can provide stronger protection to users whose passwords are among those being guessed frequently, and provide faster blocking to IP addresses that guess these passwords. To detect frequently-occurring incorrect passwords, it uses a new data structure called a binomial ladder filter. Second, this is able to identify which login attempts have failed due to typos of the users' password, and be less quick to conclude that an IP that submitted the typo is guessing than for a failure that is not caused by a typo.

**About binomial ladder filter**

Binomial filters are simple and efficient structures based on the binomial coefficients for implementing Gaussian filtering. They do not require multipliers and can therefore be implemented efficiently in programmable hardware. There are many possible variations of the basic binomial filter structure, and they provide a wide range of space-time trade-offs; a number of these designs have been captured in a parameterized form and their features are compared. This technique can be used for multi-dimensional filtering, provided that the filter is separable.

Gaussian filtering is probably the most common form of linear filtering. To overcome the problem of choosing filter coefficients against a set of conflicting constraints, we review an approximation of the Gaussian based on the binomial coefficients which results in a simple, accurate and flexible architecture. This design, called a binomial filter, does not require multiplications. It is well known that, where applicable, separating a multi-dimensional filter into a cascade of one-dimensional filters significantly reduces the number of computations required to implement the filter

# **Motivation**

Compared to other domains, providing a security to other applications that make very challenging and interesting. Technology is always changing and attackers using new technologies to do attack and we need to provide updated methodologies to prevent from it. And also we need to provide better solutions than today to protect identity of the users and sensible information.

Handling a security is a huge responsibility of protecting sensible data and work with a project that make other developers life easier is challenging. Unlike in normal way of protection, the theories and methodologies are not clear cut, and therefore system needs to follow varies technologies to identify the things to protect.

# **Aim**

The aim of this project is to design and develop a Add-on/plugin to assist the other .net based applications to protect from brute force attacks.

## **Scope**

Special design to identify brute force attacks and block those from accessing the application. This will not effect to original users and they can work with out locking or entering captcha to their accounts.

# **Objectives**

## **Project and Academic Objectives**

* Prepare and submit the Terms of Reference document including aims, objectives, scope and timeline
* Carryout literature survey on
  + existing Methodologies use for protect from attacks
  + Suitable filtering techniques used for identifying attacks
  + Identifying dictionary based prediction methodologies
  + Understand frameworks that is used for handing login functionality
* Gather and analyze new requirement of the security measurement
* Select and learn tools and technologies which will be used for implementation of the prototype.
* Critically evaluate the system by conducting evaluation surveys to measure the performance and the credibility of the proposed framework.
* Design the prototype with adding binomial ladder filter.
* Design a simulator to do a brute force attack and test it with the prototype
* Critically evaluate the system with other involvement parties and design common Add-on/library package that everyone can be used
* Identify further enhancements.
* Write up a thesis that details the entire process of project development explaining how the literature survey, analysis, design and the methodologies used, how the application was implemented, tested and conclusions derived from the entire effort.

## **Personal Objectives**

* Learn and Experience the new technologies
* Gain strength in undertaking an individual research project which has a strict timeline.
* Work with different platforms and providing service to them
* Adhere and expertise the academic writing style.

**Main Components of the project**

**Add-on/plugin**

This includes the add-on/plugin for use for other projects, this will be a simple .dll library and also available as nugget package

**Sample Application**

Sample application implement with this add-on/ plugin will include in this

**Simulator**

This simulator is used to test the add-on/plugin, this will be guessing username password and try to unlock user accounts

**Project deliverables**

**Functional artefact**

* A working model of the proposed Add-on/plugin will be available for demonstration, evaluation and future enhancements.
* Simulator for testing add one
* Sample application

**Documentation**

* Project proposal
* interim project report
* draft project report
* Final project report consisting of literature survey, design, UML diagrams.

**Knowledge**

* Knowledge gained by performing the research.

# **Resource Requirements**

## **Software Requirements**

* .Net framework 4+ application development environment

## Visual studio 2015

* Microsoft Visio
* Microsoft office word/power point

**Hardware Requirement**

* Windows Laptop/Desktop
* Internet connection

**Project Plan**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **2016** | | **2017** | | | | | | | |
| **Nov** | **Dec** | **Jan** | **Feb** | **Mar** | **Apr** | **May** | **Jun** | **Jul** | **Aug** |
| Project Proposal |  |  |  |  |  |  |  |  |  |  |
| Literature review  Documentation of literature review |  | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
| Requirement specification  Requirement Gathering  Making Requirement Specification |  | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Design  Design Library package  Design Testing framework |  | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Prototype of the system |  |  |  |  |  |  |  |  |  |  |
| Interim report |  |  |  |  |  |  |  |  |  |  |
| Demonstration of prototype |  |  |  |  |  |  |  |  |  |  |
| Draft project report |  |  |  |  |  |  |  |  |  |  |
| Final system |  |  |  |  |  |  |  |  |  |  |
| Thesis |  |  |  |  |  |  |  |  |  |  |

**References**

1. Lars R. Knudsen[,](mailto:lars.r.knudsen@mat.dtu.dk) Matthew J. B. Robshaw series 2016 “Information Security and Cryptography”
2. Marc Cameron 2015 “Brute Force (Jericho Quinn Thriller Book 6) “
3. Matt Curtin 2005 “Cracking the Data Encryption Standard”

**Other Information Sources**

1. Wikipedia about brute force attacks - <https://en.wikipedia.org/wiki/Brute-force_attack>
2. Wikibooks Cryptography>Brute\_force\_attack - <https://en.wikibooks.org/wiki/Cryptography/Brute_force_attack>