DecIdUouS: Decentralized Source Identification for Network-Based Intrusions*

H.Y. Chang, R. Narayan, S.F. Wu, B.M. Vetter, X. Wang, M. Brown, J.J. Yuill
Computer Science Department
NC State University
{wu}@csc.ncsu.edu

C. Sargor, F. Jou, F. Gong
Advanced Networking Research
MCNC
{sargor}@mcnc.org

Abstract

DECIDUOUS is a security management framework for identifying the sources of network-based intrusions. The first key concept in DECIDUOUS is *dynamic security associations*, which efficiently and collectively provide location information for attack sources. DECIDUOUS is built on top of IETF's IPSEC/ISAKMP infrastructure, and it does not introduce any new network protocol for source identification in a single administrative domain. It defines a collaborative protocol for inter-domain attack source identification. The second key concept in DECIDUOUS is the management information integration of the intrusion detection system (IDS) and attack source identification system (ASIS) across different protocol layers. For example, in DECIDUOUS, it is possible for a network-layer security control protocol (*e.g.,* IPSEC) to collaborate with an application-layer intrusion detection system module (*e.g.,* IDS for the SNMP engine). In this paper, we present the motivations, design, and prototype implementation of the DECIDUOUS framework.

Key Words:

Security Management, Integration of Network Control and Management, Monitoring, Event and Fault Handling.

*This research project is supported by DARPA/ITO under contract F30602-98-C0249.

1 Introduction

As more and more business opportunities are created over the Internet, dealing with *network-based intrusions* against those critical information/business services has been an important goal for network security. In the past few years, the network security community has made good progress in the development of attack prevention and intrusion detection technologies. These security services are very valuable in protecting hosts against network-based attacks launched remotely by an intruder. However, identifying the source of such attacks remains difficult. Today, a victim must rely on tools not intended for this purpose, such as `traceroute` and `finger`, which are at best only minimally suitable for tracing an attack source.

Another big challenge in attack source identification is to track down the intruder's location across multiple administrative domains. A simple example will be to identify an attack source from another RNP (Regional Network Provider) or ISP (Internet Service Provider). Today, a case like this will usually involve human intervention. By the time system administrators from different organizations start to cooperate, a sophisticated intruder may have escaped either logically or physically.

The objective of the DECIDUOUS project is to securely, practically and systematically identify attack sources by utilizing existing network security protocols and services. Specifically, the IPSEC authentication service [4, 2, 3] is used by the security management module to trace the source of an attack. The DECIDUOUS system deduces the source identification information from the end-point locations of the current security associations in the attacking packets. This leads to the concept of **dynamic security associations (SAs)** in DECIDUOUS: *in order to efficiently identify the attack sources, DECIDUOUS will* **dynamically decide** *where and when to establish security associations through IPSEC/ISAKMP* [5]. Thus, DECIDUOUS does not introduce any new protocol under a single administrative domain. And we only need to run DECIDUOUS as a daemon process on network entities we would like to protect.

In this paper, we will discuss the design of the DECIDUOUS framework. In Section 2, we present different classes of network-based intrusions the relations among *intrusion detection system (IDS), attack source identification system (ASIS)*, and *intrusion damage control system (IDCS)*. In Section 3, we will introduce the concept of dynamic security associations. In Sections 4, 5, 6, 7, and 8, the design and a very preliminary prototype of the DECIDUOUS framework is presented. We will first demonstrate the algorithm under a linear network topology, and then, we show how to handle a general network topology. Finally, we compare our approaches with other related works.

2 Intrusion Detection, Attack Source Identification and Response

Network-based intrusions can be classified into three different types: *simple attacks*, *network infrastructure/service attacks*, and *compound attacks*. For simple network-based intrusions (*e.g.,* SynFlood), an intruder merely *uses the available network services* to launch attack packets against some victims being connected to the Internet. On the other hand, an intruder/insider can directly attack the network infrastructure itself to disrupt or maliciously control the network services. A good example is the MAI router incidence in April, 1997, where a faulty BGP router de-aggregated and advertized thousands of network addresses which caused much of the Internet to be disconnected from 20 minutes to 3 hours. The MAI incidence implies that potentially an intruder can selectively de-aggregate certain routes to absorb/intercept all the traffic destinated to those target networks. In our lab, we have successfully demonstrated that, as long as we can intercept OSPF packets coming-in and going-out of a good router, we can convert this good OSPF router into a very malicious OSPF router [9, 7, 1] without replacing the router kernel software (such as Cisco's IOS). Finally, a sophisticated intruder can first attack the network infrastructure, and then use the compromised/controlled network services to attack some other victim hosts. Furthermore, immediately after the latter attack, the intruder can "*restore*" the network infrastructure services back to normal. Under such orchestrated attacks, existing networking utilities like *router filtering* or *traceroute* will not be able to effectively stop the attacks or identify the attack source(s).

Currently, network based intrusions are not handled systematically, and usually human system administrators must be heavily involved. Typically, a system administrator, after hearing some complains over the phone or emails, needs to spend a significant amount of time to check various log files. He then might use existing utilities like *traceroute* or *tcpdump*. If the suspected attack source is in another domain, he most likely will make a few more phone calls to get some help from the administrators in another organization. Even after all these efforts, it is still not always possible that the true attacker will be identified.

In order to deal with network-based intrusions systematically, three logical system components need to be integrated and they must collaborate:

Intrusion Detection System (IDS): The main objective of an IDS is to **decide**, maybe with a probability attribute, whether an observed network packet (or a sequence of observed network packets) forms an **attack instance** or not.

Please note that different IDS modules may reside on different protocol layers. For instance, we may have one IDS in network layer to detect

Ping o' Death attacks, while another IDS may be in the application layer to identify attacks that can only be detected efficiently in the application layer. A sophiscated IDS should be able to correlate intrusion information in different layers.

Attack Source Identification System (ASIS): Under our framework, the IDS normally does not know the attack source(s) even after it detects the attack packets. The key problem is that the source IP address in general can not be trusted. The objective of an ASIS is to utilize the information provided by IDS modules from different layers and to identify where the attacks were coming from. One unique feature about our approach is that *our ASIS module tries to force the attacker to reveal some new location information about the attack sources themselves on every single attack instance being launched.* In other words, if the attacker is forced to launch many attack instances (*i.e.,* persistent attacks), the ASIS module can identify the source(s) very quickly.

Intrusion Damage Control System (IDCS): The objective of an IDCS is to control and repair the damage caused by the attacks detected and identified by the IDS. The IDCS module is very important in handling with hit-and-run attacks. If the IDCS can repair the damage in real-time, then the attacker needs to either re-launch a similar attack or give up on attacking. If the attacker needs to frequently re-launch attacks from the same set of attacking points, we have forced the attacker to perform *persistent attacks.*

In this paper, we will only present the design of an *Attack Source Identification System.* The design and implementation of our IDS and IDCS modules are described in [1, 7].

3 Dynamic Security Associations with IPSEC/ISAKMP

The key concept in IPSEC[4] is the **security association (SA)** relation between two network entities. The basic service options on a particular SA include authentication and encryption, while ISAKMP[5] is designed for SA establishment, negotiation, and tear-down. The network administrators or designers need to specify the policy and decide *where to set up a particular SA*. An important observation, from ASIS point of view is that it depends on where the SAs have been established to decide the amount of source identification information being provided by examining the attack packet's IPSEC header. For example, if a packet's IP source address is `152.1.75.162` and this packet has been authenticated with IPSEC/AH (transport mode) from `152.1.75.162`, then we are pretty sure that the source IP address is trusted.

On the other hand, if the same packet is authenticated with IPSEC/AH (tunnel mode) from `152.1.75.129`, then we can not be sure that this packet is really from `152.1.75.162`. However, we DO know that this packet must have been forwarded by `152.1.75.129`. If this is an attack, then we need to further investigate this attack from `152.1.75.129` and beyond.

Certainly, one obvious way to ensure trusted IP source addresses is to establish IPSEC security associations (transport mode probably) between any two network entities that might need to communicate with each other. This approach is too expensive and static. If we use IPSEC everywhere, then we need to pay for the IPSEC processing overhead even when there are no attacks. Ideally, we would like to establish SAs in a more flexible way. This leads to the concept of **dynamic security associations**: *in order to efficiently identify the attack sources, the ASIS module will* **dynamically decide** *where and when to establish IPSEC SAs.*

A network entity supporting IPSEC/ISAKMP consists of three conceptual modules as shown in Figure 1: ISAKMPd (ISAKMP daemon), SAdB (Security Association database), and SPD (Security Policy Database). The entries in the Security Policy database are maintained either manually or by a security management agent (*e.g.,* BBN's the PSMS (Policy-based Security Management system) project) [11]. Each SPD entry normally has a "selector" pointing to a "bundle" of SAs defined in the SAdB. When a particular SPD entry's selector is NULL, the ISAKMPd will be waked up to establish a bundle of SAs. The local ISAKMPd will first negotiate with a remote ISAKMPd for security options and secret key exchange. The ISAKMP daemon will then update its local SAdB. At this point, the SA establishment process has been completed. To summarize, a dynamic SA is established by dynamically updating the Security Policy database.

4 Source Identification for Linear Network Topology

The attack source identification module in DECIDUOUS requires *network topology information*, which can be provided by either the network administrators or routing daemons. Furthermore, the router policy information on the routers is useful in optimizing the identification speed.

4.1 Single Attacking Source

Let's first consider a six-node network with a simple linear topology:

$$Rtr_A \bullet\!\!-\!\!\!-\!\!\!-\!\!\bullet Rtr_B \bullet\!\!-\!\!\!-\!\!\!-\!\!\bullet Rtr_C \bullet\!\!-\!\!\!-\!\!\!-\!\!\bullet Rtr_D \bullet\!\!-\!\!\!-\!\!\!-\!\!\bullet Rtr_E \bullet\!\!-\!\!\!-\!\!\!-\!\!\bullet \overbrace{Target_F}^{IDS_{jiNao}} \overbrace{Target_F}^{IDS}.$$

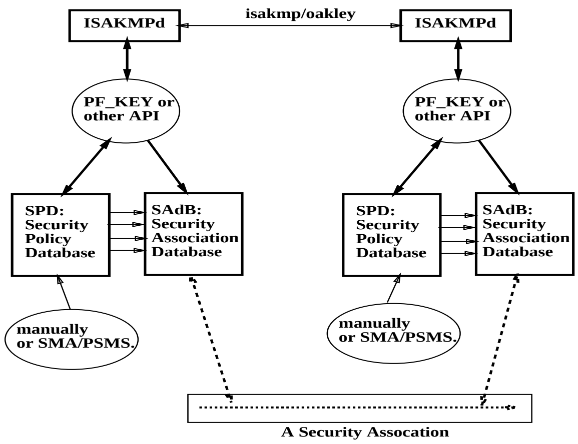

Figure 1: IPSEC Architecture

In this first example, all six nodes are under a single administrative domain. Thus, we assume that IPSEC security associations can be built among these six nodes.

Assumption 1 (Routing Behavior) *With correct network topology information, a good router should always forward the packets through the shortest path. For example, if Rtr_C is a good router, then it should always forward $Target_F$'s packets to Rtr_D. On the other hand, if Rtr_C is compromised, it might forward those packets to Rtr_B. Please note that it depends on router Rtr_B's local configuration and policy to decide how to handle packets coming in from the wrong network interface. In the case of Rtr_C being compromised, Rtr_B could either drop them, log the events, or simply forward them. For simplicity of presentation, we assume that all good routers will* **drop** *such packets.*

An unknown intruder attacks $Target_F$, and IDS_{jiNao} runs on $Target_F$ detects the attack packet(s). The question is where the attack is from. Now, we choose[1] the middle point Rtr_C and an SA (IPSEC/AH tunnel) is established between Rtr_C and $Target_F$:

$$Rtr_A \bullet\!\!-\!\!-\!\!-\!\!\bullet Rtr_B \bullet\!\!-\!\!-\!\!-\!\!\bullet \underbrace{Rtr_C \bullet\!\!-\!\!-\!\!-\!\!\bullet Rtr_D \bullet\!\!-\!\!-\!\!-\!\!\bullet Rtr_E \bullet\!\!-\!\!-\!\!-\!\!\bullet \overbrace{Target_F}^{IDS_{jiNao}}}_{SA:IPSEC/AH:\overline{CF}}.$$

This implies that Rtr_C (if it is good) will forward and authenticate (with tunnel mode) all the packets destinated to $Target_F$. When the next attack is detected

[1] This choice is attack and network environment dependent. We have developed different strategies in choosing the SAs to establish.

by IDS_{jiNao}, we need to decide whether the attack packet itself has been authenticated by Rtr_C or not. If it is properly authenticated:

$$IP^2_{hdr}(dst: Target_F)+IPSEC/AH(from: Rtr_C)+IP^1_{hdr}(dst: Target_F)+IP_{payload},$$

then we can conclude at least one attack source is from either Rtr_A, Rtr_B, Rtr_C, or network links among these four routers. Please note that, even if the attack packet has been correctly authenticated by Rtr_C, Rtr_C itself can still be the attack source. Rtr_D can not be the attacker as Rtr_C will drop (under Assumption 1) Rtr_D's packets destinated for $Target_F$. Now, we can do a binary search on the attack source(s): We can build a new SA between Rtr_B and $Target_F$:

$$Rtr_A \bullet\!\!-\!\!-\!\!\bullet \underbrace{Rtr_B \bullet\!\!-\!\!-\!\!\bullet \underbrace{Rtr_C \bullet\!\!-\!\!-\!\!\bullet Rtr_D \bullet\!\!-\!\!-\!\!\bullet Rtr_E \bullet\!\!-\!\!-\!\!\bullet \overbrace{Target_F}^{IDS_{jiNao}}}_{SA:IPSEC/AH:\overline{CF}}}_{SA:IPSEC/AH:\overline{BF}}.$$

Now, if IDS_{jiNao} detects another attack packet being authenticated only by Rtr_C, then we can conclude that at least one attack source is on either Rtr_B, Rtr_C, or the links among these three routers. At this point, the DECIDUOUS attack source identification process terminates as we can not further reduce the set of suspects by building more SAs.

Theorem 1 (Termination Condition for Linear Topology) *Under Assumption 1, in linear network topology with only one attack source, it is always possible to reduce the number of suspect routers down to 2, but it is generally impossible to reduce further by building new SAs.*

Please note that DECIDUOUS will nail down exactly which link has been compromised if none of the routers are compromised. On the other hand, DECIDUOUS will NOT nail down exactly which router or link has been compromised if some routers or security gateways themselves are compromised. In the latter case, it will identify the attack source as an small area containing a few routers (2 routers in linear network topology) and links in general.

4.2 Multiple Attacking Sources

DECIDUOUS is capable of handling multiple attacking points (even if they coordinate) at the same time. We will illustrate this capability through the following example.

Let's still consider the six-node network with a simple linear topology:

$$Rtr_A \bullet\!\!-\!\!-\!\!\bullet Rtr_B \bullet\!\!-\!\!-\!\!\bullet Rtr_C \bullet\!\!-\!\!-\!\!\bullet Rtr_D \bullet\!\!-\!\!-\!\!\bullet Rtr_E \bullet\!\!-\!\!-\!\!\bullet \overbrace{Target_F}^{IDS_{jiNao}}$$

An unknown intruder compromised both Rtr_A and Rtr_E to attack $Target_F$, and IDS_{jiNao} runs on $Target_F$ detects the attack packet(s). As before, after IDS_{jiNao} detects the first attack, we choose the middle point Rtr_C and an SA (IPSEC/AH tunnel) is established between Rtr_C and $Target_F$:

$$Rtr_A \bullet\!\!-\!\!\!-\!\!\bullet Rtr_B \bullet\!\!-\!\!\!-\!\!\bullet \underbrace{Rtr_C \bullet\!\!-\!\!\!-\!\!\bullet Rtr_D \bullet\!\!-\!\!\!-\!\!\bullet Rtr_E \bullet\!\!-\!\!\!-\!\!\bullet \overbrace{Target_F}^{IDS_{jiNao}}}_{SA:IPSEC/AH:\overline{CF}}.$$

Assume that the intruder uses Rtr_A to launch the next attack to $Target_F$. When this attack is detected by IDS_{jiNao}, the attack packet been authenticated by Rtr_C:

$IP^2_{hdr}(dst: Target_F)+IPSEC/AH(from: Rtr_C)+IP^1_{hdr}(dst: Target_F)+IP_{payload}.$

At this point, we can conclude at least one attack source is from either Rtr_A, Rtr_B, Rtr_C, or network links among these three nodes. We call this region, which contains these three routers and two network links, an *Attacker Zone (A-Zone)*. We can build a new SA between Rtr_A and $Target_F$:

$$\underbrace{Rtr_A \bullet\!\!-\!\!\!-\!\!\bullet Rtr_B \bullet\!\!-\!\!\!-\!\!\bullet \underbrace{Rtr_C \bullet\!\!-\!\!\!-\!\!\bullet Rtr_D \bullet\!\!-\!\!\!-\!\!\bullet Rtr_E \bullet\!\!-\!\!\!-\!\!\bullet \overbrace{Target_F}^{IDS_{jiNao}}}_{SA:IPSEC/AH:\overline{CF}}}_{SA:IPSEC/AH:\overline{AF}}.$$

If the next attack is launched from Rtr_E, the attack packet will not be authenticated by either Rtr_A or Rtr_C. A new attaker zone is formed: Rtr_C, Rtr_D, Rtr_E, $Target_F$, and the links among them. So, at this point, we have identified two zones that possibly contain attacking points. For each A-Zone, we need to have exactly two SAs active in order to identify the attack source within the zone.

5 General Network Topology Transformation

For handling general network topology, we present a transformation algorithm to convert any network topoloy into a linear topology. Then, we can apply the identification scheme in the previous subsection to identify the attack sources.

Definition 1 (Shortest Distance, $SD(v_x, v_y)$) *Let $G = (V, E)$ be an undirected graph and v_x and v_y are two distinct vertices. The shortest distance between v_x and v_y, $SD(v_x, v_y)$, is the number of hops on the shortest path between v_x and v_y.*

Definition 2 (A Protection Cut, $PrtCUT^d_{target}$) *Let $G = (V, E)$ be an undirected graph and v_{target} be the node we would like to protect. A d-distance protection cut for v_{target},*

$$PrtCUT^d_{target} = \{v_x | (v_x \in V) \wedge (SD(v_x, v_{target}) = d)\}.$$

In other words, $ProtCUT^d_{target}$ consists of all the nodes that have a shortest path of exactly d hops to the v_{target}.

With the definition of $ProtCUT$, it is possible to transform a general topology into a linear topology. Below is a simple algorithm to accomplish this goal:

1. Let $G = (V, E)$ be an undirected graph, where V is the set of vertices and E is the set of edges between vertices in G. G is represented by a data structure of adjacency lists, in which each vertex keep a list of its neighbors. Every node also keeps a boolean attribute flag, $CutFlag$, which indicates whether the node has already been included in some $ProtCUT$ or not. Initially, the $cutFlag$ is set to $FALSE$ for all the vertices.

2. v_{target} is the "target," and initially $ProtCUT^0_{target} = \{v_{target}\}$, and v_{target}. $CutFlag$ is set to $TRUE$.

3. For the every vertex in $ProtCUT^d_{target}$, we check its adjacency list. If the $CutFlag$ of an adjacent vertex is $FALSE$, this vertex is included in $ProtCUT^{(d+1)}_{target}$ and its $CutFlag$ is also set to $TRUE$. In this step, we ignore any adjacent vertex with $CutFlag$ already being $TRUE$.

4. This algorithm terminates when $CutFlag = TRUE$ for all the vertices in G.

6 PHIL and Application Layer IDS

DECIDUOUS utilizes the IPSEC header information to locate the attack sources. In the previous isolation example, we inherently made **an important assumption**: *the IDS module, when detecting an attack, can access the IP authenticated header of the packet(s) that carried this attack.* If the attack can be detected in the network layer, then this assumption may be valid as the IDS can be integrated into the network protocol stack. On the other hand, if the attack can only be detected in the application layer, then the IDS is normally unable to identify the original IPSEC AH-tunnel header. For example, an attacker might flood a large number of SNMPv3 messages to deny the services provided by an

SNMPv3 engine. These UDP attack messages will be dropped and detected by the SNMPv3's authentication module in the application layer. However, the IPSEC header information has already been lost at this point.

To deal with this problem, a new socket interface for IPSEC header identification has been developed in DECIDUOUS. In applications like SNMP daemon, instead of the normal system call `recvfrom` to get data from a UDP socket, the application developer will invoke a new system call `phil_recvfrom`. The `phil_recvfrom` interface introduced a new parameter, *Packet Header Identification List (PHIL)*. If the data from the new interface is an attack, then the application program can use another new system call with the PHIL identifier, `phil_retrieve(PHIL)`, to retrieve all the IP-layer authentication information[2] related to this particular attack packet from the kernel (Figure 2). Alternately, this application can pass the PHIL identifier to another process for further attack handling. Note that in order to isolate application level attacks aimed at an end host, the end host must be both IPSEC capable and have the IDS module running locally.

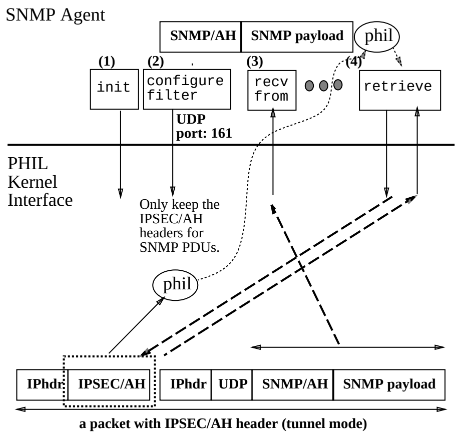

Figure 2: The PHIL Kernel Interface

7 DECIDUOUS Daemon (DECId)

Figure 3 shows the architectural diagram of a target host protected by the DECIDUOUS daemon (DECId) process. The DECId talks to one or more IDS

[2]Please note that, especially with the tunnel mode, an incoming IP packet could have more than one authentication header.

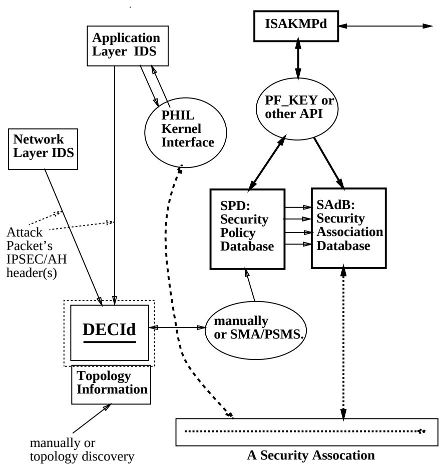

Figure 3: The DECIDUOUS Architecture

components to collect intrusion information. In particular, the IDS must provide authentication header information for those attack packets. For a network layer IDS (*e.g.,* the JiNao network infrastructure intrusion detection system [1]), usually the information is immediately available. On the other hand, for an application layer IDS, we need to utilize the PHIL (Packet Header Identification List) API (described in the previous section) to obtain IPSEC header information.

After obtaining authentication information from the IDS modules, the DECId will decide whether the attack is in one of the already-established attacker zones. If not, a new zone will be created for tracking a new attacking point. In either case, the DECId will decide whether a new SA needs to be established or an existing SA needs to be torn down. The DECId will update the security policy database either directly or through a security management system. The interface between DECId and SPD is currently implementation dependent. At this point, different venders will provide different configuration API for access to the SPD. Therefore, in order to port DECIDUOUS to another platform, we need to re-implement the DECId/SPD interface.

8 The Deciduous Prototype

We have implememnted a very preliminary DECIDUOUS prototype in C and JAVA on FreeBSD without ISAKMP. The dynamic SA mechanism and our IPSEC implementation is built on top of the `ipfw` and *divert socket* mechanisms provided by FreeBSD. Because of the divert socket, our current implementation is completely in the user space. For a simple linear topology with 5 nodes, the DECIDUOUS prototype process will terminate after the $Target_F$ detects 3 or 4 attack packets.

We have also implemented a preliminary prototype to support the *Packet Header Identification List (PHIL)* interface for UDP packets. When an authenticated UDP packet reaches the destination, it will be intercepted by the divert socket to our user-level IPSEC module. After the authentication header is verified, the IP address of the authenticating party is appended at the end of the UDP payload. Although our PHIL prottype is very preliminary, we have successfully identified the IPSEC/AH-Tunnel header information for the attack packets in UDP applications.

9 Related Works

The DECIDUOUS framework proposed here can potentially utilize the results being produced by the BBN's Policy-Based Dynamic Security Management project [11]. The PSMS project offers three novel security management components: *security specification language*, *security management system*, and *security management tools*. These three components will enable us to have a very powerful and standard way to establish dynamic SAs.

DECIDUOUS is targeting the same goal as Boeing's IDIP (Intruder Detection and Isolation Protocol) [8] and our own earlier SSGP (Sleepy Security Gateway Protocol) [10]. The main difference is that, within the same autonomous system, both IDIP and SSGP define a new protocol, while DECIDUOUS uses the existing IPSEC standards for locating the attack sources. IDIP uses IPSEC only for IDIP traffic, while SSGP uses hop-by-hop authentication. On the other hand, DECIDUOUS only depends on the availability of IPSEC/ISAKMP. This implies that we only need to run a DECIDUOUS daemon process on the victim's machine.

In the fault tolerant community, many works have been done in identifying the faulty components. Many of the existing fault identification strategies can be used in the DECId (DECIDUOUS daemon). However, one significant difference between DECIDUOUS and some other existing works is that the former may not be able to nail down the exact faulty component(s). In such cases, DECIDUOUS merely tries to identify a small set of suspected components. Many existing fault detection schemes (such as the PMC model [6]) rely on the assumption of perfect detection. *I.e.,* all neighbors of a particular component can always determine whether the component is faulty or not. This approach may be suitable for malicious attackers as they can pretend to be normal at any time. Furthermore, the PMC-like approach usually requires that every network entity will have a perfect detection component, which can be harder to deploy.

10 Conclusion

In this paper, we proposed a new security management framework, DECIDUOUS, for identifying intrusion sources. DECIDUOUS is designed to work with intrusion detection systems (IDS) and intrusion damage control systems (IDCS). The integration of DECIDUOUS, IDS, and IDCS is a powerful architecture for dealing with various network-based intrusions.

DECIDUOUS itself is simple, practical, and built on top of the IPSEC infrastructure. While tracking facilities like `traceroute` might fail in certain cases, DECIDUOUS will deliver reliable location information about the attack sources. DECIDUOUS will survive and function under various serious insider attacks. For example, it can function even some of the network infrastructure entities (*e.g.,* routers or IPSEC security gateways) are compromised. Furthermore, it can track down multiple attack sources simultaneously even when those different attack sources coordinate to attack the same victim.

We have implemented a preliminary DECIDUOUS prototype which runs on our FreeBSD routing testbed. Our preliminary experiments show that the DECIDUOUS implementation can be simple, modular, reasonably efficient, and in the user space (*i.e.,* no kernel modifications are required for FreeBSD). We are in the process of developing replacing this protocol with IETF's ISAKMP. Our final goal is to demonstate the interoperability between DECIDUOUS and the standardized IPSEC/ISAKMP infrastructure.

Acknowledgments

We would like to thank Dan Schnackenberg (Boeing) and John Zao (BBN) for providing us information about IDIP and PSMS respectively.

References

[1] F. Jou, F. Gong, C. Sargor, S. F. Wu, and R. Cleaveland. Architecture Design of a Scalable Intrusion Detection System for the Emerging Network Infrastructure. Technical Report E296, Adavnced Network Research, MCNC, April 1997.

[2] S. Kent and R. Atkinson. IP Authentication Header. Internet Draft, IETF, draft-ietf-ipsec-auth-header-05.txt, March 1998. Network Working Group.

[3] S. Kent and R. Atkinson. IP Encapsulating Security Payload. Internet Draft, IETF, draft-ietf-ipsec-esp-v2-04.txt, March 1998. Network Working Group.

[4] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. Internet Draft, IETF, draft-ietf-ipsec-arch-sec-04.txt, March 1998. Network Working Group.

[5] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol. Internet Draft, IETF, draft-ietf-ipsec-isakmp-09.txt, March 1998. Network Working Group.

[6] F.P. Preparata, G. Metze, and R.T. Chein. On connection assignment problem of diagnosable system. *IEEE Transactions on Electronic Computers*, EC-16:848–854, 1967.

[7] D. Qu, R. Narayan, F. Wang, B. Vetter, S.F. Wu, F. Jou, F. Gong, and C. Sargor. Statistical Anomaly Detection for Link State Routing Protocols. In *IEEE International Conference on Network Protocols (ICNP)*, page to appear, October 1998.

[8] D. Schnackenberg. Dynamic Cooperating Boundary Controllers. http:// www.darpa.mil/ito/ Summaries97/E295_0.html, March 1998. Boeing Defense and Space Group.

[9] B. Vetter, F. Wang, and S.F. Wu. An Experimental Study of Insider Attacks for the OSPF Routing Protocol. In *IEEE International Conference on Network Protocols (ICNP)*, pages 293–300, October 1997.

[10] Shyhtsun F. Wu. Sleepy Network-Layer Authentication Service for IPSEC. In G. Martella E. Bertino, H. Kurth and E. Montolivo, editors, *4th European Symposium on Research in Computer Security - ESORICS 96*, LNCS-1146, pages 146–159, Rome, Italy, September 1996.

[11] J. Zao. Policy-Based Dynamic Security Management. http:// www.darpa.mil/ito/ Summaries97/F318_0.html, March 1998. BBN.