

Distributed Filtering with Contags and Security-Labels

Matthias Kabatnik¹, Reiner Sailer²

¹ *Institut für Nachrichtenvermittlung und Datenverarbeitung, University of Stuttgart, Germany*

² *IBM T. J. Watson Research Center, Yorktown Heights, NY, USA*

kabatnik@ind.uni-stuttgart.de, sailer@watson.ibm.com

ABSTRACT

This contribution presents a new access control method based on distributed filtering of data packets at network boundaries. It addresses well-known security problems that occur at network interconnection points. Our method achieves finer-grained access control than existing filtering methods by accumulating context information and distributing filter stages. We enhance conventional filter criteria—such as network address, port number, or transport protocol—by including security labels and context information. Security labels store secrecy levels, integrity levels, and categories. Context tags (contags) accumulate context information, e.g., over which incoming link a data packet was received or whether a data packet was received over protected links. This information can be examined by subsequent, possibly remote, filter stages. The authenticity of these filter criteria is crucial for the overall security. We use the history information accumulated in a data packet itself to establish trust in the included filter criteria.

We focus on the structure of distributed 3-stage filters comprising receiving inspection, tagging and re-labeling, and filtering. An example shows how to use context information to improve access control for signalling networks in a heterogeneous service provider environment. This is a pre-requisite for opening service interfaces in global telecommunication networks.

I. INTRODUCTION

This contribution presents a new access control method based on distributed filtering of data packets at interconnection points between networks. Here, the emphasis is on security interworking rather than on protocol interworking.

First, we illustrate the problem of interworking in heterogeneous network environments. Then, we present a concept to structure heterogeneous network environments—i.e., to isolate homogeneous domains—to make them accessible to efficient network-level access control. The main part describes the mechanism in more detail and illustrates its value with an example.

A. Interconnection of heterogeneous networks

The interconnection of computer and telephony networks with each other and with the Internet enables services that set a new standard regarding flexibility, availability, and cost-effectiveness. This trend will continue in the future.

At the same time, this interconnection poses new challenges regarding the security of information that is

accessed through and transmitted over these interconnected networks.

One example is the Virtual Private Network (VPN). Simply speaking, a VPN is an effort to make a virtual network look like a private one—including quality of service and security—despite the fact that shared infrastructure is used. VPNs primarily offer remote access connectivity, site-to-site connectivity, and local services like individual numbering schemes [13]. From a security point of view, the emphasis is on data protection and access control. Access control is explicitly executed at the end-points of VPN tunnels by authenticating remote access users or remote VPN routers. When crossing the VPN, the data are protected by end-to-end encryption and integrity means (e.g., using IKE / IPSEC [2] possibly over layer 2 tunnels).

Even more challenging is the interconnection of networks of different enterprises or network providers because the interconnected parties have different interests and security policies. Here, the emphasis is on access control because the interconnected companies (e.g., network service providers) are competing in the market. From a security perspective, the pure interconnection is much more challenging because access to data cannot be protected by end-to-end data protection only (IP security protocols etc.). Some reasons are:

- Networks offer more and more value-added services (e.g., Intelligent Network services in public networks, ftp/telnet/www servers in IP-based networks) that process application-level service control information that originates from foreign networks. Thus, solving the interconnection security problem by closing the own network domain for any foreign application-level service control data is not an option because it disables most value-added global services.
- At the same time, processing foreign application-level service control data within a network is very critical. Processing forged service control data can lead to severe damage within the network (e.g., disturb service provisioning, lead to false billing data).
- End points of control message flows are not always clear because network addresses might not be completely resolved within the network. Remote end point information might not be trusted, e.g., if the end point is outside the local network administration domain.

Security considerations regarding the interconnection of signalling networks¹ in public telephony networks have been reported by several authors, e.g., Ward [10].

¹ The Signalling System Number 7 (SS7) allows the exchange of control and management data in public telephony networks, e.g., in the Integrated Services Digital Networks (ISDN, Broadband-ISDN).

Therefore, we developed an access control scheme that enables open service provisioning over interconnection points while preserving fine-grained and efficient access control for the network providers.

Before we present our distributed access control scheme to protect interworking in heterogeneous network environments, we need to have a closer look on how to structure these networks in order to efficiently apply access control.

B. Domain Concept

Our access control technique is based on refining large heterogeneous networks into a network of isolated and (from a security perspective) homogeneous subnetworks. Those homogeneous subnetworks are called domains.

A domain is characterized by a homogeneous security policy and consists of hosts with similar protection levels. The security policy describes the authorization of subjects (hosts and network nodes, or domains) to access objects (receive or send data packets).

Each domain is protected autonomously, and interworking between domains (inter-domain communication) is subject to sophisticated access control. Examples for domains are subnetworks or local area network (LAN) segments that are protected by firewalls or physical separation, and signalling networks of public switched telephony networks (PSTN) that are protected by message and parameter filters.

We assume that the security requirements within domains are satisfied by internal security means (local security policy and mechanisms). An ISDN network provider, for example, can autonomously enforce security requirements within the network. At interconnection points however, access control mechanisms must ensure that information flows between domains are compliant with the security policies of the source and the destination domain of the flow. For instance, control data representing management and tariff information that originate from foreign domains (or have been transported over insecure domains without protection) should not be processed inside the local domain.

Recapitulating, we apply the divide and conquer strategy to protect large networks by refining them into smaller pieces. We then protect these pieces independently and, finally, enable these pieces to communicate with each other in a controlled way using efficient access control mechanisms. We focus on the protection of inter-domain communication (interworking) that preserves the security and integrity levels for sensitive data as they are given for each domain. We present a network-level mechanism that inspects data that enter or leave a domain in order to guarantee this protection level for services spanning multiple network domains and to enable securely controlled interconnection of network domains.

C. Network-level Mandatory Access Control

In the following, we refine access control schemes that are usually applied to protect a single computer system to data packets that cross interconnection points. Here, the objects are data packets (IP packets, SS7 signalling mes-

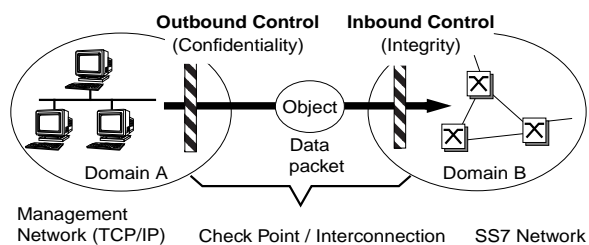


Figure 1: Interconnection of independently protected Domains

sages) and the subjects are sending and receiving domains (subnetworks). Fig. 1 depicts this scenario of protected interconnection.

We assume that a data packet is sent from one domain A to another domain B and that this data packet is subject to certain secrecy or integrity constraints. In this case, the interconnection point between domains A and B must ensure that the secrecy and integrity requirements are preserved though the data packet changes the surrounding domain.

Respective protection can be implemented in different ways: the access control mechanisms at the domain boundaries can either rely on a-priori knowledge (e.g., policy information) or they can apply precautionary protection means (e.g., encryption, message authentication codes) to bridge insecure or less secure domains.

We suggest integrity inspection for incoming data and secrecy checks for outgoing data at all interconnection points of a given domain (domain boundary):

- *Integrity*: Each message or message parameter is initially assigned to an integrity class, e.g. depending on message type and originating domain. This integrity class denotes the trustworthiness into any data included in this message with regard to processing these data by sensitive network or application functions. High integrity data can be processed by sensitive or less robust functions. Low integrity data might contain compromising arguments and should only be processed by very robust or less sensitive functions (cleared for low integrity data).
- *Confidentiality*: Each message or message parameter can be assigned to a secrecy class. Before leaving a domain, the secrecy requirements of each message are checked against the secrecy clearing of its "next hop" domain.

Before *sending* data into a neighbouring domain, the source domain A verifies whether the secrecy classification of these data is compatible with the secrecy clearance of the neighbouring domain and the transmission link between them. Thus, we validate confidentiality requirements before sending data.

Before *accepting* data from a neighbouring domain, the receiving domain verifies whether the integrity class of the data allows its processing within the domain. In doing so, we prevent internal sensitive control functions from being manipulated by compromised external data. This implements an acceptance check for incoming data.

Finally, we adapt (re-label) the secrecy and integrity class of data packets as soon as they are crossing domain

boundaries. Thereby, data can be re-assigned to a higher secrecy class when crossing domains with higher secrecy clearance or to a lower integrity class when crossing less trusted domains.

We suggest in this contribution a new framework and mechanisms to implement above re-labelling and filtering at domain entries (ingress) and domain exits (egress). Our approach does not introduce any per-flow states in the filters but rather accumulates access control (filtering) related data into the data packets itself. This allows highly distributed and efficient filtering as it supports fine-grained filtering (based on more criteria) nearer to the destination—and hence applied to a smaller amount of traffic.

II. RELATED WORK

Our work is based on the Bell/LaPadula secrecy model and the Biba integrity model for access control. Bell and LaPadula introduced in [3] general access control models based on secrecy classes. Denning [1] enhanced these models by introducing lattices, i.e., partially ordered sets of security levels, which represent the foundation for mandatory access control of today [7]. Biba introduced in [6] different access control schemes based on integrity levels. All these access control models include a so called reference monitor that mediates any access of subjects (entities) to objects (usually data) in a computer system. Reference monitors ensure that any access conforms to the access rights implied by the security policy.

Mandatory access control models, as opposed to discretionary access control models, prevent normal users from changing access rights; access rights are not at the discretion of users or programs that run on behalf of users. In this model, the reference monitor's decisions to grant or deny access are based on security classes assigned to objects and security clearances assigned to subjects. Sensitive applications might be allowed to process data with a high integrity level only. Applications processing top secret data might not be allowed to write processing results into unclassified objects.

The so called High Water Mark model (Weissman [4]) allows to dynamically classify objects. Weissman describes how to re-classify files by assigning them to the secrecy class that corresponds to the highest clearing of a subject that has written to it; because this subject could have added data with this classification to the file. For integrity models, we respectively re-assign an object to the integrity class that corresponds to the lowest integrity clearing of any subject that has accessed this object. In accordance to Weissman's High Water Mark model for secrecy, this procedure has been introduced by Biba [6] as the Low Water Mark integrity policy. Karger et al. propose in [5] an access control model for smart cards that implements both integrity and secrecy classes.

Generally, to prevent from unintended data leakage or processing of forged data, we raise secrecy classes or lower integrity classes when re-labelling.

We apply the mandatory access control model to guarantee confidentiality and integrity requirements regarding data flows spanning multiple interconnected networks.

Security labels store security classes and categories that are assigned to data packets. Security labels have been introduced for IP networks in [2]. Additionally, we introduce tags that store context information (e.g., incoming link, incoming protection such as IPSEC) that can be used for access control decisions in later processing stages (e.g. filters). We dynamically update classification information of data packets in access control devices at domain boundaries. General security issues at interconnection points have been presented in [10]. The use of security labels and context information in signalling networks (SS7 of the PSTN) has been introduced in [8].

An excellent reference for distributed system security represents the report by Rushby and Randell [15]. They introduced Trusted Network Interface Units (TNIU) that guard the access of workstations to the Local Area Network in order to implement multi-level secure distributed systems. Our work can be seen to extend their TNIU and to apply respective interfaces to form secure gateways for IP-based and SS7-based networks. For the theoretical foundation, we refer to [11]. In this contribution, we apply the general principle to protect network interconnection points of packet-switched networks.

Regarding distributed filtering, Steve Bellovin proposes a scheme [9] that pushes firewall filter functions even into user terminals (personal firewalls). We propose distributed filtering—supported by cumulated context information stored in data packets—at domain boundaries and aim at secure interconnection rather than user security. Our network mechanism is meant to work independently of users and user terminals.

III. DISTRIBUTED FILTERING ARCHITECTURE

At every interworking point between autonomous domains, a mediating device (reference monitor) must decide if certain packets may pass the network boundary. This access control monitor consists of *access control decision functions* and *access control enforcement functions*. In the next section we introduce the criteria that are applied as part of the decision functions. Furthermore we suggest a general structure of filters that enforce access control and the respective distribution aspects.

A. Security Labels and Contags

Security labels and contags store information that builds the foundation for access control decisions, e.g., if a packet is allowed to enter or to leave a domain. Security labels form the major criteria on which access control decisions are based.

Security labels store the security classes of a data packet within the packet itself. We distinguish between secrecy (or confidentiality) and integrity. The security policy assigns security and integrity clearances to domains. Data packets are initially assigned to the security class that corresponds to the clearing of the domain that creates them. The security class of a packet may change when it travels through different domains. We differentiate the following *secrecy* and *integrity* classes:

- S0* unclassified (unrestricted disclosure)
- S1* secret, e.g., payload or signalling data (to protect against profiling by third parties)
- S2* top secret, e.g., management passwords for remote access to network elements
- I0* no integrity guaranteed, e.g., user input
- I1* minimal integrity, e.g., user initiated connection control data
- I2* medium integrity, e.g., network internal data or data generated by trustworthy operators
- I3* high integrity, e.g., internal management (such as SNMP², OMAP³ messages) and tariff data

It is possible to define further classes; however, it must be confirmed that these classes are interpreted consistently by neighbored domains at interconnection points.

Security labels escort packets and can be used across network boundaries as long as their authenticity is assumed (trust) or actively protected by using security mechanisms, such as message authentication codes.

Contags are valid within a single administrative domain only—in contrast to security labels. Contags store context information that is derived from the environment (communication context). Examples for context information are the incoming link set or network, the time of arrival, the protection level on arrival (e.g., protected by IPSEC), or the identity of the authenticated sender if applies, e.g., IKE identities.

Usually, such context information is available only where it is generated and is used right there for filtering purposes. Unfortunately, the traffic load at such network nodes—e.g., IPSEC gateways and VPN servers or SS7 gateways that connect links of different network providers to the local network—is very high, which restricts the filter granularity at these points considerably.

Our technique accumulates context information (history) in the data packets using contags. These contags can be used by decision functions with finer granularity at a later point in time and at a location that is nearer to the destination of the data packet. This enables hierarchical filter stages that filter with increasing granularity on a decreasing number of packets.

Due to their impact on sensitive access control decisions, contags and security labels must be protected against manipulation when transported via insecure network domains. This can be achieved by message authentication codes (cryptographically protected message hash values). In addition to contags, conventional information elements within data packets can be used as input for decision functions. Examples are source and destination addresses, the message type, and parameter types (payload, signalling or management type, national parameters).

B. Building Blocks for Access Control at Interconnection Points

Based on security labels and contags, access control at interworking points between domains can be implemented by three consecutive stages.

The first stage (*entry validation*) decides which labels and tags are trusted by the systems within a domain. This decision depends on the security policy, which describes the trustworthiness of other domains (e.g., other operators). At the entry of a domain, the integrity class of a packet must be checked. Given a data packet P is classified by $I(P)$ and given the domain D is assigned clearance for integrity class $I(D)$, the data packet must be marked to be discarded in a following filtering stage (third stage, see below) if $I(P) < I(D)$ holds. Thus, sensitive applications within a domain are protected against potentially forged data. This validation is applied if the classifying information in the packet is trustworthy. Message authentication codes or signatures over the packet can be verified by this stage to establish trust. If trust in the data packet cannot be justified then the packet is marked automatically as non-authentic and will be re-labelled pessimistically in the second stage.

The second stage is responsible for *context tagging* and *label adaptation*. New contags can be added to the packet, e.g., as Security Options [14] in IP packet headers or in spare fields of SS7 signalling messages. Additionally, the security classes of the packet are adapted to reflect the updated history of the packet. If the local domain is cleared for secrecy class $S(D)$ and receives a data packet P of secrecy class $S(P)$ then the data packet is re-labelled as $S(P) := S(D)$. The relation $S(D) \geq S(P)$ holds since otherwise the packet would have been filtered at the exit of the sending domain. This pessimistic re-labelling is necessary since hosts within the domain could add data that is classified as $S(D)$ to the packet. A packet that is not authenticated but is received from a sending domain S with $I(S)$ is given the new integrity class $I(P) := \min(I(S), I(P))$. The default integrity classes of adjacent domains must be defined by the enterprise security policy; default classes must be chosen carefully.

A third stage (Filter) *filters* data packets according to their tags, labels, and other information contained in the packet. A domain has always both incoming and outgoing filters if the transit between the domains enables bi-directional traffic.

First, we describe the filtering at the entry of a domain. If a domain requires a higher integrity class than the packet offers (i.e., $I(D) > I(P)$) or a packet has been marked for discard by a preceding processing stage then the packet is discarded at the domain entry. The packets' contents may be logged in a secure place for later security audits, e.g., to verify whether business partners try to violate their service level agreements.

At the exit of a domain, packets are discarded if their secrecy class $S(P)$ is higher than the clearing of any domain that is a potential receiver of the packet (non-disclosure) or if it has been marked for discard by a preceding processing stage.

C. Distributed Filtering

Due to the structure of the suggested filter mechanisms and by including contags, we enable distributed implementations. In today's systems, filter functions are integrated into a single entity. Fig. 2a depicts such a

² SNMP: Simple Network Management Protocol

³ OMAP: Operation and Maintenance Application Part

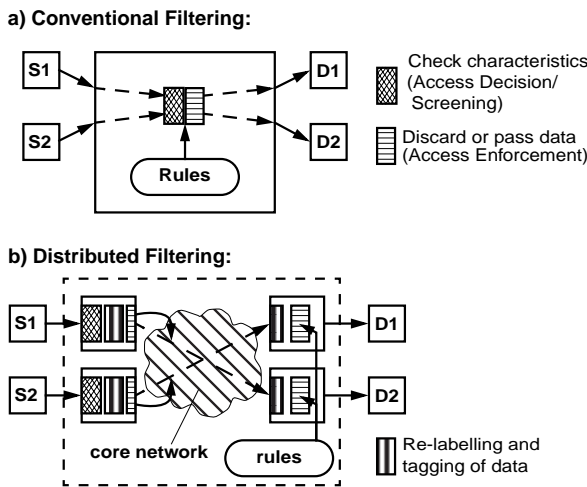


Figure 2: Centralized and Distributed Filtering based on Contags and Labels

conventional filter. All data packets received from the source domains S1 and S2 are investigated in order to determine the characteristics that are relevant for later access decisions. Thereafter, a specific data packet is forwarded or discarded according to decision rules which evaluate the characteristics found in this packet (labels, contags). This type of architecture can be found in firewall routers or packet filters. Implicit information, e.g., incoming link set, are invisible to consecutive filter stages.

Fig. 2b illustrates the distributed filtering. The screening function is located at the entry point of a domain. The filter functions are distributed. Filters for incoming and outgoing data can work on implicit information stored in contags.

Screening of the characteristics is performed immediately at the point of delivery between adjacent domains. Non-authentic data is marked. Subsequently the data is re-labelled according to the rules in section B. In the entry stage, integrity is checked. Data packets that might undergo processing within the domain⁴ must be discarded here if necessary. Before a packet is delivered to destination domains (D1 or D2), the confidentiality requirements are checked and the security policies defining the clearance of D1 and D2 is enforced. Data that has been generated within the domain is labelled according to the domain's secrecy and integrity clearance before it is handed over to the final exit filter stage.

Applying the procedures as proposed above enables us to postpone decisions about the allowance of information flows and to leave the decisions to later process stages closer to the destination. The later decision can be based on accumulated context information (history) that would be invisible without labels and contags. The secure transfer domain (dashed border line in fig. 2) must be trustworthy; otherwise integrity (and confidentiality if

applies) of data travelling on any enclosed communication path must be protected.⁵

D. Security Compatibility of Adjacent Domains

Usually, the suggested re-labelling introduced in the preceding chapter leads to over-classified packets. Assume a data packet that crosses a domain with a higher secrecy or lower integrity clearance than the security classes assigned to the packet. Even if the packet is not modified when passing the domain, the re-labelling at the exit of this domain or at the entry of the receiving domain will adjust the secrecy or integrity class of the packet. This is necessary because we usually cannot determine whether a packet has been modified or not (worst-case assumption). This over-classification leads to unintentional limitations with regard to the connectivity of remote domains.

To avoid unnecessary limitations on the communications, we consider measures that prevent from unnoticed access to data within transit domains (e. g., by means of message encryption or authentication).

We distinguish two cases concerning the compatibility of adjacent domains.

First, if data of the sending domain must be modified within the adjacent domain then the integrity-related rules of section B must be observed strictly. If the adjacent domain is just transferring the data then additive integrity protection (e.g., using message authentication codes) can be applied to bridge intermediate domains and avoid re-labelling.

Second, if the data's content must be read by the adjacent domain then the confidentiality-related rules of section B must be observed strictly. If the adjacent domain is just transferring the data then additive confidentiality protection (e.g., IPSEC, line encryptors) can be applied to bridge intermediate domains and avoid re-labelling.

In the latter case, the verification of security labels is based on the clearance of the domains that represent the end points of the protected tunnel.

- Non-trustworthy transfer domains do not cause a lowering of the data packets' integrity classification (at the entry of receiving domains) if the destination domain receives the data from the end point of a tunnel offering integrity protection. The fact of tunnelling can be stored in contags for a later evaluation. The re-labelling in the entry stage of a receiving domain (tunnel exit) takes these contags into account when determining the new integrity class of a received packet.
- Non-trustworthy transfer domains can be used to transmit confidential data if this data is tunnelled with confidentiality protection. This fact is considered when the packets are re-labelled at the exit of a sending domain (tunnel entry). The encrypted packet can be assigned to secrecy class S0. Thus, the exit filters can send these data packets through non-trustworthy domains.

⁴ This is the case, if the destination address is within the domain or if the full translation of the address is not known, e.g., global title addressing in SS7 networks.

⁵ Alternatively, the data packets can be re-labelled at the exit of the core network with the secrecy and integrity clearing of the core network (following a worst-case assumption). In this case, the core network must be assigned a uniform clearance; the lowest integrity and highest secrecy clearance of any system within the core network.

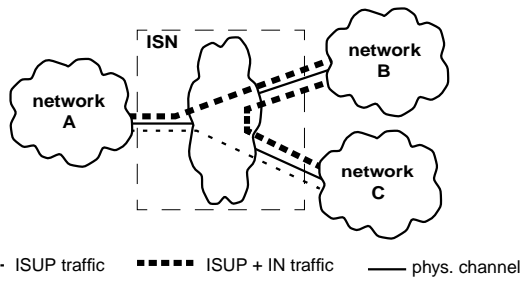


Figure 3: Interconnection Scenario

Whether or not protected channels must be applied can be retrieved from a centralized (locally cached) policy database. Whether received data packets were protected can be dynamically derived from the IPSEC policy database, from the rather static security policy data base, or from contags stored in the packet.

IV. EXAMPLE

As an example, we apply the introduced filtering principles to the interconnection of ISDN signalling networks of different network operators (A, B, and C). An intermediate SS7-based signalling network (ISN) that is controlled by network operator A connects the networks A, B, and C. Contracts between the operators define the traffic types allowed to cross the network boundaries. A accepts any ISUP⁶ signalling messages for call control from B and C. Access to A's higher layer network services (e.g., value-added services of the Intelligent Network) are restricted to provider B. The ISN shall allow the exchange of the respective higher layer signalling messages between networks B and A only (see fig. 3).

Due to the high complexity of upper layer protocols (e.g., Intelligent Network Application Protocol, INAP), it is infeasible to analyse respective signalling messages at the entry point of the ISN (unacceptable performance penalty). However, on delivery of a message to the destination network the filters can no more verify the sender of a message, since the correctness of the sender's address (Origination Point Code, OPC) cannot be checked at this stage.

Therefore, we apply distributed filtering with security labels and contags. We check the authenticity of the

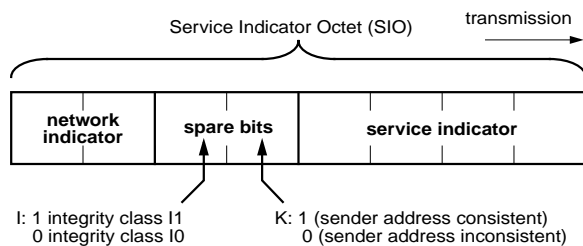


Figure 4: Storing Contags in the Service Indicator Octet of a Message Transfer Part Signalling Message

⁶ The ISDN User Part (ISUP) enables the control of circuit oriented communication in digital telephony networks. This is achieved with signalling messages that are exchanged using the message transfer part (MTP), a packet-switched network.

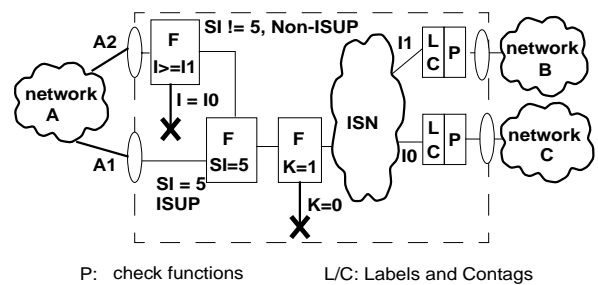


Figure 5: Distributed Filtering at the Interconnection Point of SS7 Signalling Networks

sender's address as soon as the message is received at the entry point of the ISN and annotate the result of the examination as contag K to the message. The considered messages are layer 3 messages of the SS7 (message transfer part, MTP). As depicted in fig. 4, security labels and contags are stored in spare bits of the service indicator octets of MTP messages

First of all, network A is partitioned into two logical subnetworks A1 and A2 that are connected to the ISN via dedicated access points (cf. fig. 5). Access point A1 handles ISUP traffic only (Service Indicator SI=5) all remaining traffic is directed via access point A2. Different integrity clearances (I0 and I1) are assigned to the networks B and C, respectively.

On receipt of a message by the ISN, the correctness of the sender's address (OPC) is validated according to the incoming link and the result is stored in the address header as a contag (K=1: consistent). Immediate filtering of the message is not possible, because there might exist agreements between B and C on the traffic between them that impose other requirements on the address space on a certain link⁷.

We store the integrity class of the originating network in the MTP message with an integrity label I (additional spare bit in the SIO, cf. fig. 4).

At the entry points of network A all packets with inconsistent OPCs can be discarded by filtering contags K=0 (protection from OPC spoofing). The remaining traffic is separated into ISUP and non-ISUP traffic by analysing the service indicator. For non-ISUP traffic another filter enforces the delivery of only integrity class I1 messages to the destination network. Additionally, this part of the traffic can be examined by network intrusion detection systems that are specialized on IN control messages. Finally, at the entry points to the destination network, those labels that will not be used any more are removed.

Thus, the application of contags and security labels enables distributed, intelligent filtering. Therefore, the partial opening of the network A towards other network operators is possible. Since the validity of all labels and contags is restricted to the ISN, the mechanism is transparent to other signalling protocols. Furthermore, there is no need to change the MTP protocol instances within the networks A, B, and C.

⁷ This is the case if the ISN offers signalling transfer point (STP) functionality for traffic between B and C. B might accept OPC1 and OPC2 while A only accepts OPC1. Thus, correctness for A can not be decided.

V. CONCLUSIONS AND OUTLOOK

The presented distributed filtering technique integrates local security mechanisms—such as IPSEC tunnels, network intrusion detection systems, and SS7 link protection—and security policies spanning multiple networks to protect interconnection points. It protects investments into existing network infrastructure and helps to solve well-known security problems at network interconnection points.

Additionally, distributed filtering enables the provisioning of sensitive subnetworks in a controlled manner and opens up the possibility to offer sensitive services globally. It particularly supports central network management of isolated networks (islands). The proposed technique prevents from threads originating from users' malpractice and from manipulated programs (Trojan Horses) within a domain as long as

- security policies are appropriately defined
- filters are installed at all interconnection points (mediate any traffic entering and leaving the protected domain)
- filter functions are implemented in a secure manner (secure hardware, secure software, low complexity)

Application scenarios for IP based networks (e.g., VPNs spanning multiple organizations or business partners) can be easily drawn from the example discussed in this contribution.

REFERENCES

- [1] D. E. Denning: A Lattice Model of Secure Information Flow. *Communications of the ACM*, Vol. 19, No. 5, May 1976.
- [2] S. Kent: Security Architecture for the Internet Protocol. S. Kent, R. Atkinson. November 1998. RFC 2401.
- [3] D. E. Bell, L. J. LaPadula: Computer Security Model Unified Exposition and Multics Interpretation, The MITRE Corp., ESD-TR-75-306, Ma., June 1975. (NTIS # AD A023588)
- [4] C. Weissman: Security Controls in the ADEPT-50 Time Sharing System. 1969 Fall Joint Computer Conference, AFIPS, Vol. 35, AFIPS Press, Montvale, N. J., 1969, pp. 119–133.
- [5] P. A. Karger, V. R. Austel, D. C. Toll: Using a Mandatory Secrecy and Integrity Policy on Smart Cards and Mobile Devices. EUROSMART Security Conference, June 2000.
- [6] K. J. Biba: Integrity Considerations for Secure Computer Systems. ESD-TR-76-732, HQ Electronic Systems Division, Hanscom AFB, Ma., April 1977.
- [7] R. S. Sandhu: Lattice-Based Access Control Models. *IEEE Computer*, Vol. 26, No. 11, November 1993, pages 9–19.
- [8] M. Kabatnik, R. Sailer: Modelling of Secure Interconnection. Communication Fraud Control Association, 1999 Spring International Conference in Ismaning/Germany, May 1999.
- [9] S. Bellovin: Distributed Firewalls. *login*, Nov 1999, pp. 37–39. URL: <http://www.research.att.com/~smb/papers/distfw.pdf>
- [10] K. Ward: The Impact of Network Interconnection on Network Integrity. *British Telecommunications Engineering*, Vol. 13, January, 1995.
- [11] R. Sailer, M. Kabatnik: History-based Distributed Filtering – A Tagging Approach to Network-level Access Control. Proc. 16th ACSAC, New Orleans, IEEE Computer Society, December 2000.
- [12] ITU-T Recommendation Q.704: Specification of Signalling System No. 7 — Message transfer part — Signalling network functions and messages; ITU, 1996.
- [13] B. Perlmutter, J. Zarkower: *Virtual Private Networks – A view from the trenches*. Prentice Hall, 2000.
- [14] S. Kent: Security Options for the Internet Protocol. Nov 1991. RFC 1108.
- [15] J. M. Rushby, B. Randell: *A Distributed Secure System*, Newcastle upon Tyne University (England), Computing Lab, TR-SER-182, 1982.