RZ 3493 (# 93600) 11/07/2001
Computer Science 16 pages

# Research Report

## A Routing Intrusion-Detection Scheme

Daniel Bauer, Marc Dacier, Ilias Iliadis and Paolo Scotton

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

# A Routing Intrusion-Detection Scheme

DANIEL BAUER, MARC DACIER, ILIAS ILIADIS and PAOLO SCOTTON
IBM Research, Zurich Research Laboratory
8803 Rüschlikon, Switzerland

Attackers may exploit the routing protocol in order to gain full control over data flowing through the network. It enables them to stop or redirect data traffic, intercept critical information, or modify data. Additionally they can corrupt the routing protocol, and completely stop network operations. This paper presents a new intrusion-detection system capable of warning network administrators of such attacks as well as of any abnormal behavior of the routing protocol by continuously monitoring the behavior of the network. The attacks considered are based either on the reachability prefix information advertised by the nodes participating in the protocol execution, on the rate by which elements are generated, or on the amount of topology-related information generated. The proposed model is capable of detecting abnormal behavior, such as misconfigurations that are not detectable by conventional network-management tools, and specific malicious attacks (e.g. denial of service and misrouting). It applies to the IS-IS, OSPF, and PNNI routing protocols because it uses a generic model of a network topology capable of dealing with link-state routing protocols. The scheme developed has been successfully tested on several large production networks.

## 1. INTRODUCTION

With the explosive growth of Internet connectivity and the pervasive access that users have to both internal and external networks, the number of attacks on corporate and government networks has increased tremendously. To cope with this problem, intrusion-detection methodologies and tools have been developed [Clyde 1998]. An intrusion-detection tool discovers attacks and threats throughout an enterprise, and responds to those discoveries. Such a system needs to be effective in catching attacks and, at the same time, it needs to limit false alerts. Today's complex enterprises need automated intrusion-detection tools. Based on the underlying methodology, current intrusion-detection products can be divided into three categories: post-event audit trail analysis, real-time packet analysis, and real-time activity monitoring [Clyde 1998]. Products of the first category analyze certain

UNIX audit trails for suspicious activity. Products of the second category have sufficient speed to detect attacks in real-time and respond immediately, ideally before damage is done. Products of the third category monitor security-related activities occurring on the various systems and devices that make up the network. This requires that agents cover systems and network devices throughout the enterprise. The work presented in this paper is concerned with attacks aimed at the routing protocol domain, and thus falls into the third category.

Most security problems are caused by buggy software. Flaws in software for routers, switches, and firewalls could give attackers complete control over the widely used equipment that supports the bulk of the Internet's backbone. Such a flaw was reported in [IDGNewsService 2001]; in this particular case, by requesting a particular URL from the server, a malicious user could bypass the authentication controls and execute commands on the device at the highest privilege level. Clearly, this vulnerability requires relatively little skill; an attacker can send a crafted URL and commands will be executed on the router. Once an attacker has gained access he or she could stop or redirect data traffic, intercept critical information, or modify the data. Additionally the attacker could inspect, change, or delete the device configuration, effectively disabling the router or switch until an engineer reprograms it. On the other hand, various Internet sites provide information on how to break into systems, and often include tools that automate the hacking process. Attacks on routing protocols have only very recently gained the attention of hackers. This was highlighted, for example, at the DEF CON Nine conference [DEFCON 2001], one of the largest hacker conferences worldwide, where there were presentations on how to break into routing protocols, and where also free downloadable tools for achieving this were included.

From the above it is evident that attackers can exploit the routing protocol in order to gain full control over data flowing through the network. It enables them to stop or redirect data traffic, intercept critical information, or modify data. Additionally they can corrupt the routing protocol and completely stop network operations. A variety of attacks on routing protocols have been described in [Wilson 2000; Vetter et al. 1997; Wang and Wu 1998]. In order to face these new threats, routing intrusion-detection tools are of vital importance for network managers. More recently, efforts have been made to apply intrusion-detection techniques at the routing protocol level to protect the network routing infrastructure. A comprehensive system featuring routing intrusion-detection of attacks against the standardized Open Shortest Path First (OSPF) routing protocol [Moy 1998] was presented in [Qu et al. 1998; Wang et al. 1999; Jou et al. 2000; Chang et al. 2001]. It uses a statistical intrusion-detection approach applied in the case of three OSPF insider attacks, namely, the *seq++, maxage*, and *maxseq* attacks. These attacks were subsequently implemented on the FreeBSD platform [Wang et al. 1999], and a network management architecture was proposed for their identification. The proposed system utilizes both misuse (protocol analysis) and statistical anomaly detection approaches. However, it does not analyze topological information such as advertised address prefixes. Consequently, this approach cannot detect attacks based, for instance, on advertising false reachability information as discussed in [Cosendai et al. 1999]. Our work addresses this issue. Attacks on the routing protocols can be

characterized as being either *insider* or *outsider* [Vetter et al. 1997; Baltatu et al. 2000]. An insider attack is caused by a trusted entity participating in the routing information exchange process, such as a subverted or compromised router, whereas an outsider attack involves an intruder masquerading as a router that distributes fabricated, delayed, or incorrect information. In general outsider attacks are prevented by authentication methods, although some forms of denial-of-service attacks are still possible [Etienne 2001]. These authentication methods, however, are insufficient for defending against insider attacks [Baltatu et al. 2000]. It is therefore desirable to develop detection methods for insider attacks. Interestingly, as outsider attacks are more limited in nature than the insider attacks, these methods will also serve for detecting external attacks.

The objective of this work is to present a scheme appropriate for the detection of insider attacks on link-state routing protocols. In particular, this work considers attacks on the Intermediate System to Intermediate System (IS-IS) protocol [Oran 1990], the OSPF protocol, and the Private Network–Network Interface (PNNI) protocol [ATM 1996], suggested by the International Organization for Standardization (ISO), the Internet Engineering Task Force (IETF), and the ATM Forum standard, respectively. It describes three new attacks based either on the reachability prefix information advertised by the nodes participating in the protocol execution, on the rate by which elements are generated, or on the amount of topology-related information generated. In the first case, the intruder's attack consists of injecting reachable addresses into the network that effectively overwrite already existing legitimate ones so as to misroute the traffic away from its original destination. In the case of a connection-oriented protocol, such as MPLS or ATM, the attacker can subsequently establish a connection and send the traffic to the original destination. The destination will not be able to realize that the traffic has been intercepted. To the best of our knowledge, this is the first work that makes use of the topological interpretation of the routing protocol. In the latter two cases, a malicious attack consists of an attempt to disrupt normal system operation either by creating excessive traffic or by exhausting the processing capabilities of the routers. According to the OSPF protocol, for example, element updates are sent every half hour in steady state. An attacker may create excessive traffic either by sending frequent updates or by frequently performing an add/remove operation of a given element. An attacker may also exhaust a router's memory and CPU capability by generating an excessive number of topology-information messages. As the same behavior also applies directly to the IS-IS and PNNI protocols, we shall focus on the OSPF protocol in the remainder of the paper, without loss of generality. This work also presents a new real-time intrusion-detection system capable of warning network administrators of such potential attacks as well as of any abnormal behavior by continuously monitoring the behavior of the network. The intrusion-detection system proposed requires no changes to the routing protocols themselves, and is capable of handling the new type of attacks described above as well as the `Seq++`, `MaxAge`, and `MaxSeq` OSPF insider attacks described in [Qu et al. 1998; Wang et al. 1999; Chang et al. 2001]. In contrast to the work presented in [Qu et al. 1998], which uses a statistical model, our model is deterministic because it relies on the nature of the routing protocols considered. Rather than behaving in a random fashion, these protocols

exhibit periodic patterns.

The outline of the paper is as follows. Section 2 presents some new potential routing-intrusion attacks, and Section 3 presents the method proposed for identifying such attacks in a dynamic environment, in which elements, such as nodes, links, metrics, and reachabilities, change with time because of additions and deletions. Section 4 presents the architecture of the Routing Services Platform (RSP) on which the routing intrusion-detection system is implemented. Section 5 presents the experimental results obtained by deploying the routing intrusion-detection system on several real large production networks. Finally, conclusions are drawn in Section 6.

## 2. ROUTING INTRUSION

This section examines the case of new attacks based on the reachability prefix information advertised by the nodes participating in the protocol execution. This information is taken into account along with the longest matching prefix criterion in order to determine the final destination. More specifically, every node advertises a set of reachable address prefixes. The prefix is represented by a series of bits and the number of significant bits (e.g., the 25 most significant bits of the 32-bit address 46.12.07.80 are denoted by 46.12.07.80/25). An address with a longer prefix (a larger number of significant bits) is called *overlapping address* (e.g. address 46.12.07.80/25 overlaps addresses 46.12.07.0/24 and 46.12.0.0/16 by one bit and nine bits, respectively). A node establishes a route to a certain destination by first identifying the corresponding destination node using the information contained in its database. The intruder's attack consists of injecting into the network, and consequently introducing in the database, reachable addresses with longer prefixes that effectively overwrite already existing legitimate ones so as to misroute the traffic away from its original destination. The same effect could occur by introducing duplicate reachable addresses with appropriate selected metrics. In the case of a connection-oriented protocol, such as MPLS or ATM, the attacker can subsequently establish a connection and send the traffic to the original destination. The destination will not be able to recognize that the traffic has been intercepted.

The following example, as depicted in Figure 1, demonstrates that the intrusion of the routing protocol is straightforward. On node A, the called address 46.12.07.01 matches the address 46.12.07/24 advertised by node B. Consequently, the traffic flows from node A to node B. Then, the intruder joins the network, as depicted in (b), by opening a connection on the routing channel to obtain the reachability information and the topology information. All the intruder now has to do, in order to divert the traffic, is to inject address information that overwrites already existing legitimate addresses. Thus, the intruder subsequently advertises the following two addresses 46.12.07.00/25 and 46.12.07.80/25, which exceed the address 46.12.07/24 of node B by one bit. This ensures that, according to the longest matching prefix rule, all traffic destined for address 46.12.07/24 of node B, will henceforth be destined for the intruder. The intruder now has several options: either steal the traffic and do nothing, or further forward the traffic to node B. The latter can be done only after the intruder has established a connection to node B, as depicted in (c). This implies that this is feasible in the case of MPLS or ATM, but not in the case

of the IP protocol.

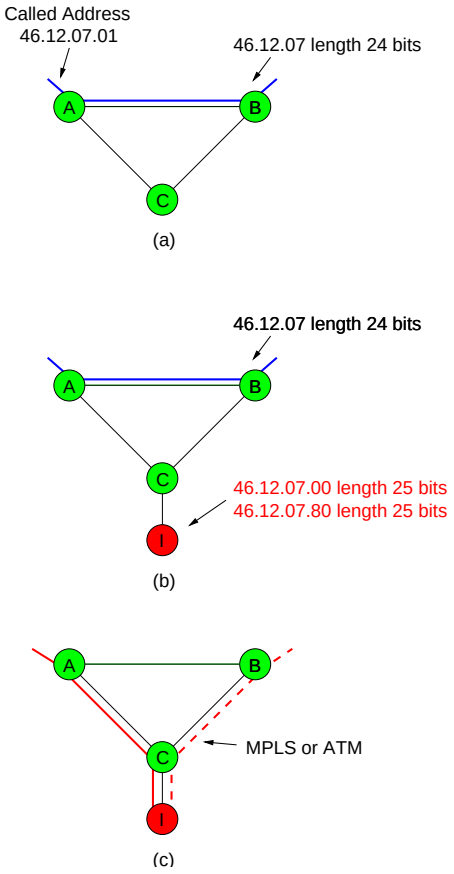


Fig. 1. Routing Intrusion.

From the above, it follows that two scenarios are possible:

## 2.1 Denial-of-service attack

In this attack, the malicious entity poses as a normal party and participates in all the necessary exchanges of the protocol. The intruder is able to observe the reachability information sent by other nodes and uses it to redirect the traffic. The false information sent by the intruder is a longer reachability. To redirect the traffic, the attacker stores and analyzes the reachable address prefixes and their length advertised by legal switches, extends the advertised prefixes (an extension of $n$ bits results in a space of $2^n$ reachable addresses), and finally advertises the constructed reachabilities as its own legal ones. The attack is effective as soon as a call is generated for an address that matches a usurped prefix. The intruder receives the call setup so that the connection becomes impossible. This leads to a denial-of-service situation.

### 2.2 Rerouting and sniffing

These capabilities are based on the basic attack as described above, which results in the misrouting of the traffic. The intruder reroutes the diverted traffic to its legal destination. This is done by opening a connection from the intruder to the destination node (this requires a connection-oriented protocol such as ATM or MPLS; IP is not appropriate because it does not fall into this category). The connection from the source to the actual destination is then achieved in two steps. To provide the connection service, the intruder has to forward all traffic coming from the source to the destination, and vice versa. In fact, it acts as a switch between the connection from the source to itself, and the connection from itself to the destination. The intruder stays transparent to the destination, in order to sniff the passing traffic.

The intruder achieves its goal by taking the following actions. It stores the properties of an advertised reachability, detects a call setup for a usurped address, obtains the call parameters, opens a connection to the destination, forwards the flow to the destination, manages the incoming and outgoing connections (traffic from the source to the destination), and finally exploits the diverted traffic. Storing the properties of an advertised reachability is necessary for knowing to whom to reroute the traffic. Detection of the call setup and acquisition of the call parameters are used to create the connection reaching the actual destination with the properties of the incoming one.

## 3. INTRUSION-DETECTION MODEL

Based on the description of the attack given in the preceding section, a signature-based intrusion-detection scheme would require that an alarm be raised each time an overlapping or duplicate address is detected. However, the experimental results reported in Section 5 suggest that such activity may be normal, and not necessarily associated with a malicious intent. Consequently, to take this into account, the routing intrusion-detection scheme proposed comprises two parts. The first part consists of a learning phase in which duplicate and overlapping addresses are identified and registered. Note that during this phase, potential misconfiguration errors may be identified. The second part is an intrusion-detection phase in which the system is continuously being monitored and newly introduced addresses are checked against the existing ones. Alerts are raised only if new overlapping and duplicate addresses are detected, and, subsequently, potential malicious attacks as well as misconfiguration errors are identified. During the detection phase the system also checks for any abnormal behavior caused in particular by an excessive generation rate or an excessive amount of topology-related information. In this way, it is argued below that the system is also capable of detecting the `Seq++`, `MaxAge`, and `MaxSeq` types of attacks. Consequently, this results in a behavior-oriented intrusion-detection system. It is also a host-based system because it analyzes events occurring in the routing protocol execution as well as a knowledge-based system because the attack specifics are used in the detection process. Finally, note that our approach requires no changes to the routing protocols themselves.

### 3.1 Intrusion alerts

The routing detection model checks for the events described below, and appropriate warnings and alerts are generated and sent to the network administrator.

3.1.1 *Duplicate addresses.* Each newly introduced address is compared with the existing ones. If it already exists and has not been registered during the learning phase, an alert is generated.

3.1.2 *Overlapping reachabilities.* Each newly introduced address is compared with the existing ones. The number of bits providing a longer prefix is called the *exceeding factor*, or *e-factor*. In the case of IPv4, the length of a reachable address prefix ranges from 0 to 32 bits. In the case of ATM addresses, the length of a reachable address prefix ranges from 1 to 160 bits, with the latter value representing the complete address of an end system. Consequently, the range of an exceeding factor is from 1 to 159 bits and from 1 to 32 bits, for IPv4 and ATM, respectively. For example, the exceeding factor of address 11.22.33.44/27 over address 11.22.33.44/24 is *e-factor* = 3.

Two cases are considered, depending on the overlapping address. If the new address overlaps an existing one, an *e-INTRUSION* alert is raised provided this address combination has not already been registered during the learning phase. Upon receiving such an alert, the origin of the potential attack can be identified such that the network administrator can take appropriate counter-measures. On the other hand, if an existing address overlaps the new address, an *e-overlap* alert is raised provided this address combination has not already been registered during the learning phase. Although an e-overlap-type alert is never associated with a direct intrusion attempt, it may be associated with an indirect one. The intruder's attack in this case consists of introducing a new address without raising suspicion (as the newly added addresses is exceeded by an existing one), and subsequently activating the new address by attacking and disabling the original address. An e-overlap-type alert is generated because it may also be related to a misconfiguration error.

3.1.3 *Excessive-rate alert.* According to the OSPF protocol, element updates are sent every half hour in steady state. The update rate of each existing element is monitored. If it exceeds a given threshold within a specified window, an alert is generated. In addition, the frequency of the performed add/remove operations is monitored and if it exceeds a given threshold, an alert is generated. An alert is also generated if the sequence number of any element is found to be equal to the maximum possible value allowed.

3.1.4 *Excessive-amount alert.* The number of the different link-state advertisements on each node is monitored. If it exceeds a given threshold, an alert is generated.

3.1.5 *Sequence-number-gap alert.* In addition to the above-mentioned alerts, the intrusion-detection system also included the sequence-number-gap alert. According to the OSPF protocol, element updates are sent by increasing the sequence number by one. If topology-information elements that carry nonconsecutive sequence numbers arrive, an alert is generated. For the reasons given in Section 5, this type of alert has been removed.

### 3.2 Additional attacks

In this section we briefly review the `Seq++`, `MaxAge`, and `MaxSeq` types of attacks presented in [Qu et al. 1998; Wang et al. 1999; Jou et al. 2000; Chang et al. 2001]. We also demonstrate how the these attacks are identified through the alerts described above.

3.2.1 `Seq++` *attack.* An attacker modifies the link-state metric and increases the sequence number of a particular link-state advertisement (LSA) by one. The updated LSA is flooded and eventually reaches the originator of the original LSA. According to the OSPF specification, the originator then generates an updated LSA carrying the correct link status information and a fresher sequence number. If the attacker keeps generating such LSAs, this would cause instabilities in the network topology. Clearly, this type of attack increases the update rate of the particular LSA, and is therefore detected through an excessive-rate alert, as described in Section 3.1.

3.2.2 *Maximum-Age attack.* An attacker modifies the maximum age value of a particular LSA by setting it to one hour and re-injects it into the system. The updated LSA, with its sequence number unchanged and its maximum age corrupted, will cause all routers to delete the corresponding LSA from their topology database. Eventually the originator of this purged LSA will also receive the MaxAge LSA, and will subsequently generate a new LSA carrying a fresher sequence number. Here the effect is similar to the `Seq++` attack. Moreover, this type of attack increases the frequency of the performed add/remove LSA operations and is therefore detected through an excessive-rate alert, as described in Section 3.1.

3.2.3 *Maximum-sequence-number (MaxSeq) attack.* An attacker modifies the link state metric of a particular LSA and sets its sequence number to 0x7FFFFFFF, i.e. the maximum sequence number. The updated LSA is flooded in the network and eventually reaches the originator of the original LSA. This LSA is now considered the most recent by other routers, as it has the maximum LSA sequence number. The originator in turn first deletes the LSA (setting MaxAge) and then floods an updated LSA carrying correct link status information and the smallest sequence number: 0x80000001. In this case the effect is similar to the `Seq++` attack. In some implementations, however, routers may not support the delete of the MasSeq LSA, implying that it will stay in their topology database for one hour before it reaches its MaxAge. Consequently, this results in a disturbance of the routing stability. In our system, if an LSA is detected with its sequence number equal to the maximum possible value, an alert is generated as described in Section 3.1.3.

### 3.3 Learning and detection phases

During the learning phase, the application is fed with all the relevant information. Its duration is chosen by the system administrator and is typically of the order of a few minutes, depending also on the size of the network. The alerts generated could help the administrator locate potential misconfigurations and also malicious attacks. The system has a memory and registers such alerts. This information is subsequently used during the detection phase for cross-checking new alerts.

In a previous work [Cosendai et al. 1999] it was suggested that for each of the

nodes the following three indicators be used. The first indicator related to the exceeding factors of all reachabilities associated with a node, and an alert was raised if this indicator exceeded a specified threshold value. The second indicator related to the number of overlapping reachabilities associated with a node, and an alert was raised if this indicator exceeded a specified threshold value. The third indicator related to the frequency by which reachabilities associated with a node change, and an alert was raised if this indicator exceeded a specified threshold value. The above indicators were introduced to filter and reduce the number of alerts. Note that the model presented here does not make use of any of the above indicators. This is due to the fact that, regarding the first two indicators, our model raises an alarm at every instant a new, exceeded or overlapping, address is introduced, which therefore yields a very robust scheme. Regarding the third indicator, it turns out that the frequency by which reachabilities associated with a node change varies widely and therefore is not well suited for intrusion-detection purposes.

## 4. IMPLEMENTATION OVERVIEW

This section addresses the issue of the architecture and implementation of the routing intrusion system proposed. The Routing Services Platform (RSP) architecture uses a two-layer approach. The lower layer is responsible for gathering topology information, whereas the higher layer consists of the routing intrusion-detection core. An overview of the architecture is shown in Figure 2.

The architecture of the lower layer is based on the property of the link-state routing protocols whereby all nodes within their domain have the same view. We call this domain 'link-state domain'. A link-state domain such as an OSPF or IS-IS area or a PNNI peer group is monitored by a single 'Topology Feeder'. A topology feeder forms adjacencies with one or more routers in the link-state domain to gather topology information. It executes the link-state routing protocol and participates in the link-state database synchronization. The continuous flooding procedure of the underlying link-state routing protocol ensures that the topology feeder has an up-to-date view of the topology. In that respect, the topology feeder resembles any other router in the link-state domain. An important difference, however, is that the topology feeder does not generate topology information itself, i.e. it does not generate OSPF or IS-IS router link-state advertisement or PNNI nodal information to describe itself. Thus, the topology feeder remains invisible to the path-computation modules of the routers.

The topology feeder creates a routing-protocol-independent description of the underlying topology. Each link-state message that is flooded by the underlying routing protocol is immediately translated into a generic topology-information element, which is subsequently forwarded to the intrusion-detection core.

### 4.1 Representation of topology information

The topology feeder translates the topology information from the routing-protocol-specific format into a more generic format using the Extensible Markup Language (XML) language. The feeder produces a stream of so-called virtual topology-information elements (VTIEs), which consist of node, link, reachability, and the corresponding metric information. Each VTIE possesses a unique identifier to identify itself and to reference other VTIEs. A node describes either a router or, in

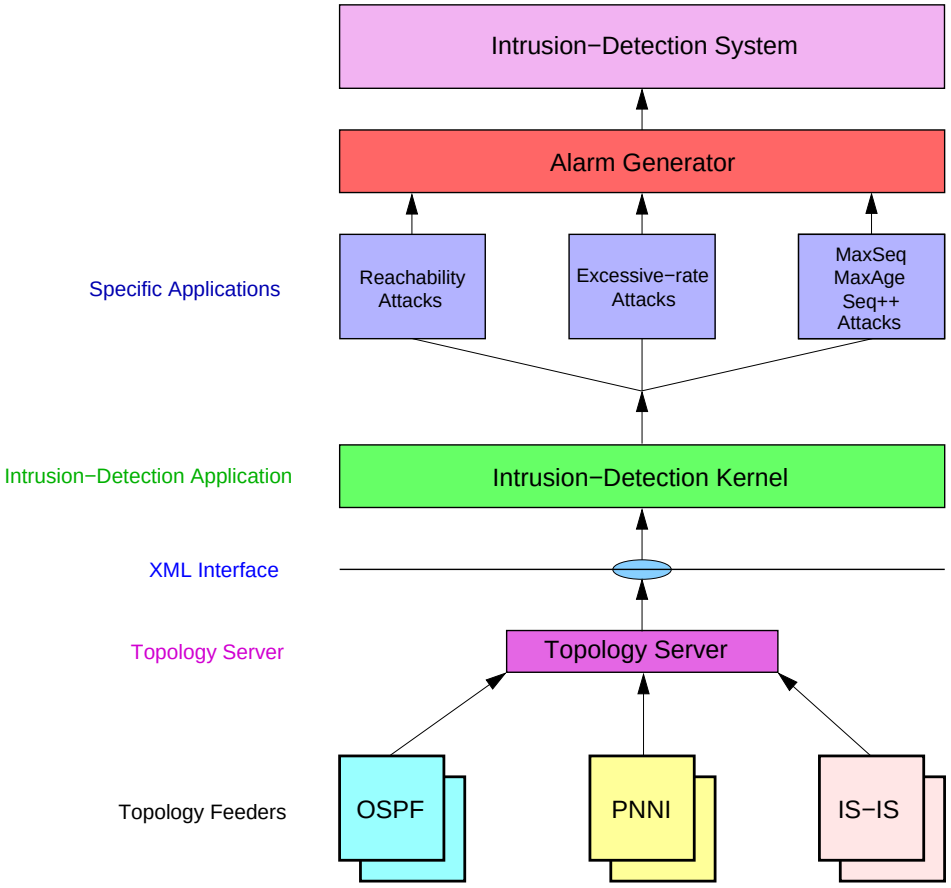


Fig. 2. Architecture Overview.

the case of OSPF and IS-IS, a broadcast network that connects multiple routers. Links describe adjacencies between nodes and are unidirectional only. For each connected node pair, a link in each direction is created. Reachabilities are associated with nodes and define address prefixes that are reachable by the associated node. Finally, metric information provides static and dynamic metrics for both reachabilities and links. In addition, each VTIE is attributed with routing-protocol-specific information such as sequence number, time-to-live (age), and advertising router. An example topology description is given in Appendix A.

### 4.2 Topology server

Large networks consist of many link-state domains. For each link-state domain, a single topology feeder is required. It is the task of the 'Topology Server' to correlate the topology description of multiple feeders and to produce a consistent description of the overall topology.

In the following, we consider the example of a topology server receiving topology information of multiple OSPF areas. The topology server creates a unified description of the overall topology by 'splicing' together the topology along the OSPF area border routers. This merged view of the topology can no longer be distinguished from a topology of a single, large OSPF area. Splicing is done by using two pieces of information. Firstly, topology information describing nodes and links contains the area ID in which they reside. OSPF border nodes belong to two or more areas[1] and are appropriately marked. Secondly, VTIEs imported from another area are marked 'imported', in contrast to locally 'generated' VTIEs. The topology server uses the area ID and node address to uniquely identify each OSPF border node. Links and reachabilities are then associated with a border node, even though their description has been generated by different feeders. Furthermore, the topology server filters out VTIEs that are marked 'imported' by one feeder and 'generated' by another feeder.

### 4.3 Intrusion-detection core

As depicted in Figure 2, the core intrusion-detection system is based on a modular architecture that allows a variety of intrusion-detection schemes for various attacks such as reachability attacks and sequence number attacks to be supported. The intrusion-detection kernel uses primitives fed by the topology server to keep track of the activity and state of the network. More specifically, the intrusion-detection application is aware of node and link-state information as well as of packet-header information. Furthermore, this application is registered to receive update events regarding topology changes from the topology server. This information is then processed by the specific applications, and an alarm is generated whenever a potential intruder is detected.

## 5. EXPERIMENTAL RESULTS AND ANALYSIS

The routing intrusion-detection system developed has been successfully tested and deployed on several actual OSPF-based production networks. Large networks, comprising about 100 nodes, were selected such that the system proposed could be tested under heavy user activity in a variety of challenging dynamic network environments. In all of these networks the routing intrusion-detection system has shown its effectiveness in detecting misconfigurations and other functional anomalies that had remained undetected for long periods of time. Moreover, because of its real-time analysis capabilities, the system was able to detect transient anomalies that are undetectable by conventional network-management tools. This is because, in contrast to network-management tools which typically look at the status resulting from the protocol execution (e.g. state of the routing tables and of connectivity), our system monitors the execution itself, thus providing a better insight.

The routing detection model generated appropriate warnings and alerts associated with the events described below, and sent them to the network administrator. In every deployment the alerts raised, as turned out in the subsequent analysis, were not due to any malicious activity. This is a typical behavior of intrusion-detection

[1] In contrast to OSPF, PNNI nodes always belong to a single peer group, whereas links may belong to multiple peer groups.

systems; when first deployed, they detect misconfigurations rather than malicious activities. Interestingly, though, it revealed to the network administrators misconfigurations that passed unnoticed for long periods. Based on the experience gained and the outputs observed, a short discussion follows each of the events regarding possible non-malicious reasons for causing them. Note that each node is accompanied by the address of the corresponding advertising node in parenthesis (e.g. node: 11.22.33.44 (adv. 14.11.11.55)), to allow their identification because nodes may also represent transit networks.

### 5.1 Duplicate addresses

Each newly introduced address was compared with the existing ones. If it already existed, an alert was generated, falling into one of the following four categories, as shown by the examples given below.

**Case 1:** Duplicate addresses on a single node.

```
DUPLICATE address: 12.55.66.77/24 on node 12.43.100.154 (adv. 12.43.100.154)
```

This indication corresponds to the case where the address 12.55.66.77/24 is advertised by an area border router that receives this address from two different areas.

**Case 2:** Duplicate address on two distinct nodes.

```
DUPLICATE address: 11.22.33.44/27 on nodes: 55.66.0.7 (adv. 55.66.0.7)
                                        and  23.45.0.8 (adv. 23.45.0.8)
```

This indication can correspond to the case where address 11.22.33.44/27 appears to both nodes, 55.66.0.7 and 23.45.0.8, with two different metrics for reliability purposes. If the primary node fails, the other one would be used.

It can also correspond to the case where 55.66.0.7 and 23.45.0.8 are two routers that both have an interface in network 11.22.33.44/27 but do not have OSPF running on these interfaces. Network 11.22.33.44/27 may also not run OSPF for security reasons. Packets are delivered to this network using the nearest interface.

It can also be the case that address 11.22.33.44/24 is associated with an old static route that has possibly been forgotten on one of the two nodes 55.66.0.7 and 23.45.0.8. This alert helps check for such misconfiguration errors.

**Case 3:** Duplicate address on two nodes advertised by the same node.

```
DUPLICATE address: 11.22.33.44/24 on nodes: 11.22.33.44 (adv. 14.11.11.55)
                                        and  14.11.11.55 (adv. 14.11.11.55)
```

This indication can occur when router 14.11.11.55, to which the transit network 11.22.33.44 is connected, reboots.

**Case 4:** Duplicate address on two nodes advertised by different nodes.

```
DUPLICATE address: 11.22.33.44/24 on nodes: 11.22.33.44 (adv. 12.34.66.77)
                                        and  14.11.11.55 (adv. 14.11.11.55)
```

This indication can occur when the transit network 11.22.33.44 changes designated router from 12.34.66.77 to 14.11.11.55.

### 5.2 Overlapping reachabilities

Several alerts of the form shown below were raised.

```
**** e-INTRUSION ALERT for address:
         11.22.33.44/27 on node: 123.234.20.0 (adv. 123.234.20.0)
   over  11.22.33.44/24 on node: 11.22.33.44 (adv. 14.11.11.44),   e-factor = 3

**** e-overlap   ALERT for address:
         11.22.33.44/29 on node: 14.11.11.55 (adv. 14.11.11.55)
   over  11.22.33.44/27 on node: 123.234.20.0 (adv. 123.234.20.0), e-factor = 2
```

The former indication corresponds to the case in which the newly added address 11.22.33.44/27 exceeds the existing one 11.22.33.44/24, whereas in the latter case the converse holds, i.e. the newly added address is exceeded by the existing one 11.22.33.44/29.

Such overlapping addresses are often introduced for reliability and backup purposes similar to those described above in the case of duplicate addresses.

### 5.3 Excessive-rate alert

In some of the networks, alerts of the form shown below were raised.

```
**** Excessive-Rate ALERT for reachability
    11.22.33.44/24  on node 14.11.11.55  (20 records in window of 2000 seconds)
```

This alert may be due to route flaps. It usually appears when address 11.22.33.44 represents a site to which a connection is realized by either a VPN over the Internet or an on-demand ISDN connection.

### 5.4 Sequence-number-gap alert

In addition to the above-mentioned alerts, the intrusion-detection system also generated alerts when topology-information elements with nonconsecutive sequence numbers arrived. A possible cause for that is that the OSPF protocol imposes a minimum period of 5 seconds between the generation of successive, distinct topology-information elements. Therefore, if changes in topology occur more frequently, for instance because of a flapping routing interface, elements with successive sequence numbers are formed but not distributed because of this constraint. The field trials showed that the generation of topology-information elements with nonconsecutive sequence numbers is quite common. Consequently, this phenomenon does not constitute any real threat. Therefore, this type of alert has been removed from the system.

## 6. CONCLUSIONS

This paper presented a routing intrusion-detection scheme developed for coping with new type of attacks on the link-state routing protocols. We have demonstrated how an attacker can exploit the routing protocol to gain full control over data flowing through the network. We presented a new real-time behavior-oriented intrusion-detection system capable of warning network administrators of such potential malicious attacks (e.g. denial of service and misrouting) as well as of any

abnormal behavior, including misconfigurations that are undetectable by conventional network-management tools and, consequently, pass unnoticed for long periods. It applies to the IS-IS, OSPF, and PNNI routing protocols because it uses a generic model of a network topology capable of dealing with link-state routing protocols. The attacks considered are based either on the reachability prefix information advertised by the nodes participating in the protocol execution, on the rate by which elements are generated, or on the amount of topology-related information generated. The scheme developed has been successfully tested on the Routing Services Platform deployed on several large dynamic production networks. From the alerts raised by the model, useful observations were derived regarding the nature of events occurring in such realistic environments and responsible for triggering these alerts. Based on the outputs obtained, it turns out that the model proposed can be further improved and refined to reduce the number of false alerts generated. This can be achieved by making use of correlations observed among series of alerts that turn out to be associated with normal routing protocol behavior.

## A. EXAMPLE TOPOLOGY DESCRIPTION

The following XML code describes a simple OSPF area with area ID 10.14.0.0 that consists of two connected nodes. The node with address 10.14.254.1 has an interface on subnetwork 10.14.40.0/25. Also, this node is an area border node that imports the address prefix 130.104.242.30/32.

```
<VTIE action="add" vtieid="1">
   <Node address="10.14.254.1">
      <Area>10.14.0.0</Area>
      <Area>border</Area>
      <Proto timestampsec="991731737" netproto="IPv4" routeproto="OSPF"
             seqno="2147484420" ttl="2924" origin="10.14.254.1"
             origintype="generated"/>
   </Node>
</VTIE>

<VTIE action="add" vtieid="2">
  <Node address="10.14.0.1">
    <Area>10.14.0.0</Area>
    <Proto timestampsec="991731738" netproto="IPv4" routeproto="OSPF"
           seqno="2147501367" ttl="2654" origin="10.14.0.1"
           origintype="generated"/>
  </Node>
</VTIE>

<VTIE action="add" vtieid="3">
  <Link srcnodeid="1" srcportid="0" dstnodeid="2" dstportid="2187919066">
    <Area>10.14.0.0</Area>
    <Proto timestampsec="991731738" netproto="IPv4" routeproto="OSPF"
           seqno="2147484420" ttl="2924" origin="10.14.254.1"
           origintype="generated"/>
  </Link>
```

```
</VTIE>

<VTIE action="add" vtieid="4">
  <Link srcnodeid="2" srcportid="2187919066" dstnodeid="1" dstportid="0">
     <Area>10.14.0.0</Area>
     <Proto timestampsec="991731738" netproto="IPv4" routeproto="OSPF"
            seqno="2147501367" ttl="2654" origin="10.14.0.1"
            origintype="generated"/>
  </Link>
</VTIE>

<VTIE action="add" vtieid="5">
  <Reachability objectid="1" portid="0" prefix="10.14.40.0" length="25">
    <Proto timestampsec="991731738" netproto="IPv4" routeproto="OSPF"
           seqno="2147501242" ttl="2786" origin="10.14.0.1"
           origintype="generated"/>
  </Reachability>
</VTIE>

<VTIE action="add" vtieid="6">
  <Reachability objectid="1" portid="0" prefix="130.104.242.30" length="32">
    <Proto timestampsec="991731918" netproto="IPv4" routeproto="OSPF"
           seqno="2148429664" ttl="3598" origin="130.104.1.15"
           origintype="imported"/>
  </Reachability>
</VTIE>


<VTIE action="add" vtieid="7">
  <Metric objectid="3" direction="outgoing" weight="100">
      <Proto timestampsec="991731738" netproto="IPv4" routeproto="OSPF"
             seqno="2147484420" ttl="2924" origin="10.14.254.1"
             origintype="generated"/>
    </Metric>
</VTIE>

<VTIE action="add" vtieid="8">
  <Metric objectid="4" direction="outgoing" weight="100">
    <Proto timestampsec="991731738" netproto="IPv4" routeproto="OSPF"
           seqno="2147501367" ttl="2654" origin="10.14.0.1"
           origintype="generated"/>
  </Metric>
</VTIE>
```

## REFERENCES

ATM. 1996. The ATM Forum: Private Network-Network Interface Specification Version 1.0. Specification Number af-pnni-0055.000, March 1996.

Baltatu, M., Lioy, A., Maino, F., and Mazzocchi, D. 2000. Security issues in control, management and routing protocols. In *Proc. TERENA Networking Conf., Lisbon, Portugal, May 2000*.

Chang, H., Wu, S., and Jou, Y. 2001. Real-time protocol analysis for detecting link-state routing protocol attacks. *ACM Transactions on Information and System Security (TISSEC)*.

Clyde, R. 1998. Intrusion detection methodologies (IT security). *Business Continuity (UK) 6,* 3, 22–26.

Cosendai, Y., Dacier, M., and Scotton, P. 1999. Intrusion-detection mechanism to detect reachability attacks in pnni networks. Presentation at the $2^{nd}$ *Int'l Workshop on Recent Advances in Intrusion Detection, RAID '99*, West Lafayette, Indiana.

DEFCON. 2001. DEF CON Nine Conference, Las Vegas, NV, July 13-15, 2001.

Etienne, J. 2001. Flaws in packet's authentication of OSPFv2. IETF draft-etienne-ospfv2-auth-flaws-00.txt.

IDGNewsService. 2001. Software flaw opens Cisco devices to hackers. http://www.nwfusion.com/news/2001/0629cishack.html.

Jou, Y., Gong, F., Sargor, C., Wu, X., Wu, S., Chang, H., and Wang, F. 2000. Design and implementation of a scalable intrusion detection system for the protection of network infrastructure. In *Proc. DARPA Information Survivability Conf. and Exposition, 2000 (DISCEX '00)*. Vol. 2. IEEE Computer Society Press, 69–83.

Moy, J. 1998. OSPF Version 2. RFC 2328, April 1998.

Oran, D. 1990. OSI IS-IS Intra-domain Routing Protocol. RFC 1142, February 1990.

Qu, D., Vetter, B., Wang, F., Narayan, R., Wu, S., Jou, Y., Gong, F., and Sargor, C. 1998. Statistical anomaly detection for link-state routing protocols. In *Proc. Sixth IEEE Int'l Conf. on Network Protocols, Austin, TX, October 1998*. 62–70.

Vetter, B., Wan, F., and Wu, S. 1997. An experimental study of insider attacks for OSPF routing protocol. In *Proc. IEEE Int'l Conf. on Network Protocols, Atlanta, GA, October 1997*. 293–300.

Wang, F., Gong, F., Wu, F., and Narayan, R. 1999. Intrusion detection for link state routing protocol through integrated network nanagement. In *Proc. IEEE Eight Int'l Conf. on Computer Communications and Networks*. 634–639.

Wang, F. and Wu, S. 1998. On the vulnerabilities and protection of OSPF routing protocol. In *Proc. IEEE 7th Int'l Conf. on Computer Communications and Networks*. 148–152.

Wilson, C. 2000. Protecting network infrastructure at the protocol level. SANS Institute, Information Security Reading Room, http://www.sans.org/infosecFAQ/threats/protocol_level.htm.