

On the Vulnerabilities and Protection of OSPF Routing Protocol

Feiyi Wang

Electrical and Computer Engineering Dept.
North Carolina State University
Raleigh, NC 27695
fwang2@eos.ncsu.edu

S. Felix Wu

Computer Science Department
North Carolina State University
Raleigh, NC, 27695
wu@eos.ncsu.edu

Abstract

This paper analyzes both the strong points and weak points of OSPF routing protocol from security perspective. On its strong points, we abstract its features of information least dependency and information hiding, which make it very robust and fault resilient, even when facing certain malicious attacks. On its weak points, we take a pragmatic look at various problems centering round secure routing protocols. By carefully investigating a special re-routing attacking case, we show how an home-made malicious router can easily disrupt the service. It also provides a concrete example for routing protection and intrusion detection. Finally, we present the active protection idea and its architectural flexibility and compatibility advantages.

1. Introduction

With the growing awareness of network security, it is believed that more sophisticated attacks could be developed and aimed at the heart of Internet daily operation: routers. Instead of attacking just one particular host or even a subnet, which is usually the focus of traditional host and network security, the purpose of routing attack is to sabotage the network service and infrastructure.

Consider routing security from the viewpoint of failure mode, there are two extreme cases. One is simple *fail-stop* failure, such as link, interface or router up and down; another extreme is Byzantine failure, which assumes router can exhibit arbitrary behavior, usually with malicious intention. A well-known fact about many routing protocol designs is that they only consider simple failures, while security is usually an afterthought. Therefore, even those routing protocols which are very robust in a common sense, have many vulnerabilities when facing a strategically placed intruder. How to secure a routing protocol remains a chal-

lenging research issue.

OSPF [4, 5] is a link state protocol designed to be used in a single *Autonomous System*(AS). The link-state based technology was pioneered in ARPANET routing protocol [2, 3], as compared to the distance vector based routing protocol such as RIP [1] and BGP [7]. The basic idea is as follows: Each router is responsible for meeting their neighbors and learning their names; Every router constructs a packet known as *link state advertisement*(LSA), which contains a list of the names of their neighbors and the link cost to the neighbors. These LSAs are flooded to all other routers periodically, or whenever any new information is available. Each router uses these LSAs to form the complete view of topology, and computes the route to each destination.

The purpose of this paper is to take OSPF as an example, analyze both its strong and weak points. On its strong points, we highlight the features like *information independency* and *information hiding* as desired secure properties of routing protocols. These highlights also help to understand security problems in other protocols. On its weak points, in addition to the general concern about some particular fields (e.g. Age field, Sequence Number, etc.), we present a re-routing attacking case study which shows how a homemade malicious router can disrupt the service. In the final section of this paper, we review the current practice of routing protocol protection and present our approach: active protection, which has the advantages such as architectural flexibility and backward compatibility.

2. Security Strong Points of OSPF

Our analysis and experiments show that, as a routing protocol, three intrinsic mechanisms of OSPF make it very robust and resilient to failures, even to some malicious attacks. They are discussed below.

Flooding and information least dependency: As we mentioned before, LSAs are propagated by flooding; the

flooding algorithm is reliable, which ensures all routers in the same area have the same topological database. Consider either a single point (router) failure case or an intruder trying to fake or modify other router's information, as long as there is an alternate path, good routers can always receive the messages, though they could be conflict messages. This triggers an interesting phenomenon in OSPF: *fight-back*, good router try to *convince* bad router by keep sending them correct information, as observed in our previous work [10]. We think it is actually an advantage in the sense that *fight-back* could be easily spotted by an *Intrusion Detection System(IDS)*.

A more profound impact of flooding individual LSA is *information least dependency*: every router uses the *raw information* from the original advertiser instead of *aggregated information* from neighbors, which gives security advantages over other pure distance vector based routing protocols.

In a distance vector algorithm (e.g., RIP), each router sends only summarized information, which is computational results based on reachability information from its neighbors. These aggregate information has two implications. First, it is very hard for a router to validate the information it receives; Second, even if a router detects incorrect information, it is still difficult to determine the source of the information.

By comparison, for a link state routing algorithm such as the one used in OSPF, each router generates information about its local topology (e.g., its neighbors), and forwards such information to other routers via flooding. This has several advantages: every router independently possesses the entire topology information for the network and each router is responsible only for its own local portion of the topology, as long as any of its neighbor is honest, it can get raw independent information through one hop further. Obviously, information independency helps (compared to distance vector) to find out which router is lying. Also, it helps using authentication to verify the origin of a message. Recently, the method proposed in [8, 9] which uses predecessor-based information to harden distance vector algorithms, justifies the principle from another side: a predecessor is essentially a piece of information provided by source to alleviate the total blindness of router. By going through the predecessor, a router can help reconstruct the shortest path tree back to the source. It is the predecessor information that makes a secure distance-vector based algorithm possible.

Hierarchy routing and information hiding: The primary goal of hierarchical routing is to deal with routing scalability issues (reduce routing table size, link bandwidth and router computing resources). But, we also see it has both robustness and security advantages. OSPF is basically a two-level routing protocol: intra-area and inter-area routing, with ABR (area border router) connected to backbone

and exchanging area summary information. There are three cases we can consider here:

- *Internal router is compromised:* The implication of two-level routing is that an internal router does not need to know the topology of the outside except its own area. Consider the case that an internal router is compromised, the damage it can do is very much limited to the area, leave the routing in other areas functioning normally.
- *ABR (area border router) is compromised:* If there is only one ABR or this ABR is the only one attached to the backbone, then the area will suffer serious consequences from compromise. If there are other ABRs online, since all ABRs for an area should broadcast the same topology information, redundancy will provide an opportunity of doing mutual verification to detect conflict information.
- *ASBR (autonomous system border router) is compromised:* OSPF uses ASBR to import external routing information into OSPF routing domain. These external routing information will be flooded through the routing domain. What ASBR does is actually punch a *hole* in the area boundary. It is probably the worst single security vulnerabilities in OSPF and there is no easy way to fix it. Database overflow protection can prevent intruder from arbitrarily flooding in junk routes in certain degree, but incapable to detect false or fake routes.

Procedural checking and constraint: The checking procedure for OSPF protocol to accept a packet is rigorous. Generally, it must pass three checking gates, as illustrated in Figure 1. Taking IP checking stage as an example. First, the conventional IP checking such as checksum verification is done in kernel since OSPF run directly over IP; Then, OSPF must guarantee that the packet is not self-originated (IP source checking) and addressed either to one of its interface address or one of two multicast addresses: AllSPFRouters (224.0.0.5) and AllDRouters(224.0.0.6). After header checking, for each different types of packet, there are further validation procedures. The point we are trying to convey is: these validation procedures at one hand, are necessary for a protocol to operate correctly and be robust; on the other hand, it increases the difficulty of sabotage: the chance of simply pitching a packet and hoping it works is rare. In the next section we will see the work need to be done to carry out an attack.

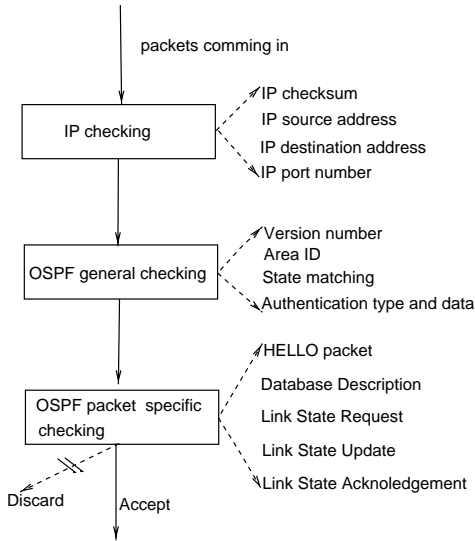


Figure 1. OSPF packet procedure checking

3. Security Weak Points from Running Environment

In this section, we consider several situations which will have a practical impact on the security of a router but may not be directly related to the design of routing protocol.

First, we make a distinction between *host based router* and *proprietary commercial router*. A host-based router usually runs both traditional operating system (OS) and some routing software, which give it the capability to forward packets. The difference between these two environments from security perspective is critical: the former environment is usually exposed to more threats because of the following reasons:

- Since a host-based router runs on traditional OS, all the vulnerabilities of that particular OS can be exploited as the entry door to gain privilege access, while the low-level architecture of commercial router is less well known, therefore less vulnerable.
- A host tends to provide more network services to the outside since people are using it for multiple purposes, these network services could be the weak points of security.

Implementation could be another problem. Many factors determine that real world implementation may not faithfully follow the protocol specification. Therefore, bugs may be introduced indirectly and manifest themselves as a security vulnerability. Our recent study [10] reveals that a glitch in public domain OSPF routing software and one commercial routing software proved to be a serious security threat.

Configuration problems are probably one of the most obvious and common mistakes. A poorly configured router often opens the door for intruders exploiting its vulnerabilities.

With all the arguments above, the relevance to the routing protocol design is that reduced complexity and simple design could make both the implementation and configuration easy, therefore reduce the risk.

If the attacking on routing protocol ever happens, where could be the launching points? We think there are three possible places.

- A compromised router. It could be either a host-based or commercial one. Since the router is previously a legitimate one, it holds all the information from the past and could be exploited for further misuse.
- An intruding device, which is assumed to have certain tapping points (physical access) and must have the capability to join into routing process, which requires some basic function such as HELLO protocol, Database exchange protocol, and certain state transition maintenance. Experience shows that without really participate into the routing domain, attacking result will hardly to be fruitful.
- A host which does not join the routing domain and therefore is invisible from the routing context. This type of attacks are rare, but under certain conditions, it can happen and hard to detect and prevent[10].

4. Protocol Vulnerabilities: An Case Study

General protocol vulnerabilities of OSPF have been analyzed in [6]. There are three fields: *metrics*, *sequence number* and *age* which are particularly vulnerable and therefore are the targets of usual attacks. However, with Keyed-MD5 protection, most of these vulnerabilities could be eliminated except the age field. We are more concerned about *man-in-the-middle* attack: what will happen if one of the router is compromised? The following case study gives a concrete example of this type of attack.

The goal of this test is to show that an intruder could maliciously reroute traffic with a *homemade* router by tricking other legitimate OSPF routers in a simplified but real network. Here, we assume an intruder has total control of its own facility - a malicious OSPF routing process; However, the intruder can not change the behavior of either hosts or other legitimate routers except by cheating. Figure 2 shows the network topology configuration.

We have two hosts: H_1 and H_2 , connected by two routers: R_1 and R_2 . The default router for H_1 is R_1 , and that for H_2 is R_2 . R_3 could be either a compromised router or an intruding device. So the route from H_1 to H_2 is

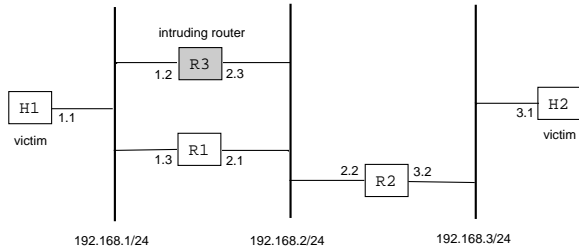


Figure 2. Network topology for re-routing

$H_1 \rightarrow R_1 \rightarrow R_2$. The attack needs to go through two phases:

Phase 1: ICMP redirect attack on H1 The *Internet Control Message Protocol (ICMP)* is the control and error message protocol for IP. One particular class of ICMP message is router-to-host redirect control message which is usually used to help host to find a *shortcut* on attached network. Since H_1 's default router is R_2 , by sending ICMP redirect packet with impersonated source R_1 , R_3 can change the default router for H_1 . Figure 3 shows the change in H_1 's kernel routing table before ICMP attack and after ICMP attack. Notice that after the ICMP attack, the Flags change from UGSc to UGmc. Here S stands for static route and M stands for modified route (by ICMP). Also, in most systems, these kinds of dynamically generated routes will expire after some time period, which requires R_3 periodically refresh the information to keep it in the kernel.

Destination	Gateway	Flags	NetIf	Expire
default	192.168.1.3	UGSc	ed0	
localhost	localhost	UH	lo0	

(a) Before ICMP attack: routing table of H1

Destination	Gateway	Flags	NetIf	Expire
default	192.168.1.2	UGmc	ed0	
localhost	localhost	UH	lo0	

(b) After ICMP attack: routing table of H1

Figure 3. Routing Tables before and after the attack

Phase 2: R3 Launch malicious OSPF process. This malicious OSPF process must have at least the following capabilities: using HELLO protocol to bring up the adjacency with neighbors (in this case R1 and R2) and maintaining this neighbor relationship by periodically

sending HELLO packet; Advertise a smaller interface cost less than R1 (in our test, R1 is using link cost 10 on both of its interfaces, while R3 advertises link cost 5 on both of its interfaces). After new rounds of synchronization of link state database and SPF tree computation, the new kernel routing table is shown in Figure 4.

Destination	Gateway	Flags	NetIf	Expire
...				
192.168.1/24	192.168.2.1	UGc	ed1	

(a) Before malicious OSPF starts on R3

Destination	Gateway	Flags	NetIf	Expire
...				
192.168.1/24	192.168.2.3	UGc	ed1	

(b) After malicious OSPF starts on R3

Figure 4. Routing Tables before and after the attack

Using `traceroute` utility also confirms that all the traffic from H_1 , will now follow $H_1 \rightarrow R_3 \rightarrow R_2$. The difficulty of this attack is modest. With today's widely distributed public domain routing software, it is quite easy to launch such attack. If R_3 is a compromised router instead of an intruding device, it becomes even harder for routing protocol to protect themselves. In the following section, we discuss possible ways to protect routing protocols.

5. Routing Protections

5.1. Related Works

The main thread of research activity centers around how to use cryptography to solve routing security problem. This includes Smith's work on secure RIP and BGP [8, 9] and Murphy's work on introducing digital signature into OSPF[6]. The basic idea of their work is to encrypt every routing control message with a secret key, only the router who has the correct key can decrypt it. These methods are effective to defend outsider attack, but can not deal with insider attack: situations such as router is compromised and key is in exposure. In addition, they suffer the following technical and non-technical problems:

- It is often impossible to design a absolutely routing protocol. In addition to the misunderstanding of protocol specification and accuracy of specification itself, we also face such problems as buggy implementation

[10], poor policy and administrative practice, backward incompatibilities of multi-vendor platform, etc.

- Complexity and performance penalty unacceptable: public key infrastructure is one of the strongest cryptography technique we can utilize, but it can be hardly accepted in a high-speed routing environment due to complexity such as key distribution and management and expensive encrypt/decrypt process.

Currently, OSPF specification [5] provides two authentication methods: simple password and Keyed-MD5. Obviously, simple password authentication is vulnerable to passive attacks currently widespread in the Internet. Keyed-MD5 is an effective way to defend outsider attacks, but still vulnerable to insider attacks.

5.2. Our Approach: Active Protection

Active protection regards the protected router as a wrapped object; the protector itself is referred to as *secure wrapper*. There are *different levels of checking gates* for input and output between protected router and the secure wrapper. The status of these checking gates will be determined by security policy database, which in turn is controlled by either a security officer or other trusted IDSs.

The basic idea is that secure wrapper will intercept both input and output stream of protected router, It also examines the content of these streams, establish corresponding internal states which will match the protected router or routing protocol and makes active response based upon the analysis result of input and output streams. The difference between this architecture and other *passive data collection and passive protocol analysis* based IDS is that it will do active response. For example, if Keyed-MD5 is not a desirable way to do authentication in a particular environment, or an old routing system does not have such authentication support, secure wrapper will establish such secure channel with its neighbor counterparts, attach additional authentication data to the packets, and remove it at the other ends. All of these must be done in a *transparent* way, that is, the protected router will be totally unaware of the existence of secure wrapper. Here we assume that there will be some kind of key negotiation between secure wrapper themselves, maybe with the help from security officer or policy database.

The advantages of this approach mainly come from its *architectural flexibility* and *compatibility*. By architectural flexibility, we can harden the routing protocol without having to heavily rely on the standard or implementation limits. Also, it facilitates us to experiment different protection schemes before it gets integrated into the routing protocol itself. By compatibility, those “legacy systems” which do not have any security protection in mind, will be strengthened

transparently. We are currently investigating this approach in our internal test bed in SHANG group.

6. Summary

This paper analyzes both the strong points and weak points of OSPF routing protocol from security perspective. On its strong points, we highlights the idea of *information least dependency* and *information hiding*, which make it very robust and fault resilient, even for certain malicious attacks. On its weak points, we take a pragmatic look at various problems centering round secure routing protocols. Especially, we investigate an attacking case in a real network and show that how easy a “homemade” router can conduct evil activities. Finally, we present our idea about “active protection”, which have the advantages of both architectural flexibility and backward compatibility.

Acknowledgements

This project is supported by U.S. Department of Defense Advanced Research Projects Agency and the U.S. Air Force Rome Laboratory under contract F30602-96-C-0325. Authors sincerely acknowledge the support from Dr. Frank Jou (MCNC) and other SHANG group members.

References

- [1] C. Hedrick. Routing information protocol, Internet RFC 1058, June 1988.
- [2] J. McQuillan, G. Falk, and I. Richer. A review of the development and performance of the ARPANET routing algorithm. *IEEE Transactions on Communications*, 26(12):1802–1811, 1978.
- [3] J. McQuillan, I. Richer, and E. Rosen. The new routing algorithm for the arapanet. *IEEE Transactions on Communications*, 28(5):711–719, May 1980.
- [4] J. Moy. OSPF specification, RFC 1131, October 1989.
- [5] J. Moy. OSPF version 2, Internet RFC 2178, July 1997.
- [6] S. Murphy and M. Badger. Digital signature protection of the ospf routing protocol. In *IEEE/ISOC Symposiums on Network and Distributed System Security*, 1996.
- [7] Y. Rekhter and T. Li. A border gateway protocol 4 (BGP-4), Internet RFC 1771, March 1995.
- [8] B. R. Smith and J. Garcia-Luna-Aceves. Securing the border gateway routing protocol. In *Proc. Global Internet'96*, London, UK, November 20-21 1996.
- [9] B. R. Smith, S. Murthy, and J. Garcia-Luna-Aceves. Securing distance-vector routing protocols. In *IEEE/ISOC Symposiums on Network and Distributed System Security*, 1996.
- [10] B. Vetter, F. Wang, and S. Wu. An experimental study of insider attacks on the ospf routing protocol. In *IEEE International Conference on Network Protocol(ICNP)*, pages 293–300, Atlanta, USA, October 1997.