Hacking and Information Security Projects

236349

## Contact Person:

Amichai Shulman

Mobile: 054-5885083

eMail: Shulman@imperva.com

# Secure Database Encryption with Proxy

## General Description

There are a number of existing approaches to database encryption. The easiest one to deploy is transparent encryption within the DB itself. This is a capability provided by most vendors today. The downside of this approach is that it exposes all the data to the prying eye of database administrators.

I would like to explore a method of encrypting the data in the database using a combination of encryption code in the database server and an encryption gateway that is not controlled by the database administrator.

The implementation will be based on experimental work by students in past semester that allows the introduction of temporary tokens into a database session by an intermediary proxy server and on specific capabilities of the Oracle database server.

## Requirements

Students will be required to implement an encryption system for a database based on:

* Existing proxy code that can introduce temporary tokens into a database session
* Oracle views and instead on triggers, or Oracle Transparent Database Encryption
* Encryption / Decryption code written on the proxy

The suggested system will enforce encryption of data on the database in a manner that does not allow a persistent attack by a database administrator.

## Goals

Understand data privacy concerns in a database.

Gain hands on experience with encryption protocols and the conflict with practical data access.

## Prerequisites

Knowledge of TCP/IP

Java programming knowledge

SQL knowledge

Computer Security course

Basic Cryptography course

# SPDY Security Research

## General Description

SPDY (pronounced "SPeeDY") is an application-layer protocol for transporting content over the web, designed by Google specifically for minimal latency.

In addition to a specification of the protocol, Google have developed a SPDY-enabled Google Chrome browser and open-source web server. Additionally, all Google web facing services support the SPDY protocol.

The goal of this research is to explore the SPDY protocol from the security perspective – mainly exploring vulnerabilities both design and in specific implementations, covering both client and server side. The project can be viewed as a single long term project or as a set of many separate smaller projects.

All the vulnerabilities discovered through the project will be reported back to software developers through a responsible disclosure process.

## Project Stages

* Introduction to SPDY:
	+ Learn the SPDY specification draft
	+ Create a live SPDY setup which consists of a client and server
* Learning to use tools
	+ A custom made SPDY proxy created by Imperva for the purpose of this research
	+ Wireshark plugin for SPDY
	+ HTTP interception proxies

**Students will be choosing from two project directions:**

* Server attacks
	+ Attacks based on compression tricks
	+ SPloris – Porting Slowloris and RUDY to SPDY
* Client attacks
	+ Pre-fetch oriented attacks
	+ Analyze the SPDY specification to find unhandled issues or ambiguities
	+ Suggest potential vulnerabilities based on the previous analysis
	+ Test the vulnerabilities with known implementations of SPDY

## Goals

* Contribute to the SPDY project by exposing SPDY vulnerabilities
	+ Protocol design vulnerabilities
	+ Implementation vulnerabilities – Client side (Chrome) and server side (Mod\_SPDY)
* Contribute the Security community by developing SPDY interception proxy

## Prerequisites

Knowledge of TCP/IP, HTTP and SSL

Programming knowledge

Computer Security Course

Basic Cryptography Course

## Recommended Reading

* SPDY: An experimental protocol for a faster web <http://www.chromium.org/spdy/spdy-whitepaper>
* The Chromium Projects - SPDY <http://dev.chromium.org/spdy>

# Clustering and Classification Applied to Comment Spam

## General Description

Sharing security information between organizations is becoming a growing trend in the industry and is also being applied to application security. One of the potential uses of sharing is the ability to extract high fidelity attack patterns from large set of undecided data. These patterns can then be applied back to security devices in order to prevent abuse in further systems.

We want to apply a clustering / classification approach to HTTP requests that contain URLs in their parameters in order to extract comment spam patterns.

## Requirements

* The students will gather real data for the experiment using an existing commercial sharing platform.
* The students will research different features of the data and will choose the features that best describe the data and fit the different models.
* The students will experiment with clustering models, such as K-Nearest Neighbors and Expectation Maximization. They will run experiments using Matlab and will present the results.

## Goals

* Learn and compare different clustering models.
* Come up with recommendations for features and models that best describe the data.
* Get acquainted with tools and methodologies used for security research.

## Prerequisites

Familiarity with Matlab / other computational platforms

Good understanding of HTTP

Computer Security Course

## Recommended Reading

SQL injection: <http://www.net-security.org/dl/articles/IntegrigyIntrotoSQLInjectionAttacks.pdf>

Introduction to clustering and classification:

<http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/index.html>