Analyzing Network-Aware Active Wardens in IPv6

Grzegorz Lewandowski, Norka B. Lucena, and Steve J. Chapin

Systems Assurance Institute
Syracuse University
Syracuse, NY 13244, USA
grlewand@syr.edu, {norka,chapin}@ecs.syr.edu

Abstract. A crucial security practice is the elimination of network covert channels. Recent research in IPv6 discovered that there exist, at least, 22 different covert channels, suggesting the use of advanced active wardens as an appropriate countermeasure. The described covert channels are particularly harmful not only because of their potential to facilitate deployment of other attacks but also because of the increasing adoption of the protocol without a parallel deployment of corrective technology. We present a pioneer implementation of *network-aware* active wardens that eliminates the covert channels exploiting the *Routing Header* and the `hop limit` field as well as the well-known *Short TTL* Attack. Network-aware active wardens take advantage of network-topology information to detect and defeat covert protocol behavior. We show, by analyzing their performance over a controlled network environment, that the wardens eliminate a significant percentage of the covert channels and exploits with minimal impact over the end-to-end communications (approximately 3% increase in the packet roundtrip time).

Keywords: covert channels, evasion attacks, active wardens, stateless, stateful, network-aware, traffic analysis, traffic normalizers, active mappers.

1 Introduction

Although as of today publicly-accessible Internet addresses are primarily IPv4, the adoption of the Internet Protocol version 6 (IPv6)[1] is becoming imminent. For example, news from the IPv6 Task Force [1] report significant progress in both deployment and policy regarding networks using IPv6 technology in various continents [2,3]. IPv6 summits and other events present applications and services that will drive commercial implementations of IPv6 [4,5,6,7]. The U.S. government established that all federal agencies must deploy IPv6 by June 2008 [8], without disregarding the challenge of the Department of Defense (DoD) of monitoring operational IPv6 networks for unauthorized IPv6 traffic [9]. That global embracement of IPv6 calls for closer examination of its security risks, especially of those which are not so obvious nor possibly overcome by IPv4 security technologies.

Lucena, et al. [10] presents a comprehensive examination of covert channels in IPv6. It analyses 22 different network storage channels at the IP level, classifying them by

[1] IPv6 is also referred as the Next Generation Internet Protocol or IPng.

J. Camenisch et al. (Eds.): IH 2006, LNCS 4437, pp. 58–77, 2007.

type of header. To defeat the identified channels, it defines three types of active wardens: stateless, stateful, and network-aware, which differ in complexity and ability to block some types of covert channels. A *stateless* active warden normalizes IPv6 traffic according to a protocol specification, without remembering anything about the packet that have already passed by. A *stateful* active warden records and recalls previous packet behaviors to discover a conceivably larger spectrum of hidden channels. A *network-aware* active warden is a stateful active warden with knowledge of network topology. The description of those active wardens is only conceptual. Until now, there has not been discussion of how one can implement network-aware wardens.

The IPv6 covert channels appear to be subtle types of aggression, when comparing to well-known buffer overflow attacks, for example. However, they are as harmful, especially under the presence of sophisticated adversaries[2]. It is feasible for an attacker to secretly transmit information into or out of a compromised machine residing on a secure network through the use of covert channels. For example, hacker Alice, after installing a key stroke logger and obtaining users' credentials, retrieves stolen information employing a covert channel. Alternatively, after installing a backdoor program, cracker Bob sends commands via a covert channel. Understanding that the use of IPv6 covert channels might be particularly damaging when an attacker utilizes them with the purpose of maintaining long-term control over a compromised machine, we present and evaluate an implementation of *network-aware* active wardens.

In this study, we consider two of the channels described in [10] and a well-known aggression in IPv4 [11,12,13,14]: the Routing Header covert channel, the Hop Limit channel, and the Short TTL Attack, respectively. The first two covert channels exemplify secret communication mechanisms of high and low bandwidth, respectively. The last one defines a relevant crossover point between the two versions of the IP protocol. The *Routing Header covert channel* takes advantage of the IPv6 source routing functionality to transfer data in a way that violates system security policies. The *Hop Limit channel* achieves a similar goal by manipulating the `hop limit` field of the IPv6 header. The *Short TTL* Attack allows an attacker to mask malicious communications or another attack from a Network Intrusion Detection System (NIDS). For a more detailed description of these attacks, please see Appendix A.

To prove that network-aware active wardens constitute an appropriate countermeasure against the selected IPv6 covert channels, we measure their effectiveness within a controlled network environment, by estimating a percentage of extermination per case and by measuring the increase over the roundtrip time of end-to-end traffic flows. We aim to defeat the selected channels, while causing roundtrip times increments no higher than 5%.

The remainder of this document is organized as follows. Section 2 compiles previous work on network covert channels in both IPv4 and IPv6, summarizing existing countermeasures. Section 3 specifies the design and implementation of the network-aware active wardens, presents results of performance tests set up on a controlled network, and discusses the implication of the obtained outcomes. Finally, Section 4 draws conclusions and suggests future directions of research related with the topic.

[2] The more secure nature of IPv6 in relation to IPv4 demands even more knowledgeable foes.

2 Related Work

Research in network covert channels [15] comprises the study of both network- and transport-layer protocols, such as IP, TCP, ICMP, and application-layer protocols, such as HTTP. It is not surprising to observe that the majority of the literature relates to network storage channels [10,16,17,18,19,20,21,22,23,24] rather than network timing channels [15,25,26,27,28]. Timing channels are presumably less attractive because of their synchronization issues and their low bandwidth in comparison to storage channels. However, it is somewhat peculiar that given the increasing use of IPv6, most of the research still concerns IPv4.

The most effective defensive mechanisms against network storage channels for IPv4 are protocol scrubbers [13], traffic normalizers [11], and active wardens [29,30,31,32]. Protocol scrubbers and traffic normalizers focus on eliminating ambiguities found in the traffic stream, carefully crafted with the purpose of evading network intrusion detection systems. *Ambiguous* network packets are those that could have different interpretations at endpoints depending on the implementation of the protocol stack. Covert channels are certainly a form of ambiguous traffic. Handley and Paxson [11] describes IP, UDP, TCP, and ICMP normalizations based on protocol semantics, highlighting the importance of preserving the end-to-end protocol semantics. In the same order of ideas, active wardens, as presented by Fisk et al. [32], are network services resembling a firewall that modify all traffic under the assumption that it is carrying steganographic content. Active wardens defeat steganography by making semantics-preserving alterations to packet headers (e.g. zeroing the padding bits in a TCP packet). These techniques, although effective for most IPv4 covert channels, do not record any state or gather network topology information.

Among the approaches and technologies that gather topology information with the purpose of detecting undesired traffic on the network are active mappers [14], NetFlow [33], network monitors such as Ntop [34], and certain implementations of the Simple Network Management Protocol (SNMP) [35], such as IBM Tivoli NetView [36], HP OpenView Network Node Manager [37], Marconi ForeView, and Sun Solstice Site Manager [38]. Shankar and Paxson [14] proposes an alternative approach to traffic normalizers [11] called *active mappers* that minimizes the performance penalties caused by packet reassembling. Active mapping involves building profiles of the network topology and the TCP/IP policies of hosts to help NIDSs disambiguate the interpretation of network traffic. The mappers gather topology information *actively*, sending specially crafted probing messages to each host on the network. Ntop, from `www.ntop.org`, is a traffic measurement and monitoring system with an embedded NIDS that gathers certain information about network topology and host relationships [34]. Ntop learns about topology based on network flows, so it actually depends on the existence of those flows: there is no knowledge without flow. Therefore, the view of the topology drawn by Ntop might be incomplete in certain situations (for example, when flows traveling to adjacent subnets do not pass by the system). NetFlow version 9, supporting IPv6, provides several services being the most important flow recording. It also provides information about traffic routing. The commercial SNMP products provide an understanding of the physical network topology through different information gathering mechanisms.

Network-aware active wardens are not exactly traffic normalizers nor active mappers, but an innovative technology that comprises some of the best features of both. Active mapping is meant to work in conjunction with NIDSs, assisting them in resolving network ambiguity. In consequence, they do not eliminate the ambiguities. They aid NIDSs to alert network administrators of unwanted protocol behavior with more precision (than without the mappers). Active wardens, with knowledge of the network topology, defeat covert channels based on network ambiguities without significant overhead, actually alleviating the workload of a NIDS positioned after the warden.

3 Network-Aware Active Wardens

As originally defined in Lucena, et al. [10], *network-aware* active wardens are the most sophisticated type of wardens. A network-aware active warden can not only reinforce protocol syntax and semantics preservation (both passively or actively), but also perform address verification using topology information about the surrounding networks. The following subsections explain the design of our implementation of a network-aware active warden, list assumptions made, and analyze performance measurements. To simplify the discussion, from this point on, a network-aware active warden will be referred simply as warden, active warden, or just Wendy.

3.1 Overview and Rationale

Objectives. The main purpose of an active warden is to the break covert channel communication or to remove the cover traffic masking an attack from a NIDS, as in the Short TTL scenario. In the former case, the goal is to disable the covert channel without affecting the legitimate usage of the exploited header. That is, only packets carrying covert data in their headers should be modified, preserving the protocol semantics[3]. In the latter case, the purpose is to remove the “mask” so the ulterior attack becomes visible to a NIDS. The warden itself does not perform the detection, but eliminates the evasion.

Assumption 1. *The warden always attempts not to break the overt communication taking place through a suspicious flow.*

A secondary, but no less important, goal of a network-aware active warden is to take advantage of network topology information to properly defeat the covert channels. As detailed in Section 2, there exist multiple ways for a warden to gather such information: scanning network administrators’ topology tables, sending probing messages to individual hosts on the network [14], and using network-monitoring tools [33,34] or particular applications implementing SNMP [36,37,38].

Assumption 2. *The warden already possesses the topology information of the guarded network, previously acquired through complementary technologies.*

[3] When preserving header functionality is not a concern, the covert channels can be defeated by simply disabling specific header support on a given network.

Location. The Internet comprises a collection of autonomous systems. An *autonomous system*[4] (AS) is a subset of routers that make up an internetwork and exchange information through a common routing protocol [39]. An *autonomous system border router* (ASBR) exchanges information between two ASs, maintaining separate topological databases for each. The location of the warden within the network topology, formed by those ASs, significantly affects her ability to detect covert communication.

There are two prevailing locations where to place the warden, depending on the network architecture Wendy wants to protect. A warden who sits on or near an ASBR (see Figure 1) is a *border* warden. A warden who sits on or near an internal router is a *link* warden. Border wardens aim to block covert communication channels established between an interior host and a point outside the local autonomous system (regardless of which participant originates the inter-AS channel). Link wardens disable intra-AS channels.

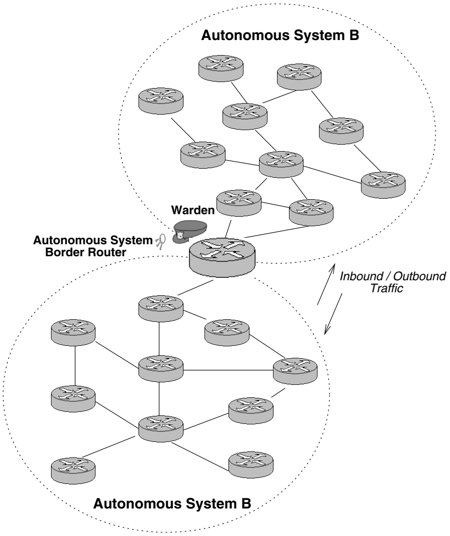


Fig. 1. Border Warden

Consequently, the location within the network topology determines the type of information the warden has available as well as the actions she can take. If Wendy is a link-level warden, she has information about all the nodes of the subnet. However, all that information is useful only to local or internal traffic verification. On the other hand, if Wendy is a border warden, she can observe inbound and outbound AS traffic, which is presumably more susceptible to attack.

Assumption 3. *Wendy is a border warden.*

[4] In the Internet protocol context, autonomous systems are called *routing domains*.

3.2 Attack Model

The implemented active warden relies on several assumptions about the adversary's capabilities. Those assumptions agree with the ones presented in [10], and generally are not stronger.

The opponents behave *actively* [30]. They have both the resources and skills to alter packets in transit, by either modifying values of protocol fields or by injecting an entire field, a header, or a crafted packet.

Assumption 4. *Adversaries can modify network packets traveling between nodes.*

As it is for the wardens, the location within the protected AS is relevant for the attackers.

Assumption 5. *Adversary Alice is located within the protected AS. Adversary Bob is located outside of Alice's network.*

Following Shannon's maxim "the enemy knows the system" [40], it is possible that Alice knows about both the existence and the location of the warden. In addition, if Alice learns about Wendy, she can also learn about the topology of the network under her attack.

Assumption 6. *Adversaries may or may not have knowledge of the existence of the warden and her location.*

Adversaries who do not know about the wardens are said to be *blind*.

3.3 Covert Channel Defense

This subsection describes the countermeasures taken by the implemented warden to eliminate the *Routing Header* covert channel, the *Hop Limit* channel, and the *Short TTL* Attack. Relevant details about the operation of these channels appear in Appendix A.

Eliminating the Routing Header Covert Channels. To defeat the covert channels in the Routing Header, an active warden has to perform several checks on the protocol semantics and behavior. We identify for different ones. The first check is somewhat simpler than the remainder four being based exclusively on the IPv6 specifications [41,42] and the address space allocation document [43].

- **Hop Address Check.** This check relies on the fact that only *aggregatable global unicast* addresses are meaningful within a packet's Routing Header [41]. *Multicast* addresses are explicitly forbidden, plus *local* addresses (both *unique-local* and *link-local*) are not supposed to cross site the boundary of the protected AS. Hence, the border warden should not observe any of the last two address types.

  In addition, "a routing header is not examined or processed until it reaches the node identified in the `destination address` field" [41], giving Alice and Bob enough opportunities through intermediate hosts of interpreting the covert message. Our implementation of active warden performs a more aggressive check, verifying at the border router that all addresses contained in the Routing Header are, in fact, *aggregatable global unicast* addresses.

Aggregatable global unicast addresses begin with the bit pattern 001. Assuming that the covert messages follow a uniform random distribution, there is one in eight chance of beginning with the bits 001. Therefore, a *blind* attacker will have, at least, 87.5% chance of being caught when injecting messages crafted as addresses in a Routing Header (see Appendix B for details).

For the following checks, it is important to distinguish between incoming and outgoing traffic. *Incoming* traffic refers to packets whose source address is outside the AS and whose destination address belongs to the AS. *Outgoing* traffic, contrarily, has a source address within the AS and a destination address outside the AS. The direction in which the traffic flows determines what types of checks are needed. Table 1 summarizes the corresponding required checks. Because the analysis of both directions is symmetric, we discuss only the case of outgoing traffic.

Table 1. Topology Checks Required Depending on the *Segments Left* Field Value (*Visited* or *Not Visited*) for Both Incoming and Outgoing Traffic

	Router Address Check	IP Range Check
Addresses Marked as *Visited*	*outgoing*	*incoming*
Addresses *Not Visited*	*incoming*	*outgoing*

- **Router Address Check.** For outgoing packets whose Routing Header addresses are marked as *visited*, Wendy verifies whether or not they are valid inside the protected AS. In addition, because only routers perform packet forwarding, those addresses must correspond to routers. Therefore, if already-visited addresses in the Routing Header of an outgoing packet do not belong to addresses of internal AS routers, she concludes that a covert channel exists.
- **IP Range Check.** Alternatively, if the addresses in the Routing Header of an outgoing packet are marked as *not visited*, the warden inspects if they fall within the range of addresses assigned to the AS. If so, it means that the packet will eventually come back to the system. Appropriately, Wendy will also suspect that a covert communication is taking place. This check is a more elaborate version of the address-based ingress/egress filtering performed by some firewalls.
- **Tandem Check.** It is possible to circumvent the last two checks by crafting an outgoing packet whose Routing Header addresses are marked as *not visited* and do not match the IP range of the AS. The converse deception also holds for incoming packets. However, if there are active wardens positioned near both the origin and the destination of the covert communication, an attacker cannot easily generate covert packets without being detected. For example, an attacker Alice wants to transmit a covert message from A to B in the scenario of Figure 2. To be able to deceive the active warden sitting on A's border router, she will have to mark all the fake addresses as *not visited* while making them different from any address within A's IP address range. However, when a packet formatted in such manner arrives to

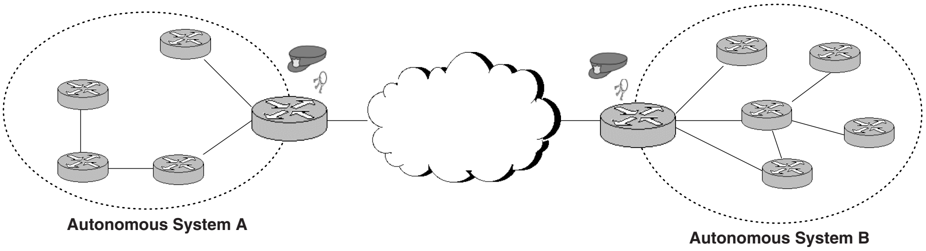


Fig. 2. Example of Tandem Wardens Performing Topology Checks

B, the active warden residing in B's border router will perform the usual verification. The only way then for the attacker be able to bypass that second warden is to have knowledge about B's router addresses. While not impossible that particular situation requires additional effort from the adversary. That is, even when Alice possesses knowledge of B's topology, she can only conceal messages that mimic actual router addresses within B's routing domain, not just any arbitrary data. The only option left for the adversary then is to manipulate the order of legitimate router addresses in the header to convey a message. That channel however has low bandwidth in comparison to the original channel, specifically, $128/log_2(r)$ times lower, where r stands for the number of router within the AS (refer to Appendix B for bandwidth calculations).

Once Wendy identifies the presence of a covert channel, she proceeds to eliminate it. The trivial way of eliminating any channel is to simply drop the suspicious packet. That action might, in most cases, break the overt communication. As stated in assumption 1, Wendy will always prefer less disruptive methods. A more appropriate solution is to strip the covert message from a packet and allow it to proceed normally. Whether the warden can actually modify the Routing Header or not depends on whether the packet is IPSec protected or not[5]. For the purposes of this study, the IPv6 traffic is not IPSec protected.

Eliminating the Hop Limit Covert Channels. The Hop Limit covert channel makes use of a `hop limit` field in IPv6 packet headers to transmit covert messages. The detection of this channel is troublesome because the value of the hop count can vary naturally as an effect of packets traveling different routes. A trivial attempt to break the channel is for the warden to reset the hop limit value in all packets in transit to an arbitrary value. That however can be potentially damaging as it prevents the `hop limit` field from its intended purpose, to avoid packets traveling indefinitely and

[5] Under IPSec, the modification of a packet header might result in failure of the integrity check, causing the packet to be discarded. It is important to note that if an attacker intercepts and modifies a legitimate packet without having access to the IPSec security context, that packet will be analogously dropped. If the adversary does know the security context and protects the covert message under the IPSec integrity check, the overt communication may not be legitimate third-party traffic and may be discarded anyway.

hence saturating the network in the case of a routing cycle. If a warden chooses to reset the field to a *small* value, it lowers the risk of encountering a cycle, but increases the probability that legitimate packets will expire on their way to the destination without reaching it.

On the other hand, a network-aware active warden applies her knowledge to manipulate the `hop limit` field in a safer manner. For incoming packets, Wendy can infer the minimum hop limit value which is sufficient to prevent the packets from expiring before their intended destination. If the initial hop limit value is enough to reach the destination, the warden resets it to the inferred value. If it is not large enough, the warden takes similar actions to the ones stated in the Short TTL Attack. In both cases, Wendy defeats the channel 100% of the times, when occurring on inbound traffic. For outgoing packets, the warden is not always able to make similar premises about the minimum hop limit value. However, when the covert communication involves two ASs (e.g., Alice resides in AS A and Bob in AS B), each of them protected by a warden as in Figure 2, it is plausible to disable the covert communication. Symmetrically, the traffic seen by one of the wardens as outgoing will, in fact, be incoming from the standpoint of the other warden. To completely, eliminate the channel when happening in outbound traffic, Wendy might reset the hop limit value as done by IPv4 traffic normalizers [11] for the TTL value, but at the risk of incurring the same drawbacks.

Eliminating the Short TTL Attack. The Short TTL Attack utilizes packets with a small hop limit value to mask another attack from being detected by a network intrusion detection system. The active warden is not concerned with detecting the covert attack, but with removing the cover traffic so that an existing network intrusion detection system is able to detect the attack.

Handley and Paxson [11] proposes to prevent the exploit through the use of a traffic normalizer that either drops packets with a short TTL or restores the TTL value to a number that would guarantee packet delivery. The first solution is not actually implemented by the normalizer because of the lack of a topology gathering mechanism. Nevertheless, Shankar and Paxson [14] did carry out the suggested approach with a successful outcome. As discussed in the previous case, resetting the TTL value in IPv4 or the hop limit value in IPv6 without any knowledge of the network topology compromises the interconnected system. Our implementation of Wendy overcomes those difficulties with her network topology knowledge, defeating the Short TTL evasion 100% of the times.

To illustrate the concept, Figure 3 shows an example of how the warden helps defeat a Short TTL attack. An adversary targeting host **X** might conceal the attack by masking the traffic with a hop limit value expiring at router **C**. If the only defense is a NIDS located before **C**, the malicious traffic might circumvent it. However, if active warden Wendy works in combination with a NIDS, she is able to detect that the packets will not reach the final destination **X** and drop them before they pass by the NIDS. In the presented scenario, Wendy should discard all packets addressed to **D** if their hop limit value is smaller than 2.

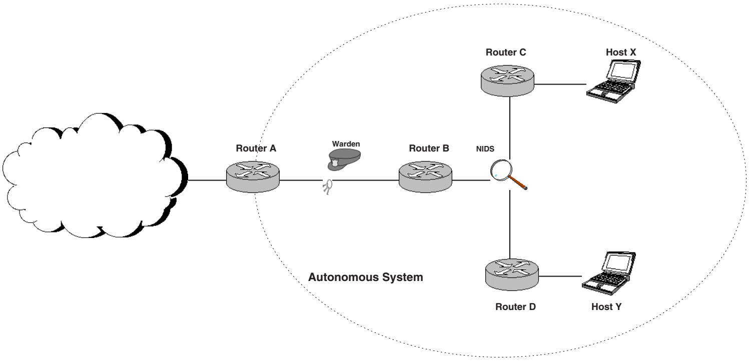


Fig. 3. Active Warden and NIDS Positioning

3.4 Prototype Implementation

We implemented our prototype of a network-aware active warden as a kernel module in a Linux router, running Fedora Core 4, kernel version 2.6.14. The prototype uses the *netfilter hooks* library to intercept and examine network traffic. The same machine also runs a firewall in permissive mode. Because the firewall operates in that mode without enforcing any complex rules, the impact of the active warden on the network performance tends to be more visible. Our Wendy acts as a border warden, as shown in Figure 4.

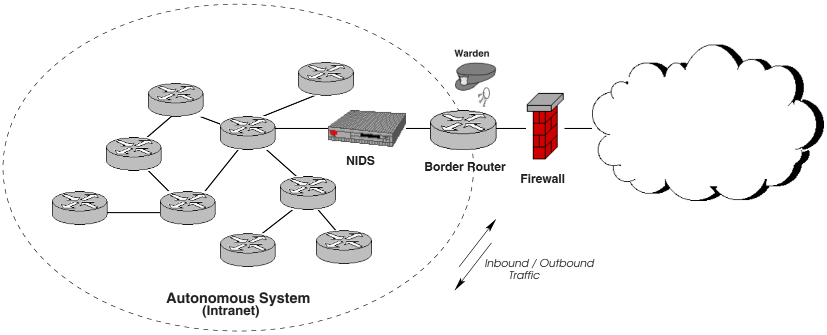


Fig. 4. Location of the Active Warden with respect to the guarded AS. Wendy renders useless the possible covert channels and evasions contained in the traffic that already bypassed the firewall before it is checked by the NIDS.

In addition, the prototype runs within a controlled network environment, which allows Wendy to have a preconfigured knowledge about the network topology. To ensure constant access times, the topology knowledge is stored in a hashtable that maps node addresses to hop distances.

The controlled network environment consists of a router connected to two subnets. One simulates the protected AS (Intranet) through a number of IPv6 addresses, varying from 10 to 1000. The second one mimics the outside world (Internet).

3.5 Results

We evaluated the effectiveness of the implemented warden computing the average roundtrip times for different packet sizes, different lengths of Routing Header, and different Intranet sizes, performing 10 measurements each time.

Figure 5 exhibits average roundtrip times for packets of 64-byte length and of 4096-byte length traveling between end points, with and without the warden siting on the border router. The obtained values for 64-byte packet were $0.3029ms \pm 0.0004ms$ (without warden) and $0.3136ms \pm 0.0002ms$ (with the warden). The difference found between the averages represents a 3.3% increase of the roundtrip time. Similarly, for 4096-byte packets the average times were $1.8939ms \pm 0.0006ms$ (without the warden) and $1.9037ms \pm 0.0003ms$ (with the warden). There was only a 0.5% increase in the average times of the larger packets.

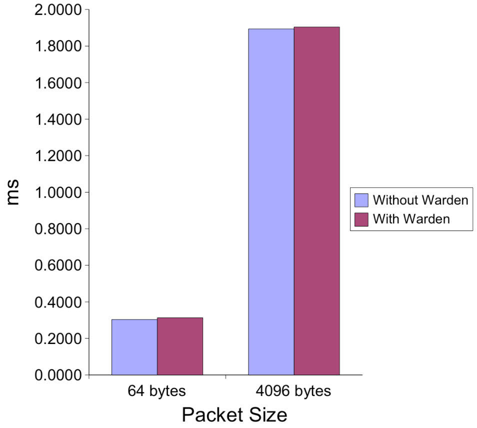


Fig. 5. Average Roundtrip Times of Packets of Sizes 64 and 4096 Bytes

Figure 6 shows average roundtrip times of packets carrying no Routing Header or Routing Header with 1 and 16 addresses. When the packets did not have a Routing Header the average roundtrip times were $0.250ms \pm 0.001ms$ (without the warden) and $0.257ms \pm 0.002ms$ (with the warden), exhibiting a total increase of 2.8%. Analogously, with a 1-hop Routing Header the average roundtrip times varied from $0.268ms \pm 0.002ms$ (without the warden) to $0.277ms \pm 0.002ms$ (with the warden), where the increment is 3.3%. For a 16-hope Routing Header, the achieved values were $0.382ms \pm 0.002ms$ (without the warden) and $0.389ms \pm 0.003ms$ (with the warden), being the case with minimum increase: 1.8%.

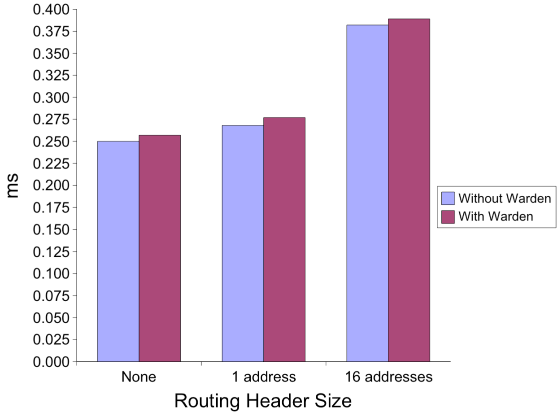


Fig. 6. Average Roundtrip Times of Packets with No Routing Header and Routing Headers containing 1 and 16 addresses

Figure 7 displays the differences in average roundtrip times when the packets traverse networks composed of 10 and 1000 hosts. As observed graphically, there was no difference at all in the average times obtained for the two network sizes. Precisely, the average times recorded were $0.303ms \pm 0.001ms$ (without the warden) and $0.313ms \pm 0.001ms$ (with the warden).

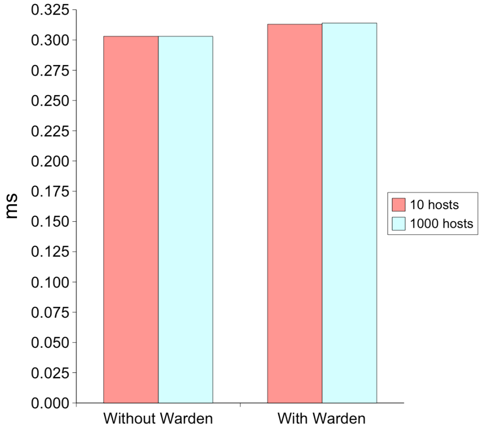


Fig. 7. Average Roundtrip Times of Packets Traversing Networks of 10 and 1000 Hosts

3.6 Discussion

Analyzing the test results presented in Subsection 3.5, we observe the following:

1. The relative delay introduced by our active warden decreases as the packet size grows. This is presumably caused by the fact that the border router works harder when distributing larger packets, while the warden's load of scanning the Routing Header stays the same. Hence, the absolute overhead remains constant, causing the relative overhead to shrink.
2. Both the presence and the size of the Routing Header affects the warden's performance. This is trivially explained by the Routing Header normalizations performed by the warden, which require scanning each of the contained addresses.
3. The size of simulated network topology does not influence the warden's performance. This is not a surprising outcome because the data structures used by the warden to store topology information exhibit constant lookup times.

We initially envisioned to produce a network-aware active warden that completely defeated the selected covert channels, without increasing packet roundtrip times in more than 5%. As detailed in Subsection 3.3, we found that it is virtually impossible to eliminate some of them under our attacker model. However, the percentages of elimination estimated for each case are significant, especially considering that several of them are close to 100% and that, even when the attacker can circumvent our warden, the bandwidth of the secret communication drops dramatically. On the other hand, regarding the overhead caused by the warden in the packet roundtrip time, the results indicate that we reached our goal. All tests showed increases in the average roundtrip times of approximately 3%, being 3.3% the highest.

When comparing our implementation of network-aware active wardens to IPv4 technologies that deal with network ambiguities [11,14], the prototyped warden presents both differences and similarities. Wendy behaves as a traffic normalizer because she also performs active protocol semantics reinforcement. Moreover, she resembles an active mapper when using network topology information to disambiguate traffic. However, our active warden differs in the way she obtains the knowledge about the topology. In addition, the prototype implementation does not compromise significantly the performance of packets traveling end-to-end. That occurs, presumably, for two reasons: a) the use of more precise methods of handling network ambiguities (when comparing to the ones in traditional normalizer), and b) the fact that the warden does not perform packet reassembling.

Finally, considering future directions of research as well as possible improvements in the warden evaluation, we identify the following factors:

- While a controlled network environment was useful for gathering initial results, this environment obviously did not provide large volume of traffic. It is necessary to repeat the tests over a real-world network and compare the results.
- Our warden defeated only two of the 22 covert channels described in [10]. It is critical to extend the warden implementation in such way that can block the rest of the channels.
- Our covert channel countermeasures may be compromised by attacker who knows the system by, for example, taking control of the warden or by launching a denial-of-service attack. It is critical to examine the robustness of the warden in future implementations.

4 Conclusions

In this study we designed and implemented a version of *network-aware* active wardens [10] to defeat the *Routing Header* covert channel, the *Hop Limit* covert channel, and the *Short TTL* attack. The warden not only normalized the protocol semantics, but also utilized network topology information to effectively defeat the covert channels and exploits. It proved to render useless instances of covert communication occurring within a controlled network environment, while causing a penalty in the packet roundtrips of only approximately 3%.

Based on our initial results, we believe that *network-aware* active wardens are a promising technology that represents a step forward in the elimination of new security threats in IPv6 such as recently discovered covert channels. We also hope that our work generate discussion regarding other adequate countermeasures and feasible fixes to the protocol.

References

1. The IPv6 Portal. Retrieved on June 22, 2005 from the World Wide Web: http://www.ist-ipv6.org/ (2005)
2. Press Trust of India: TRAI wants govt to kickstart shift to ipv6 through e-gov. http://www.hindustantimes.com/news/181_1578124,00020020.htm (2005)
3. ChinaView: China, EU to build wide-band network. Retrieved on January 12, 2006 from the World Wide Web: http://news.xinhuanet.com/english/2006-01/12/content_4045153.htm (2006)
4. United States IPv6 Summit. Retrieved on November 05, 2005 from the World Wide Web: www.usipv6.com/ (2005)
5. Global Summit IPv6. Retrieved on May 17, 2005 from the World Wide Web: http://www.ipv6-es.com/05/in/i-intro.php (2005)
6. IPv6 Forum Korea. Retrieved on October 13, 2005 from the World Wide Web: http://www.ipv6.or.kr/ (2005)
7. Luxembourg IPv6 Summit 2005. Retrieved on June 22, 2005 from the World Wide Web: http://wiki.uni.lu/ipv6/Luxembourg+IPv6+Summit+2005.html (2005)
8. Evans, K.S.: Memorandum for the chief information officers, M-05-22. http://www.whitehouse.gov/omb/memoranda/fy2005/m05-22.pdf (2005)
9. United States Government Accountability Office: Internet protocol version 6: Federal agencies need to plan for transition and manage security risks. Technical Report GAO-05-471 (2005) http://www.gao.gov/new.items/d05471.pdf.
10. Lucena, N.B., Lewandowski, G., Chapin, S.J.: Covert channels in IPv6. In: Proceedings of the 5^{th} Workshop on Privacy Enhancing Technologies, Dubrovnik (Cavtat), Croatia (2005)
11. Handley, M., Paxson, V.: Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics. In: Proceedings of the 10^{th} USENIX Security Symposium, Washington, DC, USA, USENIX Association (2001)
12. horizon<jmcdonal@unf.edu>: Defeating sniffers and intrusion detection systems. Phrack Magazine Volume 8, Issue 54 (1998) Retrieved on May 13, 2005 from the World Wide Web: http://www.phrack.org/phrack/54/P54-10.
13. Malan, G.R., Watson, D., Jahanian, F., Howell, P.: Transport and application protocol scrubbing. In: Proceedings of the IEEE INFOCOM 2002 Conference, Tel-Aviv, Israel (2000) 1381–1390

14. Shankar, U., Paxson, V.: Active mapping: Resisting NIDS evasion without altering traffic. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, Washington, DC, USA, IEEE Computer Society (2003) 44–61
15. Cabuk, S., Brodley, C.E., Shields, C.: IP covert timing channels: Design and detection. In: Proceedings of the 11^{th} ACM Conference on Computer and Communications Security, Washington DC, USA, ACM Press (2004) 178–187
16. Handel, T., Sandford, M.: Hiding data in the OSI network model. In Anderson, R., ed.: Information Hiding: Proceedings of the First International Workshop, Cambridge, U.K., Springer (1996) 23–38
17. Abad, C.: IP checksum covert channels and selected hash collision. Retrieved on January 3, 2005 from the World Wide Web: http://gray-world.net/cn/papers/ipccc.pdf (2001)
18. Bauer, M.: New covert channels in HTTP - adding unwitting web browsers to anonymity sets. In Samarati, P., Syverson, P., eds.: Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society, Washington, DC, USA, ACM Press (2003) 72–78 ISBN 1-58113-776-1.
19. daemon9 (route@infonexus.com): Loki2 (the implementation). Phrack Magazine, 51, article 6 (1997) Retrieved on August 27, 2002 from the World Wide Web: http://www.phrack.org/show.php?p=51&a=6.
20. daemon9 (route@infonexus.com), alhambra (alhambra@infornexus.com): Project loki. Phrack Magazine, 49, article 6 (1996) Retrieved on August 27, 2002 from the World Wide Web: http://www.phrack.org/show.php?p=49&a=6.
21. Dunigan, T.: Internet steganography. Technical report, Oak Ridge National Laboratory (Contract No. DE-AC05-96OR22464), Oak Ridge, Tennessee (1998) [ORNL/TM-limited distribution].
22. Giffin, J., Greenstadt, R., Litwack, P., Tibbetts, R.: Covert messaging through TCP timestamps. In: Second Workshop on Privacy Enhancing Technologies. Volume 2482 of Lectures Notes in Computer Science., San Francisco, CA, USA, Springer-Verlag Heidelberg (2003) 194–208
23. Ka0ticSH: Diggin em walls (part 3) - advanced/other techniques for bypassing firewalls. New Order (2002) Retrieved on August 28, 2002 from the World Wide Web: http://neworder.box.sk/newsread.php?newsid=3957.
24. Rowland, C.H.: Covert channels in the TCP/IP protocol suite. Psionics Technologies (1996) Retrieved on November 13, 2004 from the World Wide Web: http://www.firstmonday.dk/issues/issue2_5/rowland/.
25. Ahsan, K.: Covert channel analysis and data hiding in TCP/IP. Master's thesis, University of Toronto (2002)
26. Ahsan, K., Kundur, D.: Practical data hiding in TCP/IP. In: Proceedings of the ACM Workshop on Multimedia Security at ACM Multimedia. (2002)
27. Servetto, S.D., Vetterli, M.: Codes for the fold-sum channel. In: Proceedings of th 35^{35} Annual Conference on Information Science and Systems (CISS), Baltimore, MD, USA (2001)
28. Servetto, S.D., Vetterli, M.: Communication using phantoms: Covert channels in the Internet. In: Proceedings of the IEEE International Symposium on Information Theory (ISIT), Washington, DC, USA (2001)
29. Anderson, R.: Stretching the limits of steganography. In Anderson, R., ed.: Information Hiding: Proceedings of the First International Workshop, Cambridge, U.K., Springer (1996) 39–48
30. Anderson, R.J., Petitcolas, F.A.: On the limits of steganography. In: IEEE Journal of Selected Areas in Communications: Special Issue on Copyright and Privacy Protection. (1998) 474–481

31. Craver, S.: On public-key steganography in the presence of an active warden. In Aucsmith, D., ed.: Information Hiding: Proceedings of the Second International Workshop, Portland, Oregon, U.S.A., Springer (1998) 355–368
32. Fisk, G., Fisk, M., Papadopoulos, C., Neil, J.: Eliminating steganography in Internet traffic with active wardens. In Oostveen, J., ed.: Information Hiding: Preproceedings of the Fifth International Workshop, Noordwijkerhout, The Netherlands, Springer (2002) 29–46
33. Cisco: Cisco IOS NetFlow. Retrieved on November 17, 2005 from the World Wide Web: http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html (2005)
34. Deri, L., Suin, S.: Improving network security using Ntop. In: Third International Workshop on the Recent Adcances in Intrusion Detection, RAID 2000, Toulouse, France (2000)
35. Case, J., Fedor, M., Schoffstall, M., Davin, J.: A simple network management protocol (SNMP). Retrieved on January 13, 2005 from the World Wide Web: http://www.ietf.org/rfc/rfc1157.txt (1990) RFC 1157.
36. IBM: Tivoli NetView. Retrieved on November 17, 2005 from the World Wide Web: http://www-306.ibm.com/software/tivoli/products/netview/ (2005)
37. HP: Network node manager advanced edition. Retrieved on November 17, 2005 from the World Wide Web: http://www.managementsoftware.hp.com/products/nnm/index.html (2005)
38. Sun: Solstice site manager. Retrieved on November 17, 2005 from the World Wide Web: http://www.sun.com/software/solstice/sm/index.xml (2005)
39. Doyle, J.: Routing TCP/IP. Volume I. Cisco Press, Indianapollis, IN 46240 (1998)
40. Shannon, C.E.: Communication theory of secrecy systems. Technical report (1949)
41. Deering, S., Hinden, R.: Internet protocol, version 6 (IPv6) specification. Retrieved on October 08, 2004 from the World Wide Web: http://www.ietf.org/rfc/rfc2460.txt?number=2460 (1998) RFC 2460.
42. Hinde, R., Deering, S.: IP version 6 addressing architecture. Retrieved on October 08, 2004 from the World Wide Web: http://www.ietf.org/rfc/rfc2373.txt?number=2373 (1998) RFC 2373.
43. (IANA), I.A.N.A.: Internet Protocol version 6 address space. Retrieved on October 29, 2005 from the World Wide Web: http://www.iana.org/assignments/ipv6-address-space (2005)
44. (IANA), I.A.N.A.: IP version 6 parameters. Retrieved on October 28, 2004 from the World Wide Web: http://www.iana.org/assignments/ipv6-parameters (2004)

A Covert Channels of Communication and Exploits

The description as well as the associated adversary model summarized in Subsections A.1 and A.2 correspond to the one presented in [10]. The *hop limit* exploit characterized in Subsection A.3 reassembles the Short TTL Attack for IPv4 reported by several authors [11,12,13,14].

A.1 Routing Header Covert Channels

The *Routing Extension Header* contains a list of intermediate routers a packet in transit should visit on the way to its destination. As the packet moves through the network, routers mark their addresses as “visited” and send the packet on to the next address in the list. The IPv6 Parameters document [44] enumerates three different types of routing, but only one of them, *Type 0*, is fully described in the specification [41]. Figure 8 shows

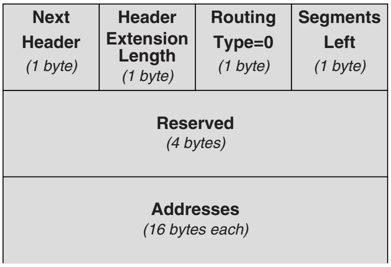


Fig. 8. Format of the Routing Header

Table 2. Identified Covert Storage Channels in the Routing Header

ID	Field	Covert Channel	Bandwidth
α	*Routing Type: 0 - Reserved*	Hide data in unused bits	4 bytes/packet
β	*Routing Type: 0*	Set one or more false addresses[6]	Up to 2048 bytes/packet

the format of the routing header when routing type is 0. Table 2 summarizes plausible covert channels exploiting such format.

α There exists a *reserved* field in the routing header structure when the *routing type* is 0. Alice can hide 4 bytes of covert data per packet using this channel.

β When the *routing type* is 0, Alice can fabricate "addresses" out of arbitrary data meaningful to Bob[7]. She appends the covert data and sets the `segments left` field accordingly. In most cases, she would like to prevent any node from attempting to process the fake addresses. Setting the segments left value to 0 will make the addresses to appear visited. Contrarily, a non zero value will indicate that such addresses need to be visited. Figures 9 and 10 display two different types of embedding in the routing header when the routing is 0:

- one where Alice chooses to create a completely new header to send Bob 48 bytes of covert information, and
- another one where she uses an already existing header to embed a covert message of 32 bytes.

Based on the maximum extension header payload length, Alice can potentially insert up 2048 bytes. Therefore, she will be extending the entire IPv6 packet by the same amount of bytes.

A.2 Hop Limit Covert Channel

The `hop limit` of the IPv6 header shown in Figure 11 indicates the number of hops a packet can still traverse before being destroyed. It is analogous to the TTL field in

[6] This covert channel, when authentication is used, requires recalculating or circumventing the ICV.

[7] In this situation, Bob does not need to be at the final destination of the packet. He only needs to observe the packet somewhere along the communication path.

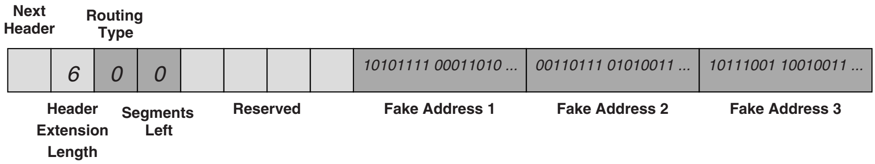
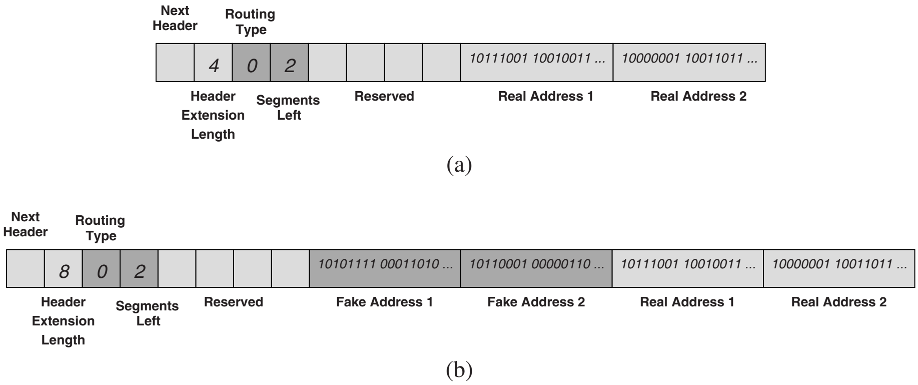


Fig. 9. β **Covert Channel in the Routing Extension Header**, when Alice creates fake addresses in a packet that did not originally a routing extension header



(a)

(b)

Fig. 10. β **Covert Channel in the Routing Extension Header**, when Alice inserts fake addresses in a packet already containing a routing extension header. (a) Original routing extension header, (b) Routing header after Alice inserts the covert data.

IPv4, however the TTL refers to the number of seconds remaining not the number of hops.

The *hop limit* channel[8] involves a crafted manipulation of its value. Alice send an initial hop limit value, h, and modifies the hop limit value of subsequent packets. Bob interprets the covert message by checking the variations in the hop limit values of packets traversing his location. One scheme has Alice signaling a 0 by decreasing the hop count from the prior packet, and a 1 by increasing the hop count relative to the prior packet. A drawback of this channel is that packets do not necessarily travel the same route, so the number of intermediate hops may vary, introducing noise. To overcome this, Alice can choose a δ that is greater than the expected noise, and use hop counts less than $h - \delta$ signal a 0, and hop counts greater than $h + \delta$ to signal a 1. Bob then compares the received hop count to h to deduce the bit. The bandwidth of this channel is limited. Alice needs to modify n packets to send $n - 1$ bits of information.

A.3 Short TTL Exploit

In the IPv4 context, an attacker can manipulate the packet's TTL field to mask another attack from a network intrusion detection system (NIDS) [11,12,13,14]. An appropriately

[8] This channel is called channel ϵ in [10].

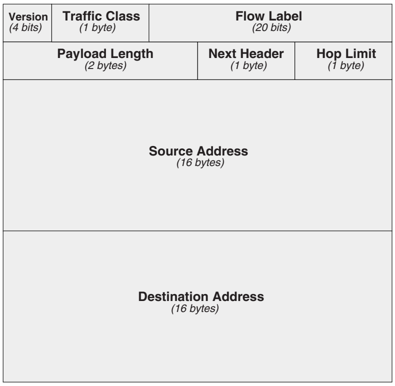


Fig. 11. IPv6 Header Format

set TTL value causes a packet to expire before it reaches its destination but after it has passed by any NIDS along the way. In consequence, the NIDS will see a different traffic pattern than the destination host will and might be unable to detect an ongoing attack. A similar mechanism can be applied to IPv6 traffic by exploiting the `hop limit` field in the IPv6 header (recall Figure 11).

B Rationale of the Percentages of Covert Channel Elimination

B.1 Routing Header Covert Channel

Case: Blind Adversary. Because *aggregatable global unicast* addresses must use the prefix 001, there is one in eight chance $(1/8)$ that a *blind* adversary will select a fake address that follows such pattern. Let $P_{Interception}$ be the probability of the active warden interception the adversary's covert communication,

$$P_{Interception} = 1 - \frac{1}{8} \tag{1}$$

In addition, every fake address the *blind* attacker wishes to inject to convey cover messages have to begin with the same pattern. Therefore, the odds of blocking a bogus address are higher with the next one inserted. That is,

$$P_{Interception} = 1 - \frac{1}{8^n} \tag{2}$$

where n is the number of injected addresses.

Case: Warden-Aware Adversary. A *warden-aware* adversary that attempt to circumvent the actions taken by an active warden has a unique alternative to manipulate the order of legitimate router addresses in the Routing Header.

Let $C_{Bandwidth}$ be the channel bandwidth measured in bits per packet, n be the number of addresses present in a Routing Header. The bandwidth of a Routing Header covert channel based on the order of the contained addresses is given by the equation,

$$C_{Bandwidth} = 128 * n \tag{3}$$

considering that each address has a length of 16 octets (128 bits).

However, if the attacker is forced to use only real router addresses, such bandwidth also depends on the number of routers, r, within the protected AS. That is,

$$C_{Bandwidth} = log_2(r^n) = n * log_2(r) \tag{4}$$

The ratio between 3 and 4,

$$\frac{128}{log_2(r)} \tag{5}$$

represents bandwidth loss the adversary will suffer when her actions are limited by the active warden.