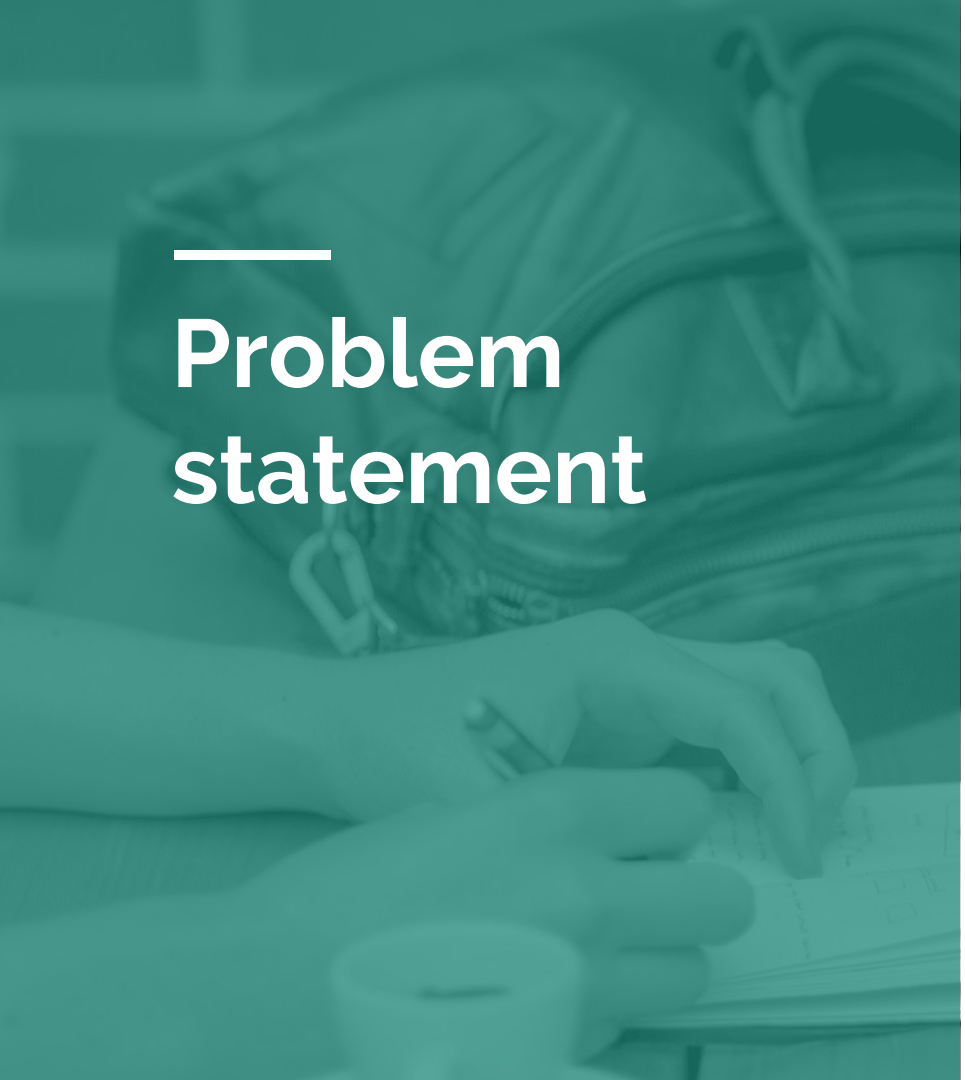
Side Channel Attack on Search boxes of Web Applications

Group members -
Pallak Srivastava (IIT2016002)
Mansi Choudhary (IIT2016016)
Shubhangi Nigam (IIT2016023)
Simran Gill (IIT2016088)
Sudhanshu (ISM2016004)

Project Supervisor -
Dr. Bibhas Ghoshal

Outline

- ➔ The Problem Statement
- ➔ Motivation
- ➔ Approach
- ➔ Next Steps
- ➔ Final Analysis



Problem statement

Attacking Search boxes of Web applications via Side Channel Attacks in order to intercept and analyse the query typed by user .

Motivation

- ➔ High level of data encryption is also susceptible to data leaks due to its physical implementation which possess a threat to all data transfers.
- ➔ A data transfer whose physical implementation is highly random tends to be more secure.
- ➔ But still there are limitations to such implementations.
- ➔ Our motive is to analyse one such random property of data transfer in case of search queries typed in a search box.
- ➔ This provides us a means of security analysis of a search engine.

Side Channel Attacks
Introduction

Side Channel Attack (SCA) is basically exploiting the information hidden in the physical implementation of a system instead of the algorithmic implementation.

In this project the data packets exchanged between the ***server*** and ***client*** serve as a side channel for analysis.

Approach

01

Analyzing Data Packets using Wireshark

- ➔ Analysed ***Application Data Packets*** based on source, destination and length fields.
- ➔ Observed a pattern in the length of data packets exchanged for a specific query.

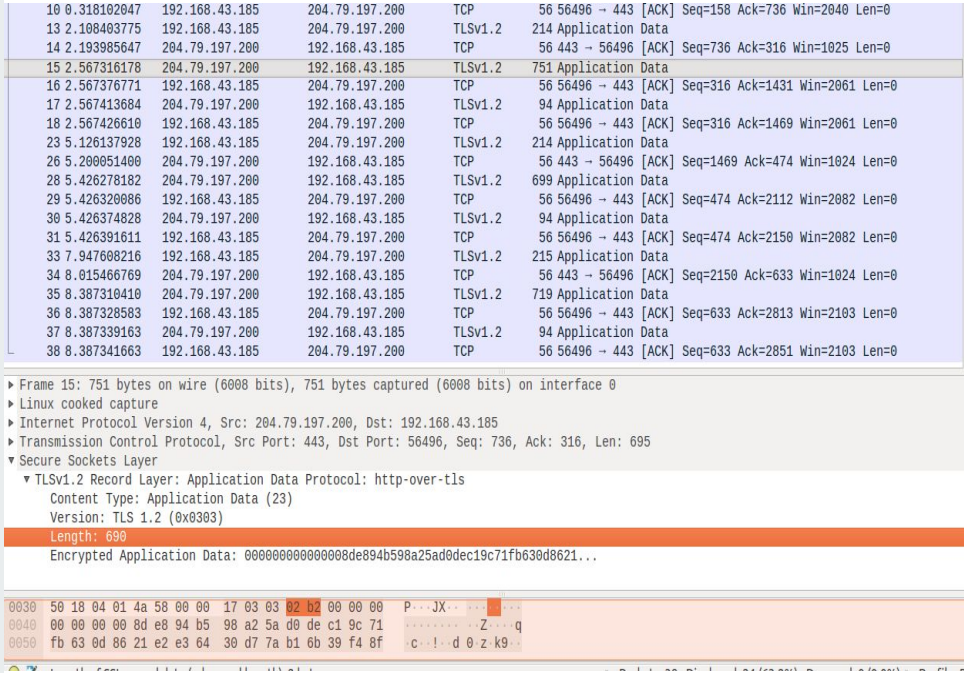


Fig1. A Wireshark window showing live capture.

02

Generating Test Strings

- ➔ Extracted valid english words of length maximum to three from dictionary .
- ➔ Generated one letter prefixes for valid two letter words.
- ➔ Generated two letter prefixes for valid three lettered words.

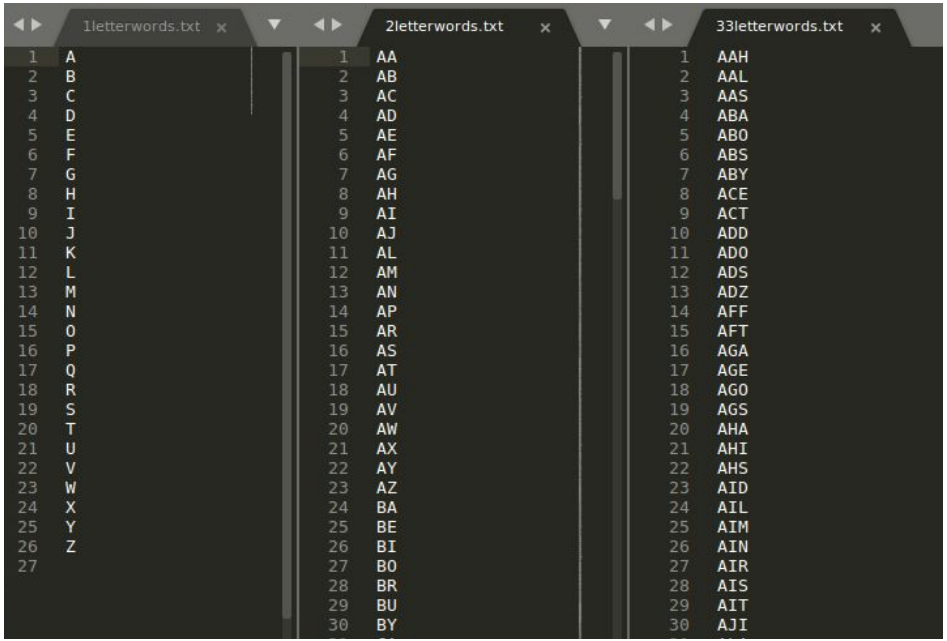


Fig2. Text files containing test strings.

03

Making a Character - Data Packet Length map

- ➔ Each character is responsible for generating some Application Data Packets whose lengths lie under a observable range.
- ➔ So a mean of the values can be mapped to each unique character.

Test Characters	Application Packet Data Lengths				
	Trial #1	Trial #2	Trial #3	Trial #4	Trial #5
A	475	664	682	668	662
B	639	645	684	643	644
C	657	660	656	655	655
D	648	645	649	650	645
E	634	630	629	633	628
F	652	650	652	649	651
G	648	644	644	649	650
H	651	646	648	646	651
I	656	652	654	651	650
J	634	636	630	630	630
K	661	660	657	660	658
L	708	708	706	707	704
M	856	630	623	625	623
N	623	625	619	619	620
O	619	615	621	617	615
P	658	653	653	656	652
Q	642	645	641	647	641
R	624	851	620	617	617
S	691	692	691	686	688
T	649	650	557,128	649	649
U	665	666	663	666	661

Next Steps

04

Generating Data Set

- ➔ Manual checking showed that each character can be mapped to an observable range of data packet lengths.
- ➔ Same could be extended to multi-character strings.

STEPWISE IMPLEMENTATION :

4.1 Automatically feeding Test Strings to the website search box.

- ➔ A Javascript snippet will iterate through files containing Test Strings and feed them to the website's search box letter by letter (at some specified interval).

4.2 Catching the concerned packets for each test string.

- ➔ A background PyShark snippet will capture the newly arrived packets as a character is fed to the search box.

4.2 Mapping strings with their packet length range.

- ➔ Making a map that stores all the test strings with their average data packet length. This is the proposed DATA-SET.

05 Finding the probable number of letters based on the sequence of packet lengths.

- ➔ For one letter typed we capture the live packet length. Propose a tolerance parameter (eg +5 and -5) and check the characters whose average packet length lie in this range.
- ➔ Select the top 10 characters in the closest proximity of observed length to generate a set.
- ➔ The same can be done for two and three letter words with the condition that the prefix should be present in the previously generated set.

Conclusion

- ➔ After generating the set we select top 5 most probable words as depicted by our approach and check whether it matches the entered query.

- ➔ Analyse the percentage of cases in which this approach is correctly predicting the query. If the percentage is high that means the search box is susceptible to data leak.

Limitations

- ➔ These should not be any spaces or backspaces in the typed query.
- ➔ Query typed is maximum of length three.
- ➔ The query must contain only valid english words.
- ➔ The wait time between typing of two characters depends on server response time.

References

- http://nsw.scrabble.org.au/backup/2&3&4%20complimentary.pdf
- https://www.lifewire.com/wireshark-tutorial-4143298
- https://thepacketgeek.com/series/intro-to-pyshark/
- https://www.javatpoint.com/document-getElementById()-method
- https://drive.google.com/open?id=0B3vCfreSb--RZFVQd0UzM3VlTjk3MkkxLTA0enR1Vk0yTlVz

Thank You !