***MID TERM REPORT ON***

**Investigation Of Network Topology Poisoning Attacks in Software Defined Networking**

*A Graduate Project Report submitted to Manipal University in partial fulfilment of the requirement for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**Electronics and Communication Engineering**

*Submitted by*

**Garuda Suma Pranavi**

**Reg. No.: 130907120**

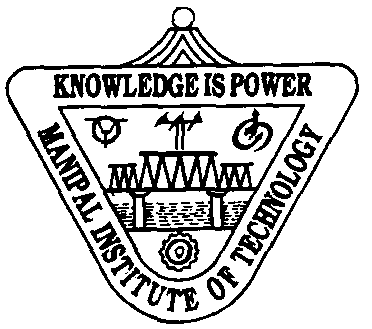
*Under the guidance of*

**Prof. Ma Maode Prof. Stanley Oswald M**

**Associate Professor & Associate Professor**

**Nanyang Technological University, Dept. of E & C Engineering**

**Singapore**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**MANIPAL INSTITUTE OF TECHNOLOGY**

(A Constituent College of Manipal University)

MANIPAL – 576104, KARNATAKA, INDIA

**APRIL 2017**

**ABSTRACT**

Software Defined Networking, a new networking paradigm has transformed the way enterprises design, build and operate networks. Network innovations in SDN are faster as they are more open, easier to program which allows operators more control, customization and optimization. Since, the technology is still immature, there are a lot of software vulnerabilities. Hence, SDN security needs to be built into the architecture, as well as delivered as a service to protect the availability, integrity, and privacy of all connected resources and information in a network. To deal with the security challenges, we first make a survey on all security issues in SDN and then we exploit one possible countermeasure against a selected network topology poisoning attack. The objective will be to design effective solutions (in the form of algorithms) against these malicious attacks.

The first two months of the internship were dedicated for a literature survey on all security issues in SDN. Then, I had to select one possible attack and work on a possible countermeasure (through formal verification tools). The algorithm is first written in BAN where we try to resolve the problems it has in reaching the goals. A more comprehensive analysis is performed on Scyther, which performs automated analysis of security protocols. In case time permits, the time taken by the algorithm will be found and optimized through MATLAB.

After the extensive literature survey, I decided to work on improving the AuthFlow protocol (Authentication and Access Control Mechanism for SDN based on host credentials). This protocol still follows MS-CHAP v2 which is very prone to man-in-the-middle and ARP injection attacks among others. Currently, I have written the protocols in BAN and am learning Scyther. I will find and prove already discovered vulnerabilities and adopt a more secure version of EAP-TLS as suggested in the future work.

Once, we adopt the secure version of TLS, we can define access control according to the privilege level of each host, by using the host identity as a new flow field to define forwarding rules and to deny access to hosts without valid credentials or revoked authorization. This is a very important step in defining security for SDN as it becomes more open and programmable as a part of the larger cloud infrastructure. The protocols are first analysed through a mathematical descriptive language called BAN (Burrows-Abadi-Needham Logic) and then automated analysis for the same is performed on Scyther.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Contents** | | | | |
|  | | | | **Page No** |
| **Chapter 1** | | | **INTRODUCTION** | 1 |
|  | | | Shortcomings in the Reference Paper | 1 |
| 1.2. | | | Project Work schedule | 1 |
| 1.3. | | | Organization of the Report | 1 |
|  | | | | |
| **Chapter 2** | | | BACKGROUND THEORY and/or LITERATURE REVIEW |  |
|  | **2.1** | | Overview of SDN | 2 |
|  | **2.2** | | DoS (Denial of Service Attacks) | 3 |
|  | **2.3** | | Firewalls/Port Knocking mechanism | 6 |
|  | **2.4** | | Stateful Failure Recovery mechanism | 7 |
|  | **2.5** | | Malicious Application mitigation/ authentication mechanism | 7 |
|  | **2.6** | | Other SDN security proposals | 8 |
|  | **2.7** | | Recommendations to deploy a safer SDN | 9 |
|  | **2.8** | | AuthFlow (protocol selected for improvement) | 10 |
|  | | | | |
| **Chapter 3** | | | **METHODOLOGY** |  |
|  | **3.1** | Introduction to BAN logic | | 12 |
|  | **3.2** | MS-CHAP v2 | | 15 |
|  | **3.3** | EAP-TLS | | 17 |
|  | **3.4** | Introduction to Scyther | | 19 |
|  | | | | |
| **Chapter 4** | | | **RESULT ANALYSIS** | 20 |
|  | | | | |
| **Chapter 5** | | | **CONCLUSION AND FUTURE SCOPE** | 21 |
|  | | | | |
| **REFERENCES** | | | | 22 |
| **PROJECT DETAILS** | | | | 28 |

**CHAPTER 1**

**INTRODUCTION**

As SDN transforms networks into an open and programmable component of the larger cloud infrastructure, the technology faces lot more security threats than the traditional networks. Hence, I have decided to work on AuthFlow, an authentication and access control mechanism which not only allows or denies access to hosts based upon their credentials but also defines privilege levels for access to network resources and transforms the forwarding of messages by using the host identity as a new flow field.

* 1. **Shortcomings in the reference paper:**

The AuthFlow paper used as reference employs MS-CHAP v2 with LDAP for verifying host credentials which is very weak in authentication and due to the rapid spread of SDN, we must make it more secure. First, we show the vulnerabilities in MS-CHAP v2 and the paper’s suggested future work EAP-TLS. Then, we try to improve upon the same by proposing a few changes and verifying the same through BAN and Scyther. The analysis done will be unique as there has been no previous research/ mathematical analysis performed for SDN security in general. (Only security modules scripted and their shortcomings guessed)

* 1. **Project Work Schedule:**

The first 2 months (Jan 26th-March 26th) have been a comprehensive literature survey on SDN and their security proposals. Then I started learning BAN and have been able to write the concerned protocols in the same. April was dedicated to analysis on BAN and making the protocols better, but due to unavailability of the assigned guidance, I have completed basic work and will refine the protocol in May. Currently I am learning Scyther and have successfully analysed some protocols (screenshots provided for reference) and will now move to analysing the concerned protocols in the same tool this month. May will then be refinement on BAN and cross-checking on Scyther which will complete the proof of my work.

* 1. **Organization of Report:**

The report is organized as follows: Chapter 1 was just introduction to the project work, Chapter 2 will be a brief insight into the already proposed solutions of SDN (2 months of survey work, which cannot be completely disclosed), recommendations proposed after analysis of these reference papers, identified shortcoming and insight into the same. Chapter 3 will be based on specifying the algorithms involved, their mathematical analysis and identifying their shortcomings.

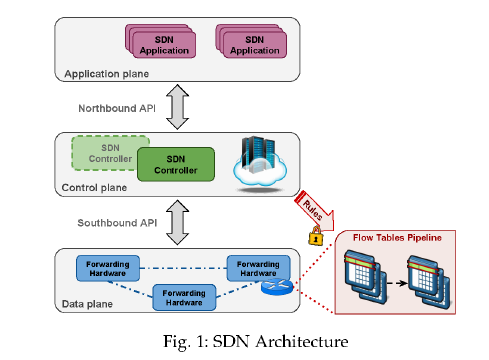
**CHAPTER 2**

**BACKGROUND THEORY**

Since the first two months have been a literature survey on SDN, security problems and their possible solutions, I will give a brief into most of the work done and a reference of the most innovative solutions. However, comprehensive analysis cannot be presented in the mid sem report due to research integrity concerns.

**2.1. Overview of SDN**

Software Defined Networking emerged as an attempt to introduce network innovations faster and to radically simplify and automate the management of large networks. In this emerging network paradigm, the control and management of the network is separated from the traffic forwarding primitives. The centralized control plane monitors the whole network and makes decisions on packet forwarding for the switches (data plane). The interface to the switches is OpenFlow which provides network administrators with a simple and uniform abstraction to the configuration of different physical or virtual multi vendor’s network devices. Specifically, the SDN controller inserts and updates traffic rules for the current traffic flows into one or more flow tables inside each switch. The resulting decoupling simplifies network monitoring, fault tolerance, security policy enforcement but it also introduces new security issues. In this report we discuss the security implications data plane programmability brings about.



The Northbound API is exposed to developers to develop simple-to-use high level abstractions devised to hide the complexity inherent in the underlying network topology and release the network administrator from the need to deal with low level network nodes configuration details. They basically allow the programmability of SDN. However, not only can they be used to develop security applications that use the network equipment as a check point for compliance, they can also constitute a privileged entering point for attackers to introduce malicious applications. Through the Southbound interface, network devices can be controlled, configured, managed and monitored by the controller so as to change their forwarding behaviour and adapt to changing business requirements and to make the network more responsive to real time traffic demands. For securing the southbound access, authentication is important. Other basic SDN review involved learning about match/action abstraction, platforms and enabling technologies such as OpenState, Fast and SDPA, programming languages such as P4, Domino, SNAP and event driven programming.

**2.2. Denial of Service (DoS) attacks**: The basic idea of DoS attacks is to overload network links or to flood a victim with a massive number of flows until it fails to serve legitimate users. Generally, there are two modes in which rules can be installed at switches: proactive and reactive. For the proactive mode, the controller first breaks down network policies into flow rules, and installs them at switches when the network bootstraps. For the reactive mode, the controller computes and installs rules only when a switch explicitly requests them. Clearly, reactive mode enables switches to quickly adapt to network dynamics and does not require switches to have large flow tables. However, the reactive rule installation also makes SDN controllers and switches vulnerable to denial of service (DoS) threats.

The Denial of Service attack has 3 main subcategories:

*2.2.1. Control channel congestion attack/ exhausting the control plane bandwidth*: Even though switches individually maintain their control channels with the controller, these logically separate channels may share some common physical links. Thus flow requests flooded by the victim switch may overwhelm common bottleneck links, and normal flow requests using those links experience congestion. In addition, if the packet buffer of the victim switch sends entire packets instead of just packet headers to the controller, it results in even higher bandwidth consumption. Compared to the switch overloading threat, which only affects those hosts connected to the victim switch, control channel congestion affects all hosts with flow requests traversing the congested links.

* + - Simple port throttling mechanisms such as Floodlight have fixed thresholds which block all flows from a certain MAC address/port punishing benign hosts too. Moreover, it cannot adapt to network dynamics. However, an improvement is also in place in form of the security enhanced floodlight which includes a digitally authenticated Northbound API, where the administrator may be asked to sign the OpenFlow application class (SE-Floodlight), or a set of actions (PermOF and OperationCheckpoint) before they start any new flows or modify old ones.
    - Additive Increase Multiplicative Decreaseis a better mechanism as it allows dynamic adjusting of the requesting rates of switches rather than temporarily blocking.
    - Resource Management through trust values allows for priority assignment while placing it in the buffer queue instead of the first come first serve mechanism. It also decides timeouts for the flows according to the current traffic in the network once it checks the number of requests from the concerned producer host.
    - Single Layer Fair Queuing polls the requests from each switch according to a weighted round robin approach. But hosts under the same switch as the attacker are punished unfairly so, the Multi-Layer Fair Queuing approach was proposed. So, initially each queue corresponds to a group of switches which is dynamically expanded to per-switch and maybe per-port queues according to the requirement and these can also be aggregated back.

*2.2.2. Controller Resource Saturation Attack***:** If the flooded flow requests arrive at the controller, they will consume the controller’s resource (CPU, memory, bandwidth etc.) for rule computation and installation. Without any protection, the controller’s resource can be saturated by the flooded requests and legitimate requests may be dropped.

* + - Avant-Guard was proposed which implements a SYN proxy (basically a connection migration tool). But this works only for SYN flooding and not against UDP or ICMP floods and it may also cause a buffer saturation attack. The LineSwitch (deployed on edge switches) overcomes this problem by proxying only the new flows to remove the possibility of SYN flooding with spoofed source IP addresses by blacklisting them.
    - The FloodGuard uses the proactive flow rule analyser and the packet migration tool to forward all table-miss packets to data plane cache. Using the real time rate of packet\_in messages, the flooding attack is identified against a certain anomaly threshold. However, the simulation performed is not practical for real time applications and the usage of a set of data plane caches for a subset of switches should be explored for scalability.
    - The controller protection protocol includes a proof of work, so that even the attacker needs to dedicate a large amount of computational resources in order to send large amounts of attack traffic, making an attack potentially prohibitively expensive.
    - The Flow Ranger combines already proposed approaches of trust value assignment (with the help of ISP’s) for each packet\_in message and for priority queues. Moreover, the message scheduling is performed according to a weighted round robin strategy.

*2.2.3. Flow table overflow attacks:* DoS attacks purposely create a large number of new flows that continuously extend and update the forwarding tables of the switches. To provide functionality even when resources or communication bandwidth are over consumed, several methods were proposed. Some versions slice the whole network into multiple logical sub-networks.

* + - One aspect is rate limiting which can be employed in various ways: by limiting the amount of traffic that a single port can send to the switch, limiting the amount of packets sent to the controller, or limiting the number of rules that the controller can insert into the switch in a short period of time.
    - Flow timeouts can be adjusted, with smaller time-to-live properties, thus making flow tables difficult to be overflowed. An eviction algorithm was also proposed which uses a dynamic event-driven triggering action such as evicting rules from flow tables and vacating some flow entries for accepting new rules. This was done on the basis of a flow checking module, which analyses every packet by validating the source IP address by querying log and adding it to the black list if it does not pass the validation or removing flows that exceed the packet per second threshold and packet bytes threshold.
    - FLIP (Fast Lightweight Policy preserving SDN Updates) combines rule replacements and additions to avoid scenarios such as blackholes and evil loops.
    - *A* peer support strategy is proposed when a switch runs out of memory space where other switches support the targeted switch by offering their idle flow table resources to install new rules and mitigate the attack. A global optimal peer supporting path and a strict rate limiting scheme will be required as part of suggested future work.
    - UDP flooding can be identified by pre-evaluating thresholds, assigning the variable state as a flooder and dropping packets if necessary. Traffic percentage triggers can be employed by comparing the outgoing traffic to the rest of the network traffic, where this threshold must not be exceeded. Amplification rate triggers can be used to keep a check on the DNS, NTP and some specific ports, as the UDP response to some commands is larger than the corresponding request.
    1. *Other DoS attack proposals:*
    - Flow management module decides timeouts and routing paths for each flow according to threat probability through intrusion detection systems. Rule aggregation modules aggregate the flow entries of malicious traffic in order to reduce the number of entries used in the switch’s TCAM’s. Monitoring modules collect multiple statistics about flows, switches and links such as flow throughput, switch TCAM usage and link bandwidth usage which helps in redirecting malicious traffic through the path having least utilized links.
    - Security planes were proposed for analysing data traffic, detecting abnormal events and triggering alarms to take necessary preventive action.
    - History based approaches calculate hash values for linearized critical events and then use a distributed verification approach such as in NetCo which propose reliable routing through unreliable routers.

**2.3. Firewalls:** They prevent unauthorized access to or from a private network. SDN can employ the centralized or distributed firewall approach. The port knocking process allows a stateful firewall to grant access to one or multiple hosts on a specific port only if they are able to successfully conclude a pre-defined sequence of connection attempts (moving target defence technology). Such ordered sequence represents a shared secret between the firewall and the hosts. However, sniffing, sequence replay attacks, port scans can identify this sequence.

* + - Advanced port knocking authentication schemes with QRC using AES was proposed where the key is shared through SMS generating an OTP which is verified through timestamps and then the sequence is encrypted through AES and compared to the generated sequence at the servers, allowing or denying access to the client by opening the port.
    - Covert communicationthrough steganography, where the encrypted sequence is hidden in the LSB’s of an image selected by the sender and sent as the payload of packets used for knocking the ports. Then the server sends an authentication packet to confirm the client’s identity providing secrecy through obscurity.
    - Methods where a pre-configured text passphrase is verified after the successful port knock sequence is used (to overcome NAT attacks) in the Secure Port Knock Tunnelling mechanism to increase the protection of the authentication process.
  + Port hopping mechanisms are also used, but synchronization is very important for these implementations. Future work can be based on port sequence selection.

**2.4. Stateful Failure recovery** (controller independent stateful link/node failure detection and recovery scheme).Restoration and protection are the most common failure recovery strategies employed. In the restoration strategy, alternate paths are installed in the flow tables after failure detection and hence network resources are dynamically allocated which delays the recovery process. However, the protection strategy has alternate backup paths preconfigured in the network.

* + - The 1:1 path protection scheme tags packets with MPLS labels upon detecting a node or link failure in order to communicate to previous nodes to use a detour path for subsequent packets (rerouting through the fast failover feature).But inconsistency and latency can be imposed by forging these tags in the absence of encryption and authentication protocols.
    - Backup resources can be computed depending on the importance level of the link- the no. of backup paths or taking the reactive recovery strategy, where the backup path is identified by VLAN tagging instead of IP matching for high speed and low memory requirements.
    - Multi Topology Routing based IP fast re-route: Multiple routing configurations provide a self-recovering SDN against single failures in the data plane by central computation of virtual topologies isolating single nodes.
  1. **Malicious application mitigation mechanisms/ Authentication mechanisms:**
     + AuthFlow is a credential based authentication and access control mechanism which uses the host identity as a new flow field to define forwarding rules and allow different levels of access to network resources according to the privilege level of each host.
     + LegoSDN was proposed to increase controller reliability in the case of SDN application failures, by proposing an isolation layer between SDN applications which support atomic updates, efficient rollbacks and a fault tolerance layer to overcome crash triggering events.
     + ROSEMARY uses a micro NOS architecture for each OpenFlow application effectively sandboxing the application to protect the control layer from any vulnerability or malicious operation of the application. It monitors and controls network resources used by each application, monitors and controls application operations and implements a safe NOS restart to improve the resilience of the control plane.
     + FRESCO facilitates the development of security applications through API’s for scripting, reusable models and a database for storage and management of session information. It defines forward, drop and group actions each using redirect, mirror and quarantine.
     + FortNox allows for efficient detection and reconciliation of potentially conflicting flow rules imposed by the dynamic OpenFlow applications through priority assignment. (OF Operator, OF Security and OF Application)
  2. **Other Security proposals:**
     + Anomaly detection mechanisms such as SPHINX use a trustworthy topology graph to detect anomalous messages from switches. (the basic idea is that messages from the controller to the switches are trustworthy whereas messages from the switches to the controller can be forced) This uses table dependency graphs to raise alerts when it detects untrusted entities triggering changes to existing flow behaviour.
     + FADE selects a small set of flows whose rule paths cover all existing rule paths. It generates a small number of dedicated flow rules and installs them in the data plane to accurately measure their flow statistics to be able to detect all forwarding anomalies.
     + To avoid host impersonation attacks, man-in-the-middle attacks and denial of service attacks, certain extensions to the SDN controller were proposed. The port manages detects the host generated traffic and contains the host list based on their MAC addresses. Host probing allows to check for the availability of a network host. The host checker verifies that no rules are violated for host migration to prevent ARP poisoning.
     + A malicious user can try to abuse the DNS messages in order to bypass the access policies and send data. DNS attacks can be prevented by keeping a track of all the resolved IP addresses through an array based global variable.
     + MTD (Moving Target Defense) technologies change network parameters irregularly over time to increase the attacker’s cost. It can be used for randomizing TCP/UDP ports, IP addresses and network paths to improve network resiliency.
  3. **Recommendations to be considered for SDN deployment today:**
     + Policy Conflict Resolution/ Network Invariant Detection: When application modules manipulate the network state, the controller should identify misconfigurations, unauthorized access, and irregularities to ensure correct functioning of the entities. Therefore, a policy conflict resolution subsystem is recommended to avoid network logic manipulation issues.
     + Mutual Authentication: SDN components should enable authentication solutions both within and across trusted domains to avoid the introduction of insecure access to network resources. This prevents data manipulation attacks, impersonation of components and ensures secure identification of network entities. (Improvement enforced through AuthFlow, which will be worked on in this project)
     + Control Plane Isolation via Slicing: Unlike network virtualization, slicing the network resources will partition the resources allocated for users sharing the infrastructure. From a security standpoint, this isolated environment can be securely instantiated and protected from unauthorized access, data manipulations and data leakages.
     + Containerized Applications: Assessing the different controller implementations, network applications are either statically compiled with the controller code, instantiated as a dynamic module with the controller software or integrated as a third party software with remote access to the controller. To prevent or restrict the impact of malicious application behaviour, it is recommended to support application containerization, which can authenticate the application during setup, control the application’s access rights on the infrastructure (bandwidth, latency, counters to monitor etc.), and limit, account for and isolate the resource usage for each application.
     + Rate-Limiting, Flow Aggregation and Short Timeouts: The correct use of flow and switch attributes such as flags, timeouts, modes of operation as well as the inherent security features such as metering to rate-limit the data flow to the control plane can ensure correct packet forwarding behaviour by avoiding overlaps, notifying flow deletes and operating securely when the connection is lost with the controller.
     + Logging/ Forensics for IDS/IPS*:* Network services and applications with monitoring capabilities require logging critical information and positively benefit from the logged information when troubleshooting and debugging the infrastructure. They should be able to determine network state at any given time and be able to track back to identify the network state at a previous point in time.

**2.8 Security proposal selected for further analysis and improvement:**

**AuthFlow** is an authentication and access control mechanism based on host credentials developed to ensure the proper operation of SDN. The main contributions are:

1. A host authentication mechanism just above the MAC layer in an OpenFlow network, which guarantees a low overhead and ensures a fine-grained access control.
2. A credential based authentication to perform an access control according to the privilege level of each host, through mapping the host credentials to the relevant set of flows.
3. A new framework for control applications, enabling Software Defined Network controllers to use the host identity as a new flow field to define forwarding rules.

The proposed mechanism applies the IEEE 802.1X standard and Extensible Authentication Protocol (EAP). EAP encapsulates authentication messages exchanged between the supplicant host and RADIUS authentication server. The translation of IEEE 802.1X messages into RADIUS packets is performed by the authenticator. The AuthFlow application allows or denies network traffic from a host depending on the result of authentication between the host and authenticator.

When a host compatible with IEEE 802.1X starts, it also starts the authentication phase by sending a start message to a reserved multicast MAC address (01:80:C2:00:00:03) with ethernet type 0x888E. The authentication procedure does not depend on any host’s prior knowledge about the network, nor on a translation of an IP address into a MAC address (avoiding address spoofing attacks) as it works just above the MAC layer.

The IEEE 802.1X packets are forwarded directly to the authenticator. The authenticator is a RADIUS client that implements 802.1X and forwards the content of EAP messages to RADIUS. The authenticator sends a confirmation message of success for POX (in this case) over a secure, encrypted and authenticated channel using SSL 3.0 and Public Key Infrastructure (PKI). As EAP allows the use of several different authentication methods, the method adopted was MS-CHAP v2 (Microsoft Challenge-Handshake Authentication Protocol) which authenticates virtual routers against a database using username and password as credentials using a Lightweight Directory Access Protocol (LDAP). It can also store other parameters that can define the privileges of the router for network access.

The AuthFlow authentication mechanism works as follows: A virtual router sends an authentication request, standardized by IEEE 802.1X, and the controller redirects it to the authenticator. Authenticator responds to it and the host sends its credentials. The authenticator checks the credentials of the supplicant against RADIUS server, running the authentication method defined in EAP. If the authentication procedure succeeds, the authenticator sends a success message to the host and a confirmation message for the controller through a SSL channel. This message identifies the supplicant by its MAC address, confirms the success of authentication and also informs the identity of the supplicant host. After authentication, our AuthFlow application allows the host to access network resources. In case of revocation of authentication, the authenticator communicates with the AuthFlow application which immediately denies the host access to the network, erasing and blocking all flow entries from and to the banned host.

The implementation proposed was RADIUS authentication against a LDAP database. However, this mechanism is considered weak in the realm of authentication and is extensible to other authentication methods (as configurators can select their own authentication and encryption methods), such as EAP-TLS or its modified versions which authenticates hosts based on certificate exchanging for access credentials.

**CHAPTER 3**

**METHODOLOGY**

This chapter describes how we propose to identify shortcomings in protocols through BAN logic. We first give an overview of the mathematical descriptive language, by stating semantics and logical inference rules. Then, I have described the 2 main protocols I will be working on: MS-CHAP v2 and EAP-TLS and their implementation in BAN. The proof of the same, cannot be sent now(due to research integrity concerns), but I will show it in my final report. However, assumptions, messages and goals for the same have been mentioned to give an indication of the work that I’m doing. Once, we identify the attacks possible, then we try to make some changes (described below) to give a better solution. The automated analysis of the same will be done in Scyther. I have given an idea of the tool and it’s working below. Next month will be dedicated for analysis on Scyther.

* 1. **BAN (Michael Burrows, Martin Abadi and Roger Needham) Logic**

Authentication protocols are the basis of security in many distributed systems, and it is therefore essential to ensure that these protocols function currently. Unfortunately, most protocols found in literature contain redundancies or security flaws. A simple logic, such as BAN has allowed the description of the beliefs of trustworthy parties involved in authentication protocols and the evolution of these beliefs as a consequence of communication.

This formal method of analysis helps us in answering the questions:

* Does this protocol work? Can it be made to work?
* Exactly what does this protocol achieve?
* Does this protocol need more assumptions than another protocol?
* Does this protocol do anything unnecessary?
  + 1. *Semantics of BAN are as follows:*
*  : P is entitled to believe X and can take it as true.
* : Someone has sent a message containing X to P, who can read and repeat X (mostly after some decryption). Therefore, P sees X.
* : The principal P at some time sent a message including the statement X. It is not known whether the message was sent long ago or during the current run of the protocol, but it is known that P believed X then.
* : P has jurisdiction over X. The principal P is an authority on X and should be trusted on this matter.
* Fresh(X): X has not been sent in a message at any time before the current run of the protocol. Usually true for nonces, which might commonly be a timestamp or a number that is used only once. The expression was invented for the purpose of being fresh.
* P Q : P and Q use the shared key K to communicate. The key K will not be discovered by any principal except P or Q or principal trusted by either P or Q.
* .  P : P has K as a public key. The matching secret key (K-1) will never be discovered by any principal except P or a principal trusted by P.
* P  Q: The formula X is a secret known only to P and Q, and possibly to principals trusted by them. Only P and Q may use X to prove their identities to one another. (such as passwords)
* **:** This represents the formula X encrypted under the key K.
* **:** This represents X combined with the formula Y, it is intended that Y be a secret and that its presence prove the identity of whoever utters . In implementations, X is simply concatenated with Y.
*  **:** P has a good public key Kp
*  **:** P has a good private key Kp-1
*  **:** X is signed with P’s private key Kp-1
* **:** X holds good in the time interval (t1,t2)
* : (t1,t2) is a good time interval
* : P is the intended recipient of X.
* : the certificate of P issued by CA.
  + 1. *Logic Postulates*

1. For private keys:



If P believes that the key is shared with Q and sees X encrypted under K, then P believes Q once said X. (For this to be sound, we must first guarantee that P did not send the message himself.

For shared secrets:



If P believes that the secret Y is shared with Q and sees  then P believes that Q once said X given that it was not uttered by P himself.

1. The nonce-verification rule expresses the check that a message is recent and hence the sender still believes in it.

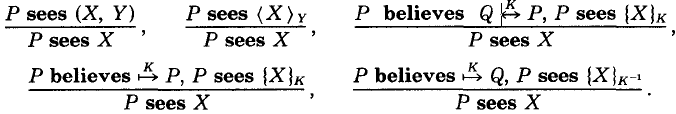


If P believes that X could have been uttered only recently and Q once said X, then P believes that Q still believes X.

1. The jurisdiction rule states that if P believes that Q has jurisdiction over X, then P trusts Q on the truth of X.

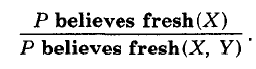


1. If a principal sees a formula, then he also sees its components, provided he knows the necessary keys:



Note that if P sees X and P sees Y, it does not follow that P sees (X,Y), since that means X and Y were uttered at the same time.

1. If one part of the formula is fresh, then the entire formula must also be fresh:



Extended BAN Inference rules:

1. Message meaning (for public key) rule:



If Q has a good public key KQ and a good private key KQ-1 and X is signed with Q’s private key, then it must have said X.

1. The see-signed message rule:



If P sees X signed with the private key of Q, then it must have seen X also.

1. The certificate duration stamp rule:



If Q has said that X holds true in (t1,t2) and (t1,t2) is a good time interval, then Q will believe in X.

* 1. **MS-CHAP v2 (Microsoft Challenge Handshake Authentication Protocol Version 2)**

MS-CHAP v2 was proposed to provide security for remote access connections. The process of authentication is described below in brief, followed by its descriptive representation:

1. The new client/ application first requests a login challenge from the authentication server.
2. The remote access server sends a challenge message to the remote access client that consists of a session identifier and a 16-byte arbitrary challenge string.
3. The remote access client sends a response message that contains the user name, a 16 byte random peer authenticator challenge, a 8 byte challenge by hashing the received challenge, peer authenticator challenge and the client’s username, a 24 byte reply by encrypting the 8 byte challenge with the MD-4 hashed version of the client’s password.
4. The server then uses the hashes of the client’s password stored in the LDAP database, to decrypt the replies. If the decrypted blocks match the challenge, the client is authenticated. The server then sends an indication of the success or failure of the connection attempt, by using the 16 byte peer authenticator challenge as well the client’s hashed password.
5. The remote access client also computes the authenticator response and if the computed response matches the received response, mutual authentication is successfully completed and the connection is used, else it is terminated.

We assume that Nb is the nonce generated by B and is fresh, Na is the nonce generated by A and is also fresh and the user password is kept secret.

Message 1: B→A: Nb

(Authenticator challenge)

Message 2: A→B: Xa, Na, H (Xa,Na,Nb), ({H (Xa,Na,Nb)}Ya1,{ H (Xa,Na,Nb)}Ya2,{H (Xa,Na,Nb)} Ya3)

(Peer Response/Challenge)

Where H (Xa,Na,Nb)= C (8 byte challenge), where H is secure hash algorithm (SHA)

Pu = User password

Ya1, Ya2, Ya3= H’ (Pu)= Ya , and H’ is Windows NT hash

Message 3: B→A: H((H(H’(H’(Pu)), ({H (Xa,Na,Nb)}Ya1,{ H (Xa,Na,Nb)}Ya2,{H (Xa,Na,Nb)} Ya3))), (Auth Response) H (Xa,Na,Nb))

The client also computes the auth response and if it matches the message received from the server, then mutual authentication is successful and the port is opened for communication, else the peer must end the session. (I’ve been having some problems in framing the goals for the same in BAN, which will be complete by next month as a guide affluent in BAN will be available).

* 1. **EAP-TLS (The Extensible Authentication mechanism - Transport Layer Security protocol)**

The EAP-TLS protocol, based on the Secure Sockets Layer allows applications to communicate securely through the exchange and verification of certificates. TLS specifies a framework that enables mutual authentication through symmetric or asymmetric encryption, negotiation of the specific encryption algorithm (the cipher suite) and a secured exchange of keys to be used for encrypting messages. After the authenticator and the peer negotiate EAP. The authenticator typically sends an EAP Request/Identity packet to the peer and the peer responds with its user ID in an EAP response/ identity packet. From now on, the conversation occurs between the authentication server and the peer and the authenticator just acts like a pass-through device which encapsulates the packets for transmission. The protocol proceeds as follows:

1. Once the EAP server receives the peer’s identity, it must respond with an EAP-TLS start packet with the start bit set and no data.
2. The peer then sends an EAP Response packet, the data field of which will contain a TLS client\_hello handshake message encapsulating the peer’s TLS version number, a session ID, a random number and a set of ciphersuites supported by the peer.
3. The EAP server responds with an EAP\_Request packet which contains a TLS server\_hello handshake message (with the TLS version number, another random number, as session ID and ciphersuite), TLS server\_certificate (which contains unique serial number of the certificate, name of issuer, validity period of the certificate, certified public key of the server and the private key of the certifying authority signs the certificate), server\_key\_exchange (the public key is sent in the message to compute the premaster secret if it wasn’t sent in the server\_certificate), certificate\_request and a server\_hello\_done message.
4. Once the peer receives the certificate request, unless it is configured for privacy it must send the client certificate, client\_key\_exchange (the client computes a premaster secret using the server\_key\_exchange and then encrypts it with the server’s public key), certificate verify, change\_cipher\_spec and TLS finished.
5. After receiving this packet, the EAP server will verify the peer’s certificate and digital signature and then send the TLS finished message. (After the client\_key\_exchange, we use a random ID in client hello, a random ID in server hello and the premaster secret to compute the master secret).

For analyzing, we first define the final goal which the protocol aims to achieve. Then, we add a logical formula for each statement of protocol to define a status after it has been performed. Finally, we verify if the goals are met through inference rules and assumptions.

Xa : Session ID

Na : Nonce, random number generated by client.

Nb : Nonce, random number generated by server.

Kca-1 : Private key of the certifying authority

Ka : Public key of the client

Kb : Public key of the server

PMS: Pre-master secret

KMS : Master secret key

Assumptions:

A   Na  B   Nb

A   Xa  B   Xa

A  A  B B  A  B

CA  CA (certifying authority believes that Kca is its public key and only trusted entities have Kca-1)

A   A B  B

A  A  B B  A  B

A   ( B) B   ( A)

Goals:

A  A  B B  A  B

A   (Kb, B) B   (Ka, A)

Protocol:

Message 1: A→B: Xa, Na

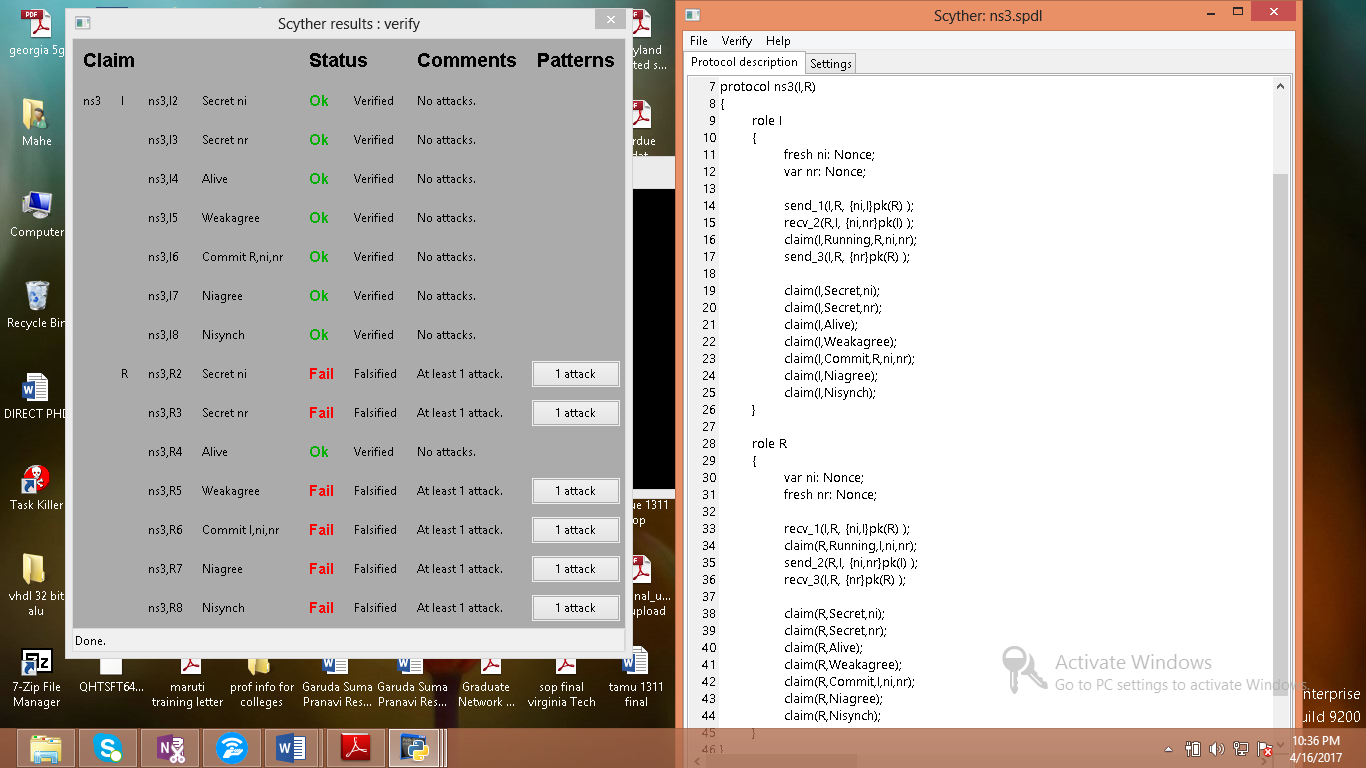
Message 2: B→A: Xa, Nb, {B, Kb}Kca-1

Message 3: A→B: {A, Ka}Kca-1, {PMS}Kb, {H(Na, Nb, B, PMS)}Kb

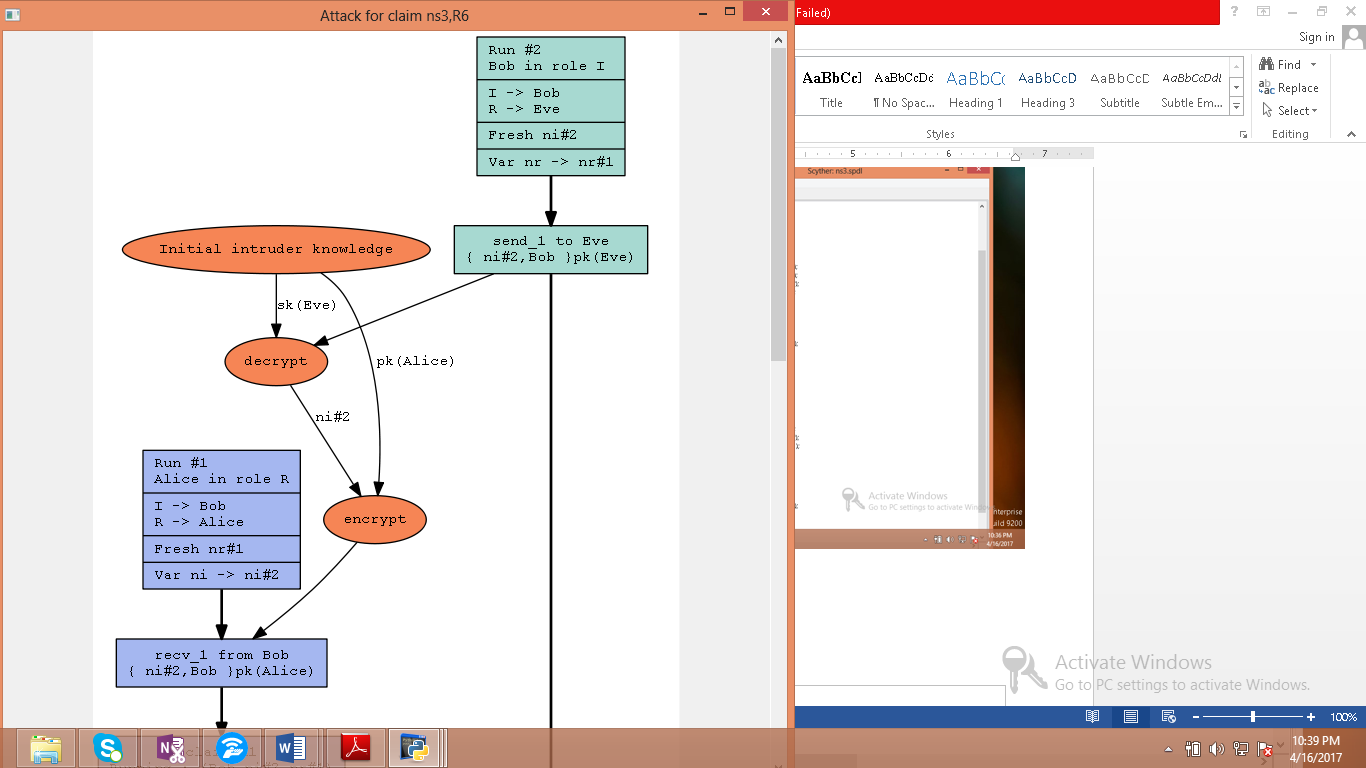
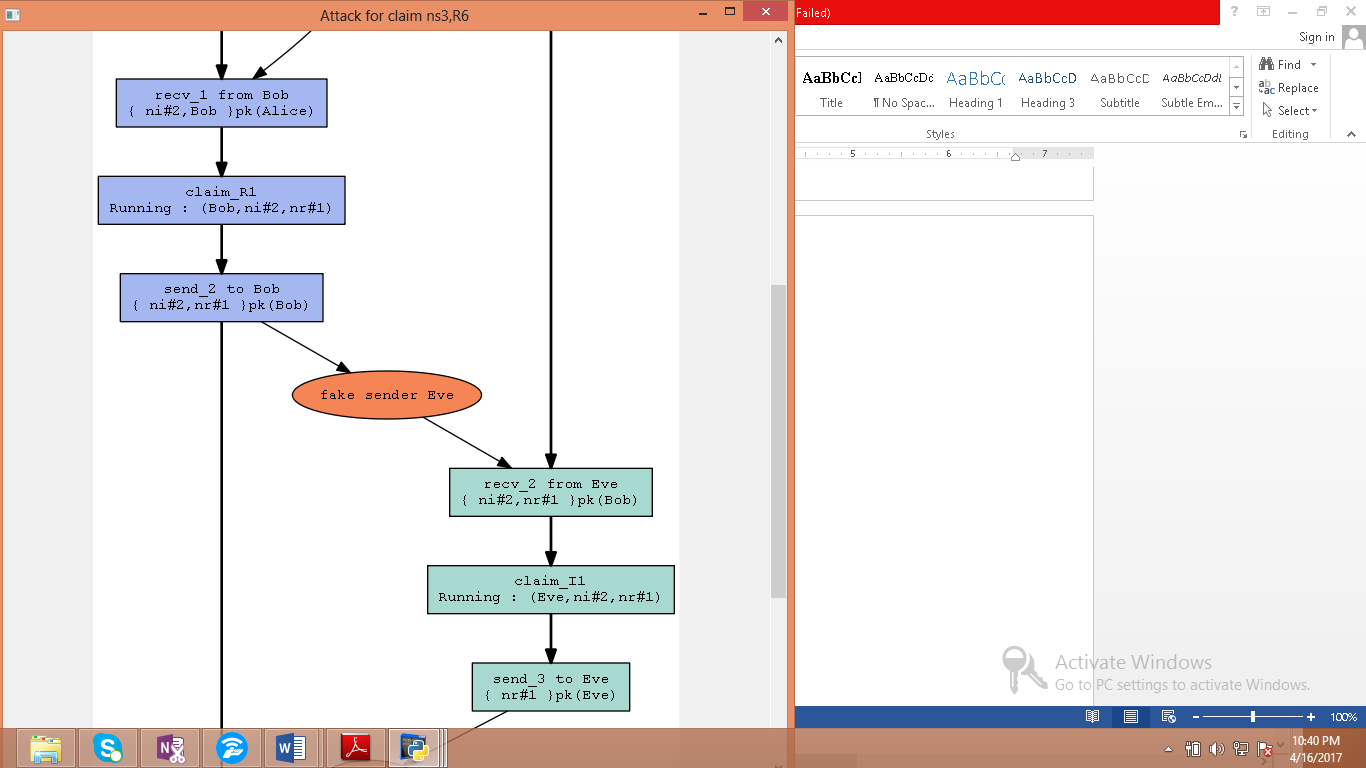
Message 4: B→A: {H(PMS, Na, Nb)}KMS

* 1. **Scyther**

Scyther assumes that all cryptographic functions are perfect: the adversary learns nothing from an encrypted message unless he knows the decryption key. However, the tool can be used to find problems that arise from the way the protocol is constructed. This language describes protocols through roles which are in turn described by events that denote the sending or receiving of terms. Scyther has semantics for defining constants, variables, symmetric and asymmetric keys, hash functions, events and claims. Shown below is a screenshot of the analysis of Needham Schroeder protocol. On the left, I have written the protocol and checked its claims on Scyther. In the right we can see, which claims fail and which encounter possible attacks.



We can further analyse each attack in the protocol by studying the attack graphs (mentioned below for the Needham Schroeder protocol- for key exchange)). Next month I would be performing similar analysis for the modified MS-CHAP v2 and EAP-TLS after BAN analysis.

**CHAPTER 4**

**RESULT ANALYSIS**

Since the challenges/nonces and username are sent in plain text in MS-CHAP v2, the root of all evil starts here. This information can be captured by a sniffer through a man-in-the-middle attack. An ARP injection attack is used to send all the traffic to the attacker which can find both the challenge and the response. There has also been a tool called asleap designed to recover weak LEAP and MS-CHAP passwords through a dictionary attack. It generates the MD4 hash from a plaintext password included in the dictionary file resulting in the NT hash. Then it encrypts the challenge 3 times with the NT hash of the password using DES and when the result is the same as our sniffed response, we can see that the password is cracked. The proof of these shortcomings has been presented through BAN (Will be specified in the final report). Moreover, MS-CHAP v2 has many redundancies such as hashing the data multiple times which does not help in improving the secrecy of exchanged content.

Hence, we improve these protocols and better, employ EAP-TLS which authorizes clients based on their signed certificated and eliminates man-in-the-middle and replay attacks. It can also dynamically generate session based and user based keys encryption keys to secure communication. However, since each user must have certificates, this places an administrative burden to distribute, revoke and manage certificates, which we’ll try to resolve.

**CHAPTER 5**

**CONCLUSION AND FUTURE SCOPE OF WORK**

The problem statement assigned was: Investigation of network topology poisoning attacks. Work till now was majorly focused on the literature survey and on identifying a solution (AuthFlow).I was then told to perform mathematical analysis of the protocols, after some research I chose BAN and Scyther for the same. Preliminary analysis of the protocols through BAN is now underway, parts of which I have mentioned above. Automated analysis through Scyther for checking these improved protocols will be the next step. An introduction into the same has also been pictorially represented, as I have started learning it. The last will be deployment of the proposed solutions in AuthFlow and checking on MATLAB if possible.

SDN will change the way networks are viewed and corporates are increasingly employing them to enable rapid innovation and also automate management of networks. AuthFlow is just one of the authentication mechanisms which can be used in SDN.

Innovative solutions which can overcome denial of service attacks, port knocking attacks and rerouting attacks are also encouraged in the research community.

AuthFlow can itself be made better by improving the protocol employed or by employing a new protocol. EAP-TLS is secure, but it does cause an administrative burden. Making one fool proof mechanism which can be employed in SDN’s throughout the globe allows for interoperability in devices which is very important in the ever expanding world of network and connectivity.

Future proposals can try to incorporate AuthFlow with FortNOX, where third party applications/users first get verified through AuthFlow and then FortNOX can decide whether they have the privilege required to make network configuration or rerouting changes as first and second level of security.

This combination of AuthFlow and FortNOX can also be applied to containerize applications for better utilization of resources according to requirements.

**References:**

1. Tooska Dargahi, Alberto Caponi, Moreno Ambrosin, Giuseppe Bianchi and Mauro Conti, “A Survey on the security of stateful SDN data planes,” IEEE Communication Surveys and Tutorials, Vol. PP, Issue 99, 2017.
2. S. Scott-Hayward, S. Natarajan, and S. Sezer, “A survey of security in software defined networks,” IEEE Communications Surveys & Tutorials, 2016.
3. Jakob Spooner, Dr. Shao Ying Zhu, “A review of solutions for SDN-exclusive security issues, International Journal of Advanced Computer Science and Applications,Vol.7, No.8 2016.
4. Jaouad Benabbou, Noureddine Idboufker, Khalid Elbaamrani, Raja Elassali, “Software-Defined Networks, Security Aspects Analysis,” 11th International Conference on Information Assurance and Security, 2015, pages 79-84.

**DoS**

1. Bin Yuan, Deqing Zou, Shui Yu, Hai Jin, Weizhong Qiang and Jinan Shen, “Defending against Flow Table Overloading Attack in Software-Defined Networks,” IEEE Transactions on Services Computing, Volume: PP, Issue 99, 2016.
2. Haopei Wang, Lei Xu and Guofei Gu, “FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks,” 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2015, pages 239-250
3. Peng Zhang, Huanzhao Wang, Chengchen Hu and Chuang Lin, “On Denial of Service Attacks in Software Defined Networks”, Network forensics and surveillance for emerging networks, Volume 30, Issue 6, 2016, pages 28-33.
4. Ali Hussein, Imad H. Elhajj, Ali Chehab and Ayman Kayssi, “SDN Security Plane: An Architecture for Resilient Security Services,” IEEE International Conference on Cloud Engineering Workshops, 2016, pages 54-59.
5. Lobna Dridi and Mohamed Faten Zhani, “SDN-Guard: DoS Attacks Mitigation in SDN Networks,” 5th IEEE International Conference on Cloud Networking, 2016, pages 212-217.
6. Tilman Wolf and Jingrui Li ,“Denial-of-Service Prevention for Software-Defined Network Controllers,” 25th International Conference on Computer Communication and Networks, 2016, pages 1-10.
7. Heng Cui, Ghassan O. Karame, Felix Klaedtke, and Roberto Bifulco, “On the Fingerprinting of Software-Defined Networks,” IEEE Transactions on Information forensics and security, vol.11, No.10, 2016, pages 2160-2173.
8. A. Wang *et al.*, “Scotch: Elastically Scaling Up SDN Control- Plane Using Vswitch Based Overlay,” Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies, 2014, pages 403-414.
9. “Floodlight Project,” <http://www.projectfloodlight.org/floodlight/>.
10. D. K. Yau *et al.*, “Defending Against Distributed Denial-of-Service Attacks with Max-Min Fair Server-Centric Router throttles,” IEEE/ACM Transactions on Networking, vol. 13, Issue 1, 2005, pages 29–42.
11. S. Shin *et al.*, “AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks,” Proceedings of the 2103 ACM SIGSAC conference on Computer and Communication Security, 2013, pages 413-424.
12. Liron Schiff, Stefan Schmid2, “Study the past if you would define the future: Implementing Secure Multi-Party SDN Updates,” IEEE International Conference on Software Science, Technology and Engineering, 2016, pages 111-116.
13. Moreno Ambrosin, Mauro Conti, Fabio De Gaspari and Radha Poovendran, “LineSwitch: Tackling Control Plane Saturation Attacks in Software-Defined Networking,” IEEE/ACM transactions on networking, Vol. PP, Issue 99, 2016, pages 1-14.
14. Trung V. Phan, Truong Van Toan, Dang Van Tuyen, Truong Thu Huong and Nguyen Huu Thanh, “OpenFlowSIA: An Optimized Protection Scheme for Software-Defined Networks from Flooding Attacks,” IEEE Sixth International Conference on Communications and Electronics, 2016, pages 13-18.
15. Ying Qian, Wanqing You and Kai Qian, “OpenFlow Flow Table Overflow Attacks and Countermeasures,” European Conference on Networks and Communications, 2016, pages 205-209.
16. Amir Shoeb and Dr T. Chithralekha,  **“**Resource Management of Switches and Controller During Saturation Time to Avoid DDoS in SDN,”2nd IEEE International Conference on Engineering and Technology (ICETECH), 2016, pages 152-157.
17. Ricardo Macedo, Rafael de Castro, Aldri Santos, Yacine Ghamri-Doudane and Michele Nogueira, “Self-Organized SDN Controller Cluster Conformations Against DDoS Attacks Effects”, IEEE Global Communications Conference (GLOBECOM), 2016, pages 1-6.
18. Liancheng Zhang, Qiang Wei, Kejun Gu, Huiqiang Yuwen, “Path Hopping Based SDN Network Defense Technology”, 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, 2016, pages 2058-2063.
19. Osamah Ibrahiem Abdullaziz, Yua-Jia Chen, Li-Chun Wang, “Lightweight Authentication Mechanism for Software Defined Network using Information Hiding,” IEEE Global Communications Conference, 2016, pages 1-6.

**Port Knocking Attack**

1. Liancheng Zhang, Yi Guo, Huiqiang Yuwen and Yu Wang, “A Port Hopping Based DoS Mitigation Scheme in SDN Network,”12th International Conference on Computational Intelligence and Security, 2016, pages 314-317.
2. Juan Wang, Yong Wang, Hongxin Hu, Qingxin Sun, He Shi and Longjie Zeng, “Towards a security enhanced Firewall application for OpenFlow Networks”, Cyberspace Safety and Security, 5th International Symposium, Vol. 8300, 2013, pages 92-103.
3. Vikas Srivastava, Alok Kumar Keshri, Abhishek Dutta Roy, Vijay Kumar Chaurasiya and Rahul Gupta, “Advanced Port Knocking Authentication Scheme with QRC using AES”, International Conference on Emerging Trends in Networks and Computer Communications, 2011, pages 159-163.
4. Mariam Khader, Ali Hadi and Amjad Hudaib, “Covert Communication Using Port Knocking,” Cybersecurity and Cyberforensics Conference, 2016, pages 22-27.
5. Fakariah Hani Mohd Ali, Rozita Yunos and Mohd Azuan Mohamad Alias, “Simple Port Knocking Method Against TCP Replay Attack and Port Scanning”, International Conference on Cyber Security, Cyber Warfare and Digital Forensic, 2012, pages 247-252.
6. Mehran Pourvahab, Reza Ebrahimi Atani and Laleh Boroumand, “SPKT: Secure Port Knock-Tunneling, an Enhanced Port Security Authentication Mechanism,” IEEE Symposium on Computers & Informatics, 2012, pages 145-149.

**Rerouting attacks/ fast failure recovery**

1. Shujuan Zhang, Ying Wang, Qichao He, Jinke Yu, Shaoyong Guo, “Backup-Resource Based Failure Recovery Approach in SDN Data Plane,” 18th Asia-Pacific Network Operations and Management Symposium, 2016, pages 1-6.
2. Pankaj Thorat, Rajesh Challa, S. M. Raza, Dongsoo S. Kim and Hyunseung Choo, “Proactive Failure Recovery Scheme for Data Traffic in Software Defined Networks”, IEEE NetSoft Conference and Workshops, 2016, pages 219-225.
3. Hengtong Li, Qing Li, Yong Jiang, Ting Zhang and Lei Wang, “A Declarative Failure Recovery System in Software Defined Networks,” IEEE International Conference on Communications, QoS, Reliability and Modeling Symposium, 2016, pages 1-6.
4. Hung-Chuan Wei, Yung-Hao Tung and Chia-Mu Yu, “Counteracting UDP Flooding Attacks in SDN”, IEEE NetSoft Conference and Workshops, 2016, pages 367-371.
5. Laura Mutu, Rania Saleh and Ashraf Matrawy, “Improved SDN Responsiveness to UDP Flood Attack,” IEEE Conference on Communications and Network Security, 2015, pages 715-716.
6. Thuy Vinh Tran and Heejune Ahn, “A Network Topology-aware Selectively Distributed Firewall Control in SDN”, International Conference on Information and Communication Technology, 2015, pages 89-94.
7. Selcuk Cevher, Mustafa Ulutas, Sedat Altun, Ibrahim Hokelek, “Multi Topology Routing Based IP Fast Re-Route for Software Defined Networks,” IEEE Symposium on Computers and Communication (ISCC), 2016, pages 1224-1226.
8. Jue Chen, Jinbang Chen, Junchen Ling, Wei Zhang, “Failure recovery using VLAN tag in SDN: High speed with low memory requirement,” IEEE 35th International Conference on Performance Computing and Communications, 2016, pages 1-9.
9. Behrouz Raeisi, Alessio Giorgetti, “Software-based fast failure recovery in load balanced SDN-based datacenter networks”, 6th International Conference on Information Communication and management, 2016, pages 95-99.
10. Capone A, Cascone C, Nguyen AQT, Sanso B, “Detour Planning for fast and reliable failure recovery in SDN with Open State,” 11th International Conference on the Design of Reliable Communication Networks, 2015, pages 25-32.

**Others**

1. Chunhui Pang, Yong Jiang, Qi Li, “FADE: Detecting Forwarding Anomaly in Software-Defined Networks,” IEEE International Conference on Communications, 2016, pages 1-6.
2. Tri-Hai Nguyen, Myungsik Yoo,“Attacks on Host Tracker in SDN Controller: Investigation and Prevention”, International Conference on Information and Communication Technology, 2016, pages 610-612.
3. Sean Peisert, Adrian R. Chavez and William M.S. Stout, “Techniques for the Dynamic Randomization of Network Attributes,” 49th International Carnahan Conference on Security Technology, 2015, pages 1-6.
4. Anja Feldmann, Philipp Heyder, Michael Kreutzer2, Stefan Schmid, Jean-Pierre Seifert, Haya Shulman, Kashyap Thimmaraju, Michael Waidner, Jens Sieberg, “NetCo: Reliable Routing With Unreliable Routers,” 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, 2016, pages 128-135.
5. Stefano Vissicchio, Luca Cittadini, “FLIP the (Flow) Table: Fast LIghtweight Policy-preserving SDN Updates,” IEEE INFOCOM The 35th Annual IEEE International Conference on Computer Communications, 2016, pages 1-9.
6. Seugwon Shin1, Phillip Porras, Vinod Yegneswaran, Martin Fong, Guofei Gu1, Mabry Tyson. “FRESCO: Modular Composable Security Services for Software-Defined Networks”, In Proceedings of Annual Network and Distributed System Security Symposium (NDSS), 2013.
7. Diogo Menezes Ferrazani Mattos, Otto Carlos Muniz Bandeira Duarte, “AuthFlow: Authentication and access control mechanism for Software Defined Networking”, Article in anals of Communication, 2016
8. P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, “A security enforcement kernel for OpenFlow networks,” Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012, pp. 121–126.
9. S. Shin, Y. Song, T. Lee, S. Lee, J. Chung, P. Porras, V. Yegneswaran, J. Noh, and B. B. Kang, “Rosemary: A Robust, Secure, and HighPerformance Network Operating System,” Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014, pp. 78–89.
10. B. Chandrasekaran and T. Benson, “Tolerating SDN application failures with LegoSDN,” Proceedings of the 13th ACM Workshop on Hot Topics in Networks. ACM, 2014.
11. O. O. MM and K. Okamura, “Securing Distributed Control of Software Defined Networks,” International Journal of Computer Science & Network Security, vol. 13, no. 9, 2013.
12. Akshay Baheti, “Extensible Authentication Protocol Vulnerabilities and Improvements”, Master’s Thesis and Graduate Research, San Jose State University.

**BAN Logic**

1. Michael Burrows, Martin Abadi and Roger Needham, “A logic of authentication”, ACM Transactions on Computer Systems, Vol. 8, No. 1, 1990.
2. Kening Liu, Junyao Ye, Yinglian Wang, “The security Analysis on Otway-Rees Protocol Based on BAN logic”, Fourth International Conference on Computational and Information Sciences, 2012., pages 341-344.
3. Jan Wessels, “Applications of BAN logic”, 2001.
4. Li Tingyua, Liu Xiaodong, Qin Zhiguang, Zhang Xuanfang, “An improved security protocol formal analysis with BAN logic”, International Conference on Electronic Commerce and Business Intelligence, 2009, pages 102-105.
5. Ron Rivest, Lecture on Kerberos, BAN logic proof of Kerberos, BAN logic for proving the security of protocols, Computer and Network Security, 1997.
6. Bill Young, Lecture on “Foundations of Computer Security: The BAN logic”, Department of Computer Science, University of Texas at Austin.
7. Ruikang Zhou, Yingxu Lai, Zenghui Liu, Yinong Chen, Xiangzhen Yao and Jiezhong Gong, “A security authentication protocol for trusted domains in an autonomous decentralized system”, International Journal of Distributed Sensor Networks, Article No. 4,2016.
8. Kai Fan, Hui Li, Yue Wang, “Security Analysis of the Kerberos protocol using BAN logic”, Fifth International Conference on Information Assurance and security, Volume 2, 2009, pages 467-470.
9. L Acton, University of Cambridge Computer Library, Security Engineering: A guide to building Dependable Distributed Systems.
10. Christian Haack, Verification of Security Protocols: BAN logic, 2008.
11. <http://www.techrepublic.com/article/ultimate-wireless-security-guide-an-introduction-to-leap-authentication/>
12. https://technet.microsoft.com/en-us/library
13. <https://tools.ietf.org/html/rfc2759> (MS-CHAP v2)
14. <https://tools.ietf.org/html/rfc5216> (EAP-TLS)

**Scyther**

1. Cas Cremers and Sjouke Mauw. “Operational Semantics and Verification of Security Protocols”, Information Security and Cryptography, 2012.
2. Cas Cremers, “Scyther User Manual”, 2014.

PROJECT DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| *Student Details* | | | |
| **Student Name** | **Garuda Suma Pranavi** | | |
| Register Number | 130907120 | Section / Roll No | A/13 |
| Email Address | suma.garuda@yahoo.com | Phone No (M) | +65 83448045 |
|  | | | |
| *Project Details* | | | |
| **Project Title** | **Investigation of Network Topology Poisoning Attacks in SDN** | | |
| Project Duration | 5 months | Date of reporting | 26th Jan 2017 |
| Expected date of completion of project | 24th June 2017 |  |  |
|  |  | | |
| *Organization Details* | | | |
| **Organization Name** | **Nanyang Technological University, Singapore** | | |
| Full postal address with pin code | **Nanyang Technological University,**  **50 Nanyang Avenue, Singapore- 639798** | | |
| Website address | **http://www.ntu.edu.sg/Pages/home.aspx** | | |
|  |  | | |
| *Supervisor Details* | | | |
| **Supervisor Name** | **Prof. Ma Maode** | | |
| Designation | **Associate Professor** | | |
| Full contact address with pin code | **S2. 2-B2-27, School of Electrical and Electronic Engineering**  **Nanyang Technological University, 50 Nanyang Avenue, Singapore** | | |
| Email address | **EMDMa@ntu.edu.sg** | Phone No (M) | **+65 67904385** |
|  |  | | |
| *Internal Guide Details* | | | |
| **Faculty Name** | **Prof. Stanley Oswald Maben** | | |
| Full contact address with pin code | **Dept. of E&C Engg., Manipal Institute of Technology, Manipal – 576 104 (Karnataka State), INDIA** | | |
| Email address | **stan.maben@manipal.edu** | | |