

1231 S Dorsey Ln, Apt #203, Tempe, AZ – 85281

[naveen.k.tiwari@gmail.com](mailto:naveen.k.tiwari@gmail.com) | <https://www.linkedin.com/in/naveenktiware> | [www.naveentiwari.com](http://www.naveentiwari.com) | 623-734-2016

**Naveen Kumar Tiwari**

## Educational Background

### **Arizona State University**

*Master of Science, Computer Science*

**July 2017**

*GPA 3.88/4*

### **Indian Institute of Information Technology, Allahabad, India**

*Bachelor of Technology, Information Technology*

**May 2012**

*GPA 8.47/10*

## Technical Skills

Programming Languages - C, C++, Java, Python, Shell Scripting, JavaScript, Ruby, PHP, HTML, MATLAB, R

Framework/OS - Hadoop, Qt, Docker, Linux, Windows, MacOS-X

Other Skills - Xerces, XML, JSON, GDB, Visual Studio, CSS, REST, SOAP, OWL Clingo, Git, Wireshark, tcpdump, objdump

Graduate Courses- Algorithms, Adv. OS, AI, Data Mining, Software Security, Applied Cryptography, Network Security

## Professional Experience

**Arizona State University (Research Assistant @ SEFCOM)**

**August 2016 to May 2017**

**Tally Solutions Private Limited, India (Senior Software Engineer)**

**July 2012 to July 2015**

- Database and database Schema for Tally's proprietary database in C++
  - 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. Have implemented a prototype of HTTP and HTTPS honey-proxy in C and SSH honey-proxy in Python. The goal of the proxies is to log all the encrypted communication (MITM) after decryption and modify the response if fingerprinting attack is detected on honeypots. Currently, the logged data is being analyzed to find exploitable vulnerability to prevent attacks on production systems.
- **iCTF Hosting:** The iCTF competition is multi-site, multi-team hacking contest, hosted jointly by ASU and UCSB (<https://shellweplayagame.org/>). Contributed in writing Python-3 scripts for data extraction and monitoring the network traffic along with generating periodic tcpdump, hosted on AWS Server.
- **Security vulnerabilities in HTTP/2:** Identified 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 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>).
- **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. Implementation in C.
- **Secure (RSA) communication module:** Implemented a secure communication channel between parties using the OS trust store or the trusted certificates collected by the user. Implemented in Java.
- **Optical Character Recognition for Tibetan Script:** Researched and developed (in C) all modules – segmentation (extracting characters from image), 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. A training set generator based on Qt framework was also developed.
- **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.