International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 3, May - June 2013 pp-1577-1581 ISSN: 2249-6645

Easier Management Strategy for Small and Large Enterprise Networks with Enhanced Security

Sreelakshmi Ganesh[1], Binu A[2]
Post Graduate Student, Department of Information Technology, Rajagiri School of Engineering & Technology, Kerala, India
Assistant Professor, Department of Information Technology, Rajagiri School of Engineering & Technology, Kerala, India

ABSTRACT: *The goal of this paper is to make Enterprise Network more manageable, secure and thereby reducing complexities without adding a new layer of protocols, applications and scripts to the existing network. This goal is achieved by proposing new network architecture for Enterprise known as Ethane. Ethane is built on the principle that, the only way to manage and secure networks is to identify the origin of all packets, and hold the user (or machine) accountable for it. So first, using Ethane all users, hosts and switches used in the network are well authenticated and tracking is performed in intervals. All packets are identified with their sender or origin. Secondly, Ethane implements a network-wide policy language in terms of users, machines and services. That is, before a packet flows into the network, it is verified against certain access-deny policies, whether to access or deny the flow of data into the network depending upon the behavior and type of flow. Ethane uses simple flow-based switches called Ethane Switches that will permit or deny flows under the control of a centralized controller. Each network contains a centralized controller that decides if a flow is to be allowed into the network. It makes its decisions based on a set of rules that make up a policy. This architecture has been found backwards-compatible with many existing switches and hosts. Ethane can be deployed in both hardware and software supporting both wired and wireless hosts. To avoid broadcast traffic for service and resource discovery, Ethane proposes a new device, the EtherProxy that can be inserted into an existing Ethernet to suppress broadcast traffic.*

Keywords: *ACL, Central controller, Ethane, FSL, NAT*

I. INTRODUCTION

Security and management can be considered as a remarkable part of all our present–day enterprise networks. Each member of the enterprise network is responsible for the security, protection and management of the electronic information resources over which he or she has control. If we consider the Enterprise network, security has become a big problem, as most commonly everyday critical data has to be received and transferred. Enterprise networks are indeed huge networks that can run many applications and protocols mostly working under high security and reliability constraints. It has become a risky task for many network engineers to manage the network as the business productivity is more affected by network misconfigurations and disruptions. The current solutions are weak making the enterprise network more expensive and error prone. Also, most of present day networks require skilled network engineers for the manual configurations of network to attain better security, which is again not a better solution. Different approaches are developed to make an enterprise network highly secure and manageable. The first method is to make use of dedicated network appliance hardware's like middleboxes that can transform, inspect, filter or otherwise manipulates traffic. These middleboxes can exert their control effectively only if placed at network choke-points.

Second method is adding network functionalities which includes providing network diagnosis tools , adding VLANS , for broadcast segmentation, NAT, Access control list (ACLs) to access and deny type of flow , and filters to isolate users, to instrument the routing and spanning tree algorithms to support better connectivity management, and then to collect packet traces to allow auditing. All this will add complexity to the existing network as new layer of protocols, scripts and applications are added. Increasing complexity will make the network less scalable and difficult to manage, less reliable and ultimately less secure. Instead, we believe the answer lies in removing complexity from the switches and routers, reducing them to the very simplest forwarding devices. We believe that the decision of who can communicate with whom should be managed by a network-wide policy running in software in a centralized location.

To make our network more manageable and secure we change the existing network architecture by suggesting ETHANE framework which is based on Ethernet. We need to rethink the enterprise network to make it more manageable without adding new layer of complexity on the top of existing network. Ethane has been built successfully in Stanford University Computer Science Department supporting 300 hosts and it is found that Ethane can be used in campus and university network supporting wired and wireless hosts.

Ethane works based on three principals. First, design a network where connectivity is governed by central policy over high-level names enforced robustly. It is convenient to declare which services a user is allowed to use and to which machine they can connect. Secondly, allowing the network manager to determine the route of packets via policy. These include packets to pass through some intermediate middleboxes like intrusion detection system, firewall and so on. The third principle is that, there must be a strong and secure binding between the packets and their origin. Most of today's bindings have problems.

An Attacker can always interpose between any of the bindings and perform IP or MAC spoofing. Also there arise problems when the bindings change dynamically or when the physical network changes. Since the ARP and DHCP are unauthenticated we go for strong and secure bindings between packets and the source.

To achieve these principles in enterprise network, centralized control architecture was adopted. Centralized solutions are a suitable method for the management of enterprise networks.

The paper is organized as follows. Section I is the introduction to the paper. Section II describes overview of ETHANE design. Section III elaborates the working of ETHANE. The controller components are described in Section IV. Section V, describes a policy language FSL that is used to manage the Ethane implementation. Related Work is elaborated in Section VI followed by proposal for improvement in Section VII. The rest of Section deals with Conclusion.

II. ETHANE DESIGN OUTLINE

The two components involved in the ETHANE design are (i) Controller and (ii) Ethane switches. The controller acts as the main leader in the enterprise networks and has control of the entire network topology. A central policy is declared at this controller and helps to decide which user can communicate with whom after explicit permission. The policy can be access policies, deny policies and waypoints. When a new packet flow arrives, Controller verifies the flow against the network wide security policy. If the flow is allowed, the Controller chooses a route for the flow and installs the accepted flow-entry to entire switches along the path. But, If the flow is denied, no flow-entry will be added and the packets are not forwarded further.The controller keep track of all the bindings (mainly handling IP address, DNS lookups, authentication of user, switches and end-hosts) providing more security and authentication in the network.

The second component Ethane Switch plays a very important role in managing the Ethane design. Ethane design can know the origin of packet,by keeping track of all the users, hosts and authenticating namespace and addresses. The Ethane switch can replace the normal Ethernet switches and are basically simple switches. These switches include a managed flow table and establish a secure connection with the controller. The flow table consists of flow entries that contain a header, an action and per-flow authorised data. Header is needed to decide the type of flow (TCP, UDP, and IP), Ethernet headers and physical port information of the switch. Action helps the switch to decide what to do with the packet. Action can be forwarding packet, updating packet-byte information, setting the number of inactive entries. Controller will remove the inactive entries due to timeout. Per-flow authorised flow has per-flow entries mainly for application data flow and per-host entries for the hosts that are misbehaving.

Instead of adding functionalities like ACL, NAT, MAC or IP address look up , Spanning tree etc. to entire switches along the path , a method to reduce the cost, power consumption and complexity can be achieved in the network by adding more functionalities in a single location like a controller by enforcing some global security policies.

The Ethane switch requires a local switch manager to establish and maintain a secure connection to the controller for monitoring link status, providing an interface for any additional switch-specific administration and diagnostics.

2.1 Name space, Bindings , Policy language

A Controller evaluates the incoming packets against set of rules like guest visitor accessing the http using proxy server. Today's namespace stores lot of names (hosts, users, services, protocols).These names are bound to network realities (like DNS names mapping to IP, MAC to IP and so on). If the mappings between these names, IP, MAC are unauthenticated, attackers can easily attack the network and it becomes a well-known weakness in the current network.

A controller running in the Software or hardware PC can make the namespace consistent as components join, leave and move around the network. Any network state changes require updating the bindings at the controller. Ethane provides strong and secure bindings between the packet and the origin. Ethane takes over all the bindings of addresses. When host uses DHCP to request an IP, Ethane assigns it knowing to which switch port the machine is connected, enabling ethane to attribute an arriving packet to a physical port.Secondly, the packet has to come from a machine that is registetred on the network.The controller can record all the packet flow entries and bindings in a log, that can be used later to regenerate the network events.

For a controller to enforce network fine grained policy, it is essential to write a policy language. The policies are specified using a simple declarative policy language. These policy language runs on high level names supporting large enterprise networks. Ethane uses a policy language known as *FSL* (Flow-based Security language). FSL includes set of rules pertaining to a flow consisting of a condition and corresponding action. These rules or policies can be allow, deny and waypoint policies.

III. WORKING OF ETHANE

The working of a simple Ethane Network consists of the following steps which is demonstrated using figure 1.

a) **Registration:** The switches, hosts and users in the network must be registered at the controller along with the credentials or authorizations for the purpose of authentication. Every host can be authenticated using MAC address, users with username and password, and switches using secure certificates.

b) **Bootstrapping:** The connectivity is maintained by bootstrapping switches by creating a spanning tree with the root as the controller. Spanning tree creation helps to avoid network loops and broadcast storms. Each switch in the spanning tree authenticates with the controller and establishes a secure channel to it. After maintaining a secure connection, the switches send link state information to the controller, using which it can reconstruct the network topology.

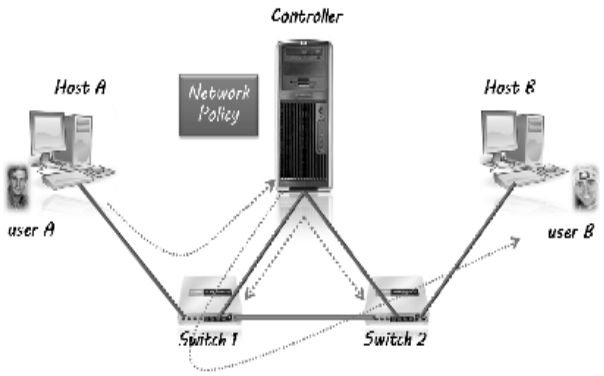

Figure 1: Communication in Ethane Network

c) **Authentication:** From the above figure 1, the user A will come into connection with Host A. Since there are no flow entries for the new user or new host existing in Switch 1, it will initially forward all the Host A packets to the controller. Host A request a DHCP to the controller. After checking host A's MAC address , the controller will allocate an IP address (Ip A) for it, binding host A to IpA., IpA to Mac A , and Mac A to a physical port on switch 1.UserA opens a web browser where the traffic is first directed to Controller, authenticating using a Web form . Once authenticated , User A is bound to Host A.

d) **Flow Setup:** A connection is initiated by User A to User B. The packet is forwarded to the Controller by Switch 1 after determining that the packet does not match any active entries in its flow table. On receiving the packet, the Controller decides whether to allow or deny the flow, or require it to traverse a set of waypoints based on the type of flow and using policy language. If the flow is allowed, the Controller computes the route of flow. The Controller then adds a new entry to the flow tables of all the Switches along the path.

e) **Forwarding:** If the Controller allowed the path, it sends the packet back to switch 1, which forwards it based on the new flow entry. Subsequent packets from the flow are forwarded directly by the Switch and are not sent to the Controller. The flow-entry is kept in the switch until it times out

IV. CONTROLLER COMPONENTS

Controller who is the root of Enterprise Networks consists of several components. These includes:

a) **Authentication:** It authenticates users and hosts using credentials stored in the registration database.The credentials for the user include username, password and are authenticated via Web interface.Switches can authenticate using their MAC address, registered in the database.

b) **Policy File:** When a new flow arrives, it is verified against certain policies or rules so as to accept deny or route through a waypoint.

c) **Route Computation:** It uses the network topology to pick the flow's route. Route can be calculated using shortest path algorithm.

d) **Switch Manager:** The topology contains a switch manager, which receives link updates from the Switches so that it becomes easier to reconstruct the network topology for route computation.

e) **Registration Database:** All entities like hosts, protocols , switches ,users etc. are registered via a Web interface to the controller in the database. Example: Authentication of switches are done by seeing if they are registered.

f) **Bindings:** can easily track all the bindings between names, addresses, and physical ports on the network, even as Switches, hosts, and users join, leave, and move around the network. Controller can even record or log the bindings so that each log can be queried to know the bind state at any timestamp.

g) **Permission Check and Access Granting:** Upon receiving a packet, the Controller checks the policy to see what actions apply to it.

h) **Policy Compiler:** This is required to compile the policy file in the controller.

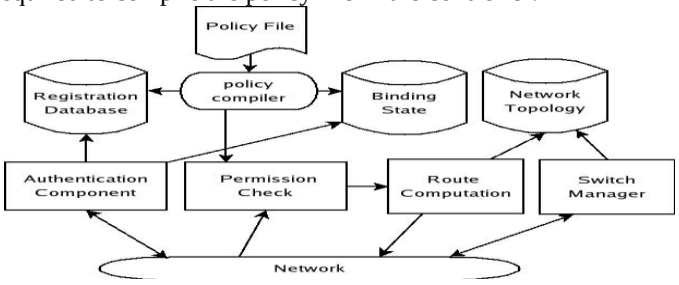

Figure 2: Controller Components

Fig 2 illustrates the high-level view of controller components. In many cases when the centralized controller fails, there are techniques and approaches to replicate the controller and improve the fault tolerance and scalability.

V. POLICY LANGUAGE

Policy Language is a declarative language to define policies or set of rules in the controller and Ethane Network.These policy languages are used for expressing and enforcing Flow-based Network security policies and for the easy administratition of the controller.To control the Ethane network, a DATALOG-based language with negation called FSL has been developed. FSL policies are set of rules used to bind unidirectional flows to constraints that should be placed on the flows. The policy language includes a condition and an Action. For example, the rule to specify that user *bob* is allowed to communicate with the Web server using HTTP is:

$$[\mathbf{allow}() <= \mathbf{usrc}(\text{“bob"}) \wedge \mathbf{tpdst}(\text{“http"}) \wedge \mathbf{hdst}(\text{“websrv"})].$$

A condition is a conjunction of zero or more literals describing the set of flows an action should applied to. Example : If the user initiating the flow is “bob” **and** the flow destination transport protocol is “HTTP” **and** the flow destination is host “websrv,” then the flow is *allowed*.

The Actions include allow, deny and waypoint. The language or policy can be developed in C++ or python. Sample policy file developed using python is shown below:

```
allow() <= tpdst(8888) ∧ hdst(“emerson”)
fn_action(“http_redirect”) <= in(‘laptops’,HSRC) ∧ usrc(“unauthenticated”)
# allow ARP and DHCP
allow() <= protocol(‘arp’)
allow() <= protocol(‘dhcps’) ∧ hdst(“gateway”)
allow() <= protocol(‘dhcpc’) ∧ hsrc(“gateway”)
# allow computers to ssh into anyone
allow() <= protocol(‘ssh’) ∧ in(‘computers’, HSRC)
# dissallow testing machines from communicating externally
deny() <= in(‘testing’, HSRC) | in(‘testing’, HDST)
# servers should be inbound-only
deny() <= isConnRequest() ∧ (in(‘servers’, HSRC) | in(‘printers’, HSRC))
# printers should be inbound-only
deny() <= isConnRequest() ∧ (in(‘printers’, HSRC) | in(‘printers’, HSRC))
# laptops and mobile devices should be outbound-only
deny() <= isConnRequest() ∧ (in(‘mobile’, HDST) | in(‘laptops’, HDST))
# allow workstations unfettered access
allow() <= in(‘workstations’, HSRC) | in(‘workstations’, HDST)
# allow known devices to communicate as long as they abide by the
# previous rules.
allow() <= in(‘all’, HSRC)
# default deny
deny() <= True
```

Figure 3: Sample Policy file format

VI. RELATED WORK

The network goals of any enterprise can be achieved by enforcing global network policies in a centralized controller. Better security and management can be achieved by use of Ethane design thereby reducing the overall complexities .These policies are maintained on the flow of packets. By simply updating the controller at central location adding of functionalities and features, for achieving security and management becomes easy. Ipsilon Networks work for providing a switched, multiservice fast and simple path to traditional IP routers by caching routing decisions.Using FSL ,apart from allow –deny policies , waypoints can also be configured. FSL was evolved from Pol-Eth, the old Ethane policy language, and supports dynamic group memberships, negation, conflict resolution, and distributed authorship.FSL policy language can be applied to all types of packet flow thereby setting priorities for different type of users. For a large enterprise networks, setting VLAN (logical broadcast domain) becomes a hectic task as it requires much hand-holding and manual configuration each time. Rather than configuring VLAN on all switches, Ethane controller can provide simpler control over isolation, connectivity and diagnostics. Also many identity-based networking custom switches provides less control over the network data-path. This control can be achieved using ETHANE.

VII. PROPOSAL FOR IMPROVEMENT

To avoid broadcast traffic for service and resource discovery, paper proposes a new device, the EtherProxy that can be inserted into an existing Ethernet to suppress broadcast traffic. For protocols that use broadcast, an EtherProxy caches protocol information carried by protocol messages passing through it. Then for each of those protocols, the EtherProxy employs an algorithm that may suppress subsequent broadcast protocol messages with the aid of the cached protocol information. For example, an EtherProxy can cache the ARP response to an ARP request. Subsequent ARP requests for the same IP address can be served directly from the EtherProxy’s cache rather than broadcasting the request over the network. EtherProxy is backward compatible and requires no changes to existing hardware, software, or protocols. Moreover, it requires no configuration. This allows a single Ethernet network, with a single broadcast domain, to scale to much larger sizes than before. The EtherProxy intercepts broadcast packets it recognizes then it either (1) responds to a host’s query, sent via a broadcast packet, itself using its cached information and without forwarding the request packet to the network, (2)

replaces the broadcast destination address in the packet with the appropriate unicast destination address, cached at the EtherProxy, and then sends the packet over the network, or (3) passes on the broadcast packet as is, if it contains information that needs to be disseminated to all hosts in the network. If the broadcast packet's protocol is not recognized and there is no rule for handling it, then the packet is just passed through. Every protocol using broadcast packets can be handled by one of the approaches mentioned above.

VIII. CONCLUSION

Using ETHANE, it becomes much easier to manage the Enterprise network than expected. Ethane Network is highly Scalable and enables adding new hosts, switches, new users, new protocols. It provides fast and easy method to set new policy rules in a single Central location. In case of failure of central location, the Controller can be replicated. It allows extending the policy language, adding new routing algorithms. Controller can scale to support quite large networks. Ethane Switch will be significantly simpler, smaller, and lower power than current Ethernet switches and routers that are very costly.

To avoid broadcast traffic for service and resource discovery, paper proposes a new device, the EtherProxy that can be inserted into an existing Ethernet to suppress broadcast traffic. For protocols that use broadcast, an EtherProxy caches protocol information carried by protocol messages passing through it. Then for each of those protocols, the EtherProxy employs an algorithm that may suppress subsequent broadcast protocol messages with the aid of the cached protocol information. For example, an EtherProxy can cache the ARP response to an ARP request. Subsequent ARP requests for the same IP address can be served directly from the EtherProxy's cache rather than broadcasting the request over the network. EtherProxy is backward compatible and requires no changes to existing hardware, software, or protocols. Moreover, it requires no configuration. This allows a single Ethernet network, with a single broadcast domain, to scale to much larger sizes than before. The EtherProxy intercepts broadcast packets it recognizes then it either (1) responds to a host's query, sent via a broadcast packet, itself using its cached information and without forwarding the request packet to the network, (2) replaces the broadcast destination address in the packet with the appropriate unicast destination address, cached at the EtherProxy, and then sends the packet over the network, or (3) passes on the broadcast packet as is, if it contains information that needs to be disseminated to all hosts in the network. If the broadcast packet's protocol is not recognized and there is no rule for handling it, then the packet is just passed through. Every protocol using broadcast packets can be handled by one of the approaches mentioned above.

REFERENCES

[1] Casado,M. Stanford Univ., Stanford, CA, USA Freedman, M.J, Pettit,JianyingLuo,Gude,N.,McKeown,N., Shenker, S "*Rethinking Enterprise Network Control*" ,Networking, IEEE/ACM Transactions on Aug. 2009 Volume: 17 , Page(s): 1270 – 1283" .
[2] Roscoe, S. Hand, R. Isaacs, R. Mortier, and P. Jardetzky,"Predicate routing: Enabling controlled networking," *Comput. Commun. Rev.*, vol.33, no. 1, 2009.
[3] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S.Shenker, "*Ethane: Taking control of the enterprise*," in *Proc. SIGCOMM*, Kyoto, Japan, Aug. 2008, pp. 1–12.
[4] A. Myers, E. Ng, and H. Zhang, "*Rethinking the service model: Scaling Ethernet to a million nodes*," presented at the HotNets, Nov. 2008
[5] *"Cisco Network Admission Control,"* [Online]. Available: http://www.cisco.com/
[6] *"Microsoft Network Access Protection,"* http://www.microsoft.com/technet/network/nap/default.mspx
[7] *"Alterpoint,"* [Online]. Available: http://www.alterpoint.com/
[8] G. Xie, J. Zhan, D. A. Maltz, H. Zhang, A. Greenberg, and G. Hjalmtysson, "*Routing design in operational networks: A look from the inside*," in *Proc. SIGCOMM*, Sep. 2004, pp. 27–40.
[9] D. Caldwell, A. Gilbert, J. Gottlieb, A. Greenberg, G. Hjalmtysson, and J. Rexford, "*The cutting edge of IP router configuration*," *Comput. Commun. Rev.*, vol. 34, no. 1, pp. 21–26, 2004.
[10] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "*A clean slate 4D approach to network control and management*," *Comput. Commun. Rev.*, vol. 35, no. 5, pp. 41–54, Oct. 2005.
[11] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown, and S. Shenker, "*SANE: A protection architecture for enterprise networks*," in *Proc. USENIX Security Symp.*, Aug. 2006, vol. 15, Article No. 10.
[12] T. Hinrichs, N. Gude, M. Casado, J. Mitchell, and S. Shenker, "*Practical declarative network management*," presented at the ACM Workshop: Res. Enterprise Netw., 2009.
[13] C. Demetrescu and G. Italiano, "*A new approach to dynamic all pairs shortest paths,"* in *Proc. STOC*, 2003, pp. 159–166.