Customized Data Exchange Gateway (DEG) for Automated File Exchange across Networks

*****Abhishek Vora B. Lakshmi C.V. Srinivas

National Remote Sensing Center (NRSC), Indian Space Research Organization (ISRO), Balanagar
Hyderabad, India
**Mr.Abhshek Vora was formerly with NRSC*

ABSTRACT

This paper addresses a customized solution to ensure security of the trusted network while receiving files from applications residing on less trusted networks. The solution is a four layered secured file transfer service which controls and authenticates the data transfer through service blocking mechanism and digital signatures at the first two levels and by introducing a novel concept of privileged socket creation and finger printing TCP packets at layer three and four by customizing the Linux kernel. The solution is deployed using a pair of systems connected peer to peer running customized Linux kernels and the solution will ensure that the first system on the gateway accepts only authentic data and transfers to second system which accepts the data only when it originates from the first system. The link between the two systems and the systems as such are physically protected. Data is received only from the first system. In this paper we explain its security architecture and discuss implementation on Linux kernel 2.6.24.2

Index Terms
Network Security, Secured automated file exchange, Kernel hardening.

1. INTRODUCTION

Connectivity and Seamless data exchange is a prerequisite to automate the business work flow across distributed locations. Generally the connectivity is through computer networks using TCP/IP based protocols. This connectivity makes the networks vulnerable to attacks from each other. The degree of vulnerability is higher when any of the networks is connected to Internet. Hence ensuring secured mechanism for transferring the files across the participating network is a requirement.

In addition to configuring the commercially available Firewall and Intrusion detection and prevention systems deploying a customized solution suiting the specific enterprise requirements will ensure total security. The task of the customized solution is to provide total security for automatic movement of data across the networks of varying level of trust knowing the involved networks, systems, applications, types of files, direction of traffic flow, file format and several other attributes. The solution addresses a secured mechanism for the preregistered applications to forward the data to the storage accessible to processing network either through fibre channel interface or other TCT/IP network.

Section 2 discusses the need for customized solution even when generic security systems like Firewall and IDPS exist. Section 3 explains five layered security architecture. It also discusses how this proposed solution counters different types of attacks. In section 4 we explain implementation on Linux kernel 2.6.24 running inside Operating System RHEL 5.0.

2. INADEQUACY OF 'GENERIC' SECURITY SYSTEM

Generally networks are secured using generic solutions like Firewall and Intrusion Detection & Prevention System (IDPS). But when connecting the systems of varying trust customized security systems are required as the generic solutions can not address specific requirements of the enterprise.

Firewall functions on the basis of pattern matching. It protects networks against attacks by filtering the traffic based on policy rules set in advance. The policy rules use various packet attributes like port number, IP address, protocol etc. to filter the traffic. Due to its full reliance on packet header attributes to control the traffic, firewall fails in protecting network from two broad categories of attacks viz. Masquerading and Vulnerability Exploitation [6].

In first type of attack if attacker injects traffic with fraud header attributes directly at Data Link Layer (through RAW or PACKET sockets) or hijacks port from legitimate application, firewall will fail to detect the fraud and attacker can get into the network. In second type of attack if the attacker exploits vulnerability in a service accessible over the network, firewall will fail in detecting attack as its filtering capabilities does not extend to packet payload. Thus, when immaculate security is the need, firewall can not be relied upon fully [7].

Networks can be secured against vulnerability exploitation attacks by deploying Intrusion Detection & Prevention System (IDPS) along with Firewall. Signature based IDS are capable of inspecting packet payloads and taking filtering decision based on the signatures found in packet payload. Anomaly based IDS monitors system & network activities and alerts an attack, whenever threshold values are exceeded. IDPS too fails in detecting an attack in two cases. In the first case, if the attack is conducted using advance mechanisms for which no signature is yet published or the signature Database is not updated. In the second case, if the attack does not cause system or network activity so high as to reach the threshold value.

3. DATA EXCHANGE GATEWAY (DEG)

The proposed solution Data Exchange Gateway is a security mechanism that needs to be deployed on the two gateways of the involved networks. DEG is designed with multilayer security architecture. The security mechanisms like service blocking, authentication, integrity checks & content filtering, customized traffic filtering, packet finger printing are deployed to protect the networks against attacks ***at different*** layers.

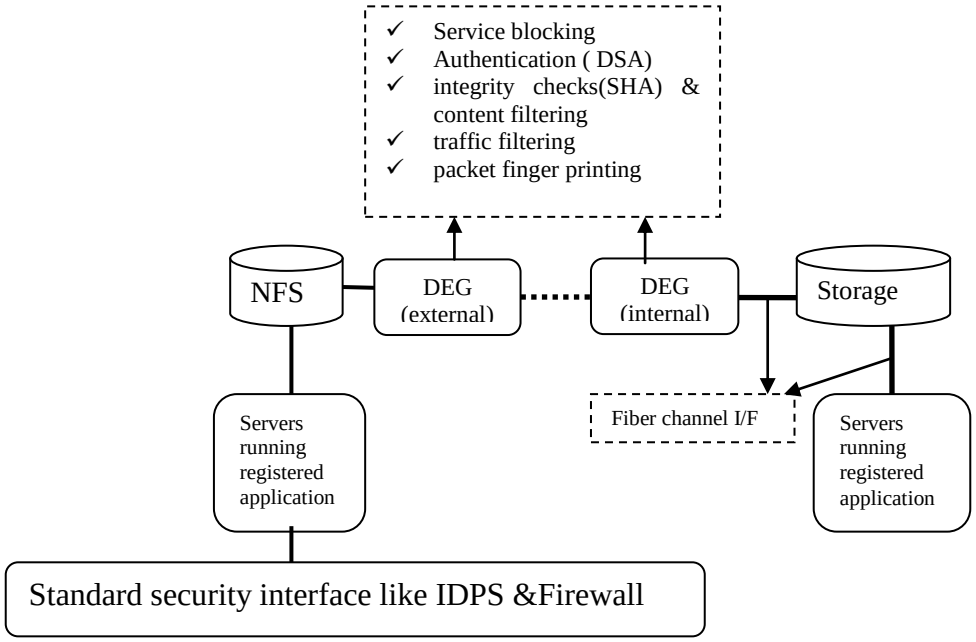

Figure : 1 Deployment of Data Exchange Gateway

The proposed Data exchange gateway (DEG) is configured using one system on each of the network connecting to the respective network through disk storage accessible to the authentic nodes and is used to post and deliver the data from and to the applications network nodes through a customized file transfer service (FTS). The two systems are connected P2P and they together act as controlled gateway for data exchange. The FTS is expected to ensure the security by dispatching the files which are posted by authentic applications alone. The kernel services related TCP/IP server/client on each of the system are customized to enable FTS to control the traffic, hence physically security of the two systems is a prerequisite for the effectiveness of solution. If the kernel is reloaded the solution fails.

3.1 Layered architecture

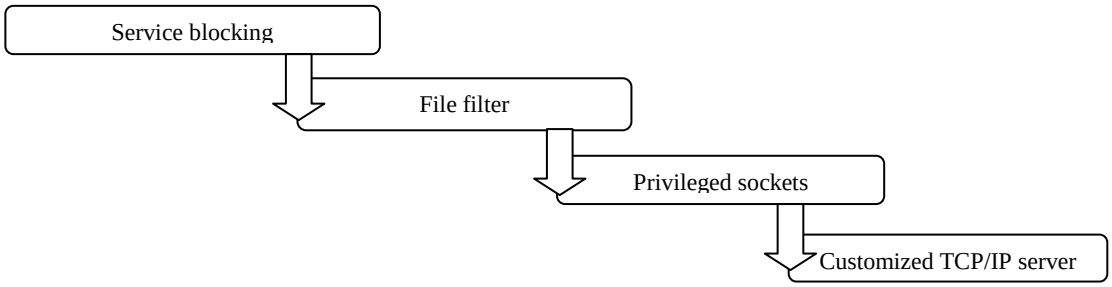

Figure 2 Layered Security architecture of DEG

Layer-1 Service Blocking
First layer restricts network service access to DEG system by blocking all network services other than NFS. Disabling remote command execution and file transfer services significantly reduces the chance of DEG being compromised from systems through network.

Layer 2 File Transfer Service
The customized file transfer service identifies and drops the files which are not from authentic source (preregistered applications with digital signature) and the file attributes are not as per the predefined properties.

Layer-3 privileged sockets for communication
The customized TCP socket creation and registration enables the file transfer service to register and communicate on privileged sockets only.

Layer-4 customized TCP/IP client-server
The customized TCP server/client on each DEG ensures that FTS transfers/receives only the finger printed data packets, which establishes the authenticity of the source of the packets.

4. IMPLEMENTATION

Secured file exchange by DEG is achieved using customized FTS and modified TCP Server & TCP Client. FTS is the

service for transferring the files from storage of one network to the storage on the other network. A registration process is implemented to register the application which needs the service. The service registration is an executable which needs to be executed on the DEG system by giving all the required details of the application and the system from where the application executes. After successful registration is a key pair is generated and passed on to the application. On sender side the FTS verifies the authenticity of files on storage and transfers them to receiver which verifies the origin of the packets and hands over the authentic packets to its FTS service to deliver onto the storage. In addition to authentication FTS implements file filtering, gets access to the customized sockets authorized to access the TCP client which fingerprints TCP packets and delivers to receiver network. The FTP service on receiver side checks the authenticity of the packets and dispatches them to the disk storage on receiver side.

4.1 Implementation of layered security

4.1.1 *Layer 1[4] [5]*

Both DEG systems run software based firewall (using IPTables) which filters incoming traffic and allows only the traffic destined to NFS port. This will reduce the attacks using the network services to a large extent.

4.1.2 Layer 2

Each client application which needs the files to be delivered to the other network through DEG FTS service has to register through a registration process and obtain a public-private key pair for digitally signing the files. The registered client can post the digitally signed files on to NFS mounted storage. The FTS at this layer filters the files first by the verifying the application digital signature and then by checking the against the predefined file attributes. The files which do not pass through the file filter are dropped and warning messages are logged for monitoring. The files passing through the filter proceed to next process in FTS.

4.1.3 Layer-3 Access Restriction and traffic filtering [8] [9] [10] [11]

The access restriction is based on new system service namely privileged socket creation and registration. This is done by customizing the TCP/IP protocol stack and socket structure modification. Traffic originating from privileged socket is only allowed. The packets received from such sockets are fingerprinted which is done at kernel level.

Security at this layer restricts applications other than FTS from sending traffic to remote networks by filtering outgoing traffic. Instead of using IP_FORWARD to disable routing, it is stopped within kernel by modifying ip_rcv() function.

Traffic filtering based on port numbers, process ID or process name will fail in stopping an attack if the attacker masquerades identity. User-id based filtering also fails to stop attacker's packets as it allows all the applications running under that user to send traffic.

Here traffic filtering is implemented by introducing a concept of 'privileged' socket. To send any traffic to network, it is required to register its socket. After successful registration the sockets gets 'privileged' status. Later, all the traffic from this socket is allowed to go to remote network. To implement this, TCP/IP protocol stack of kernel is customized by redefining the socket structure. This registration is enabled by introducing new system call. Structure 'socket' has been augmented with a field 'privilege'. Before registering a socket, the system call authenticates the process using symmetric key. The customized TCP Client and TCP Server are the only applications having knowledge of symmetric key. Hence no other application can create privileged socket and hence will be able to send data across the network.

In TCP/IP protocol, default implementation of data link layer function dev_queue_xmit() has been augmented with filtering logic. This logic allows traffic from only privileged socket to go to the destination network. When this function gets packet (structure sk_buff) to transmit to remote network, it finds out which kernel socket (structure sock) has caused it following which it identifies corresponding user space socket (structure socket). If the user space socket is not privileged to send traffic across the networks the packet is dropped, otherwise it is fingerprinted and sent to the DEG on destination network.

The system call '*socket ()*' has been modified to stop creation of non AF_INET (TCP) socket. Hence no process can create RAW or PACKET sockets and so direct packet injection at data link layer or packet sniffing will not be possible on this customized kernel.

.1.4 Layer 4

Layer-4 security protects DEG systems in the event when an attacker gets physical access to the link between DEGs. This is achieved through Selective reception. Instead of accepting all incoming packets the receiver DEG filters the incoming packets based on fingerprint attached. Selective Reception is implemented by modifying functions dev_que_xmit() (dev.c) and ip_rcv() (ip.c). The former is augmented with fingerprinting logic and later with validation and filtering logic.

On receiver DEG whenever the function ip_rcv() receives an incoming packet, before passing it to higher layer function (ip_rcv_final), it validates the packet fingerprint. If fingerprint is not present or is found to be invalid, packet is dropped. If fingerprint is found to be valid, this function removes it from packet and then passes the packet to higher layers of protocol stack.

Lack of knowledge about packet fingerprint and about symmetric key for fingerprinting stops an attacker from creating the packets which can accepted by receiver DEG. So even if attacker succeeds at tapping and injecting packets, those packets will be dropped by receiver DEG as long as kernel of the system is reloaded.

4.2 Software architecture

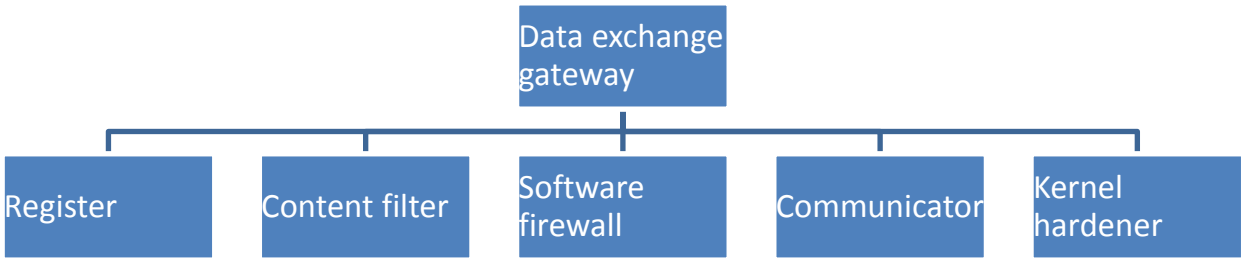

Figure 3 software components

The five components of the software deployed on the Data exchange gateway system are described in this section.

4.2.1 Register

This component of Data Exchange Gateway implements registration facility. As discussed before, to use the FTS service to transfer the data the application has to get registered with the DEG. On successful registration, a pair of public-private keys are generated which are to be used for signing the file to make the file authentic.

4.2.1 Software Firewall

This component implements service blocking. As discussed earlier, all the services other than NFS will be blocked on DEG. This component will be software based firewall, implemented through iptable/ipchains. This forms the first layer of security.

4.2.2 Content Filter

Content filter filters the files before sending it to 'destination DEG'. File filtering at layer-2 is done at two levels. First level of filtering is done through file authentication. On receiving a file FTS tries to establish identity of sender application using the digital signature. If FTS fails in authenticating the sender it drops the file. After successful authentication, it passes through second level of filtering which is based on file attributes.

If either of checks fails, file is dropped. Besides filtering Layer-2 security also alerts administrator if it finds an unauthenticated file. It is possible to find out the node from the unauthenticated file was sent as different nodes in local network are assigned different mount points on the gateway.

4.2.4 Communicator

Communicator implements required interface to fetch the files posted by authorized applications and despatch them on the destination for authorized applications to collect.

4.2.5 Kernel Hardener

The basic functionality of this component is to filter the traffic to ensure TCP packets generated by DEG are only accepted by another DEG. This is achieved by custom build TCP/IP based Client-Server programs on each of the system. This will stop any attempts through IP Spoofing or Packet injection.

The implementation of traffic filtering is done by defining privileged TCP sockets and fingerprinting of TCP packets. Provision for this socket creation is incorporated in the kernel by defining new system call. The privileged socket is accepted only by the customized TCP client-server implementation.

At the Data Link Layer of the sender gateway, all the packets from privileged socket is fingerprinted with the MAC code calculated over packet payload using secret key. Receiver DEG will calculate fingerprint for the received packet using same secret key, valid fingerprint ensures that it has been produced by DEG.

5. FUTURE SCOPE

The assumption made while designing DEG was that gateway systems are physically protected. Present version of DEG will fail if attacker gets physical access to gateway & rebuilds the kernel. Feasibility of a firmware based implementation is being analyzed to overcome this limitation.

6. CONCLUSION

The four layer security solution built with Linux kernel customization cannot be compromised through any type of known or unknown network vulnerabilities to send unauthorized or malicious data to internal DEG as the gateway system does not accept the data packets unless they originate from the external DEG system application.

7. ACKNOWLEDGEMENT

The Authors acknowledge the efforts of Ms.Naseeb Shaik who is responsible in carrying out extensive testing and deploying it to operational use. We express our sincere gratitude to Director NRSC for providing us the opportunity.

8. REFERENCES

[1] Linux Kernel Version 0.8-3, David A Rusling

[2]Introduction to TCP/IP Network Attacks, Guang Yang

[3]TCP/IP Architecture, Design and Implementation in Linux (Practitioners) Sameer Seth and M. Ajaykumar Venkatesulu

[4] Martin A. Brown, "Guide to IP Layer Network Administration with Linux V 0.4.5", Ch-7 March 2007 Available: http://linux-ip.net/html/ch-packetfilter.html

[5] Gianluca Insolvibile, "Inside the Linux Packet Filter", Feb 2002 Available www.linuxjournal.com/article/4852

[6] William Stallings, "Cryptography and Network Security: Principles & Practice" 5th edition, Pearson ch 6, 11, 13, 19. pp. 192-218, 327-362, 395-407, 615-647

[7] Shari Lawrence Pfleeger and Charles P. Pfleeger, "Security in Computing" 4rd edition, Prentice Hall PTR, Oct 2006 Ch- 2, 7

[8] M. Tim Jones, "Anatomy of Linux Network Stack", June 2007,Available http://www.ibm.com/developerworks/linux/library/llinux-networking-stack/

[9] Arnout Vandecappelle and Mind, "Kernel Flow", Nov 2009, Available: http://www.linuxfoundation.org/collaborate/workgroups/networking/kernel_flow

[10] Chrisitan Benvenati, "Understanding Linux Network Internals", O'Reilly Dec 2005. Ch-18, 19, 20, 21

[11] M. Tim Jones, "Kernel Command Using Linux System Call", Feb 2010, Available:http://www.ibm.com/developerworks/linux/library/lsystem-calls/