1231 S Dorsey Ln, Apt #203, Tempe, AZ – 85281 Naveen Kumar Tiwari
naveen.k.tiwari@gmail.com |https://www.linkedin.com/in/naveenktiwari | www.naveentiwari.com | 623-734-2016

Educational Background

Arizona State University **July 2017**
Master of Science, Computer Science *GPA 3.88/4*
Indian Institute of Information Technology, Allahabad, India **May 2012**
Bachelor of Technology, Information Technology *GPA 8.47/10*

Technical Skills

Programming Language	- C, C++, Java, Python, Shell Scripting, JavaScript, Ruby, Ruby on Rails, PHP
Framework/OS	- Hadoop, Qt, Docker, Linux
Other Skills	- Xerces, XML, JSON, GDB, Visual Studio, HTML, CSS, REST, SOAP, OWL Clingo, MATLAB
Graduate Courses	- Algorithms, Advance OS, AI, Data Mining, Software Security, Applied Cryptography

Professional Experience

Arizona State University ***August 2016 to May 2017***
Research Assistant @ SEFCOM Lab at ASU
Tally Solutions Private Limited, India ***July 2012 to July 2015***
Senior Software Engineer

- Database and database Schema for Tally's proprietary database
  - Designed and developed accumulators (computed persisted values on the disk for instant lookup) resulting in total processing time being reduced to 1sec from 20 min for 1000x1000 records.
  - Designing of database indices with speed gains of up to 30% than the standard for look-up of the records using B-Tree and forest data structure
- Design and implementation of REST protocol to make Tally product REST compliant.
- Conducted technical and functional training for new team members to help them understand the product design and code base.

Projects

- ***SDN based Honey Net (ongoing)***: Research and design a SDN based Honey Net to attract attackers and collect the associated data during the attack. The data has to later analyzed for finding exploitable vulnerability to prevent attacks on production systems. The key idea is to avoid honey pots to being fingerprinted and exploited.
- ***Security vulnerabilities in HTTP/2***: Investigating vulnerabilities in the next generation HTTP protocol that may compromise security aspects of both Client and Server. Apache HTTPD web server versions 2.4.17 through 2.4.23 did not apply limitations on request headers correctly when experimental module for the HTTP/2 protocol is used to access a resource. The net result is that a server allocates too much memory instead of denying the request. This can lead to memory exhaustion and hence Denial of Service attack on the server by a properly crafted request (URL - *http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-8740*).
- ***Intel's QuickAssist technology for hardware assisted encryption:*** Added ALPN support to OpenSSL-async library and modified nginx web server 1.10.0 to handle asynchronous behavior of OuickAssist enabled device in order to test the performance of HTTP/2 protocol.
- ***Diabetic Retinopathy Classification :*** Implementation of multiple classifiers types - Bayes, function, meta-classifiers, rules and trees using MATLAB and R to select most accurately appropriate set that included function-based (SVM, logistic regression, multilayer perceptron), tree-based and meta-classification approaches (random forest, rotation forest, random subspace). Finally using majority vote algorithm and the results from each classifier achieved an accuracy of around 78%.
- ***Web Server:*** Designed a Web Server with minimum implementation of HTTP1.1 (RFC 2616) for GET and POST request using zlib, libnet and libpcap libraries. Has a backdoor which upon deployment on a vulnerable host, remains undetected while taking full control of the host. Supports data exfiltration and trusted host discovery.
- ***Secure (RSA) communication module:*** Using the OS trust store or the trusted certificates collected by the user to establish a secure communication channel between two parties.
- ***Optical Character Recognition for Tibetan Script:*** As part of an Indian Government project, researched and developed all modules – segmentation, classification and post processing for digitizing Tibetan scripts with an accuracy of 90%. Further improvement of 5% was achieved by a dictionary module based on n-gram model and classification confidence.
- ***Flowchart to Code converter software**:* Designed and implemented a Visual C++ GUI based application to generate a C++ code from a simple flowchart input by the user.