

Shi Tang

Address : 2807 Cresmont Ave., 102C. Tel: 571-325-8589.

City: Baltimore, Maryland

E-mail: tangshi6666@gmail.com

Education

Johns Hopkins University, Baltimore, America

Master of Science in Information Security(GPA: 3.97)

Aug.2019-

Beijing University of Chemical Technology, Beijing, China

Bachelor of Engineering in Information Management and Information System

Sep.2015-Jun.2019

Skills & Learning

- **Programming Languages:** Python, R, C, Golang, Java, SQL
- **Tools:** Kali, Linux, WireShark, VScode, MySQL, Django, MetaSploit, VMware, R Studio, Ghidra, Burp suite, XAMPP
- **Learning:** Getting CTF training, Getting AWS solutions architect cert training, Prepare to get OSCP certification

Security Experiences

University of Austin Capture the Flag Competition(Rank top 45 of 1000 teams)

March.2020

Project Experiences

Develop and implement protocol stack similar to OSI model (Project: <https://github.com/TsGanT/Live>)

Aug.-Dec.2019

Keywords: TCP/IP, TLS protocol, C/S, OSI model, Python, Handshake, DH, AESGCM, X.509, Certification Chain, packet

- Implemented the mechanism of the **TCP/IP** and **TLS** protocols based on self-build environment similarly to OSI model.
- Implemented a client/server **interactive game** by python and used it as the application layer.
- Implemented **three-way handshake** including **Nonce**, which can ensure **Integrity**, to realize TCP protocol initialization.
- Used **asyncio** to determine the timeout and connection lost in TCP packet transform and used **packet slicing** and **hash** function to slice the application layer data into small slicing and encapsulated them into signal packet to realize TCP packet transform to ensure **Availability**.
- To implement TLS layer, I used **Diffie-Hellman algorithm** for key exchange between client and server and used **AESGCM** for data encryption and user authentication to ensure the **confidentiality**. Utilized **X.509** for signature and implemented a **Certification Chain** from professor to team member for security in order to avoid tampering with the contents of a certificate by man-in-the-middle attack.

Hacking the Parrot Bebop 1 Drone

Jan.- March.2020

Keywords: penetration test, Netdiscover, nmap, Wireshark, Nessus, DoS, ARP, Python

- Led other 5 members to do **penetration test** on a drone named Bebop Parrot and find three zero-day vulnerabilities
- Used **NetDiscover** to find the certain host under the given network and used **nmap** to find the opening ports and use **Nessus**.
- Set up cloned controllers to test the maximum number of connections that exist. Got the AR Discovery Process in MDNS by **Wireshark**.
- Implemented a **python** script which sends numerous costumed **JSON** data initializing the connection to launch **flood attack**.
- Launched **DoS ARP attack** based on python against AR Discovery Process and break the connection between drone and its controller.

Implement AES-CBC and CBC-CTR with SHA256 and HMAC and implemented Padding Oracle Attack

(project: <https://github.com/TsGanT/Golang/tree/master/src>)

Jan.-Mar.2020

Advisor: Prof. Matthew Green

Keywords: AESCBC,CBCCTR,HMAC,PADDING ORACLE ATTACK,Luck13, Golang

- Analyzed and designed the **HMAC algorithm** and implemented **HMAC-HASH256** according to RFC 4634 to defend the modify attack.
- Implemented the design of HMAC and added it to **AES-CBC** and **CBC-CTR** model by **Golang** and used **PKCS#5** as padding method.
- Divided the encryption and decryption in Client and Server and server had hard coded key in server to determine the validation of paddings.
- Implemented **Padding Oracle Attack** and **Luck13** to modify hex bits in PKCS#5 paddings by analyzed the responds from the server and used XOR operation to revers the plaintext which encrypted by AES-CBC and CBC-CTR bit by bit without secret key.

Used Metasploit shell reverse TCP with self-build payload to implement shell reverse attack

Mar.2017-Jul.2016

Keywords: Assembly coding, C, Metasploit, payload, Kali

(project: https://drive.google.com/file/d/10VYobZU8r8-sDDtX1EVbetVY_wZ3Kllu/view?usp=sharing)

- Used **assembly coding** to spawn a shell in Linux 64 and converted assembly code into shell code by **NASM** and **Objdump**.
- Implemented a C code to test this shellcode based on function pointer and used **GCC** to compile the C file into executable file to get shell.
- Modified the file named shell_reverse_tcp.rb file in **Kali Linux** by used self-build shellcode and used **msfvenom** to generate new **payload**.
- Opened **Metasploit** and used **Handler** for listening on the attack machine to get the reverse shell after the target machine which is a ubuntu VM downloaded this payload and executed it.

Design and Implementation of Customer Relationship Management System Based on Python.

Mar.- June.2019

Keywords: MySQL, Sqlite, Rights Management, RFM, Django, Python

- Established the databased based on **MySQL** and **Sqlite** to save the consume information of customers.
- Fulfilled the pro-posed functions through simulation and implemented the system function optimization, such as rights management, administrator log in/out, Customer information input, and Corporate Assets Change.
- Designed the **RFM** system, which is a system can classify different kinds of customers based on the consume record of customers. Entrepreneurs can use this system to decide who is VIP of the company.
- Completed the interface design and optimization of the system in **python** based on **Django** frame.

Honors & Awards

- “Meng-ya Cup”College Students Innovation Competition the first Prize
- “RenMin” Scholarship 3 times
- “Beijing College Student Music Festival” the second Prize