Persistent OSPF Attacks

Gabi Nakibly
National EW Research & Simulation Center
Rafael – Advanced Defense Systems Ltd.

Alex Kirshon and Dima Gonikman
Computer Science Department
Technion – Israel Institute of Technology

Dan Boneh*
Computer Science Department
Stanford University

Abstract

Open Shortest Path First (OSPF) is the most widely deployed interior gateway routing protocol on the Internet. We present two new attacks on OSPF that expose design vulnerabilities in the protocol specification. These new attacks can affect routing advertisements of routers not controlled by the attacker while evading the OSPF self-defense "fight-back" mechanism. By exploiting these vulnerabilities an attacker can persistently *falsify large portions of the routing domain's topology thereby giving the attacker control over how traffic is routed in the domain. This in turn can lead to denial of service, eavesdropping, and man in the middle attacks. We discuss a number of mitigation strategies and propose an update to the OSPF specification that defeats these attacks and improves overall OSPF security.*

1 Introduction

Open Shortest Path First (OSPF) is the most popular interior gateway routing protocol on the Internet. Its aim is to allow routers within a single autonomous system (AS) to construct their routing tables, while dynamically adapting to changes in the autonomous system's topology. OSPF is currently used within most autonomous systems on the Internet. It was developed and standardized by the IETF's OSPF working group. This work is concerned with version 2 of the protocol [9] which was specifically designed for IPv4 networks, hence it is practically the only version used today. Version 3 [4] has been standardized to accommodate IPv6 networks, in which the fundamental mechanisms of version 2 have been kept.

*Supported by NSF and a MURI grant.

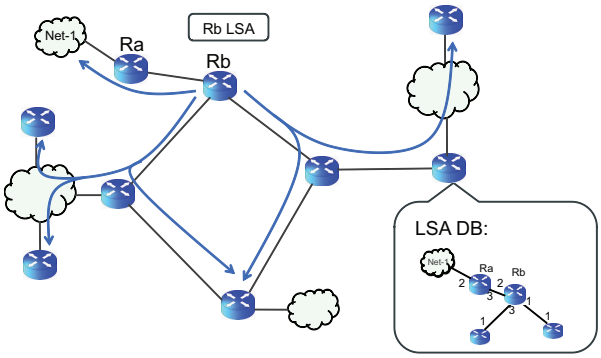


Figure 1. An example of LSA flooding

OSPF is a link-state routing protocol which means that each router advertises its links to neighboring routers and networks as well as the links' costs. These advertisements are termed Link State Advertisements (LSAs). The cost of a link is usually statically configured by the network administrator. Each LSA is flooded throughout the AS where a router receiving an LSA from one of its neighbors resends it to all its other neighbors. Every router compiles a database of the LSAs of all routers in the AS. The databases are identical on all routers. Using this database a router obtains a complete view of the AS topology. This allows it to employ Dijksatra's algorithm [5] to calculate the least cost paths between it and every other advertised network or router. As a result, a next hop is derived for each destination, which forms the router's routing table.

Figure 1 illustrates the flooding of an LSA throughout the AS while the routers build their LSA database to construct their view of the AS topology.

In this work we present two new powerful attacks that exploit the functionality of OSPF. The attacks significantly advance the state of the art and shed new light on the security weaknesses of OSPF. The attacks exploit design vulnerabilities of the protocol specification as defined in [9]. We emphasize that the attacks do not rely on implementation vulnerabilities and consequently *all OSPF routers may be vulnerable to these attacks*. The attacks enable an attacker to persistently falsify LSAs of OSPF routers not controlled by the attacker. Previous OSPF attacks [11, 12] that attempt to do that trigger the "fight-back" mechanism by the victim router which advertises a correcting LSA thereby making the attacks' effect non-persistent. Consequently, it is a common misconception that an attacker – even an insider – cannot persistently falsify LSAs of routers it does not control.

The attacks presented here are the first to evade the "fight-back" mechanism. They enable an attacker to persistently subvert the view that other routers have of the AS topology and consequently affect their routing tables. Gaining persistent control over the routers' routing tables lets an attacker divert traffic away from its intended routes and enables a number of attacks on the AS. The first is denial of service where the attacker's goal is to degrade the network's ability to forward traffic with a desirable quality of service. The attacker can do so using one of the following strategies:

1. Link overload – Diverting large volume of traffic thorough a limited capacity link.
2. Long routes – Diverting traffic over unnecessarily long routes while wasting network resources.
3. Delivery failure – Making some portion of the network mistakenly believe that it is disconnected from the AS.
4. Routing loops – Routing traffic in loops between two or more routers while consuming network resources before being dropped.
5. Churn – Changing traffic routes rapidly while resulting in a network instability and performance degradation of congestion control mechanisms (e.g. TCP).

Another potential attacker goal is eavesdropping. Here the attacker can divert remote traffic to pass through a router or a network the attacker has access to thereby letting the attacker eavesdrop on the traffic. Traffic diversion may also facilitate man-in-the-middle and impersonation attacks.

As in most previously published OSPF attacks we assume the attacker has the ability to send LSAs to routers in the routing domain and that routers process them as valid LSAs. This can be done by an insider, namely an attacker who gains control over a *single* router in the AS. The attacker can gain control of a router by conspiring with an authorized personnel having physical access to the router or by remotely exploiting an implementation vulnerability on the router. Several such vulnerabilities have been published in the past (e.g., CVE-2010-0581, CVE-2010-0580, and CVE-2009-2865).

The paper is organized as follows. Section 2 gives a brief overview of the OSPF specification and principal functionality. Section 3 reviews known attacks that exploit design vulnerabilities of OSPF. Section 4 presents the new found attacks. Section 5 evaluates the power of attacks and their effects on real-world AS topologies. Section 6 proposes mitigation measures and Section 7 concludes the paper.

2 OSPF Basics

We begin with an overview of the OSPF protocol. We only give here the background needed to understand our attacks. We focus on version 2 of the protocol as specified in [9]. Note that OSPF does not use a TCP/IP transport protocol, its messages are encapsulated directly in IP datagrams with protocol number 89. OSPF handles its own error detection and correction functions.

2.1 Protocol Fundamentals

OSPF is a link state routing protocol: each router advertises an LSA containing the links to neighboring networks and routers and their associated costs. Each LSA is flooded throughout the AS. Routers construct a complete view the AS topology by compiling all the LSAs they receive into a single database. From this global view routers compute their routing tables.

Each LSA is advertised periodically every 30 minutes, by default. An LSA includes a Sequence Number field which is incremented for every new instance. A fresh LSA instance with a higher sequence number will always take precedence over an older instance with a lower sequence number. In addition, an LSA includes an Age field indicating the elapsed time since the LSA's origination. When it reaches 1 hour the LSA instance is removed from the LSA database.

A local network having two or more routers directly attached to it is called a *transit network*. A router connected to a transit network advertises a link to the network rather than to the neighboring routers. In addition, one of the neighboring routers is chosen to act as a *designated router*. This router advertises an LSA on behalf

of the local network, in addition to its own LSA, advertising links back from the network to all the routers attached to the network (including itself).

A router dynamically discovers its neighbors using a Hello protocol. A router periodically multicasts Hello messages on its attached links. The message includes the identities of all the routers from which it has received Hello messages. After mutual discovery two neighboring routers may set up a special relationship called an *adjacency*. To alleviate memory and processing load an adjacency is set up only when one of the two peers acts as a designated router. The purpose of a setting up an adjacency is to make sure that the two routers have identical copies of the LSA database. This is done by having each router send to its peer the summaries of LSAs currently installed in its database. The summaries are sent using Database Description (DBD) messages. At the beginning of the exchange the two routers negotiate their master/slave status. The router with the higher ID is chosen to be the master. The exchange of the database description packets is done in a stop-and-wait fashion. A router sends its next message only after it receives one from its peer. To distinguish between database description messages, a sequence number is included in every message. The sequence number is initialized arbitrarily by the master and incremented by the master with every new message it sends. The slave sends its messages with a sequence number that equals to the last message received from the master. A DBD message includes 3 flags: I, M, and MS. The 'I' flag is set to indicate a master/salve negotiation. The 'M' flag is set to indicate the router has more LSA summaries to send. The 'MS' flag is set to indicate the router declares himself to be the master.

Once the exchange has finished a router may request its peer newer instances of LSAs the router does not have. After the router receives these newer LSAs the adjacency is set up and the router enters the Full state. From this point on the router will include in its LSA a link to its peer. Figure 2 depicts an example of adjacency set up. In this example we assume R1 is chosen to be the master.

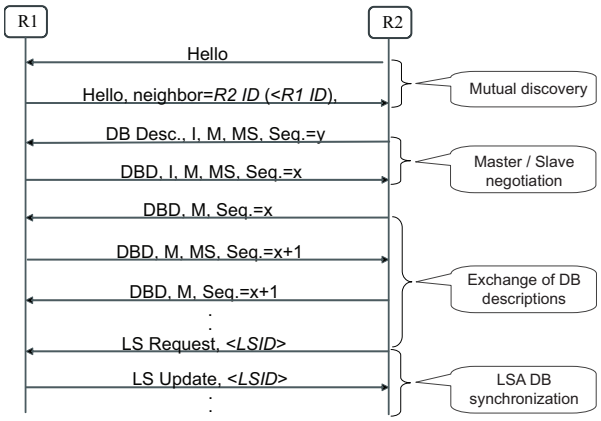


Figure 2. An example of an adjacency set up between two routers

2.2 OSPF's Security Strengths

We next list the dominant security strengths of OSPF and explain the difficulties the attacker has – even as an insider – to persistently falsifying LSAs of router it does not control.

1. Per-link authentication – Every OSPF packet sent on a specific link may be authenticated. The authentication is based on a secret shared by all the routers directly attached to that link[1]. At every hop the OSPF packet is re-authenticated using the secret of the current link. This prevents an OSPF packet originated by an outsider from being processed. Due to lack of defined secret key management mechanism, a network operator must manually configure the secrets at every router [8] this leads to a situation where for many ASs today the secret is the same for all their links.
2. Flooding – Every LSA is flooded throughout the AS. Consequently, a malicious router can not prevent an LSA from reaching other routers as long as there is a path from the originator of the LSA that does not go through the malicious router.
3. "Fight-back" – Once a router receives an instance of its own LSA which is newer than the last instance it originated, it immediately advertises a newer instance of the LSA which cancels out the false one. This mechanism prevented all previously published OSPF attacks from persistently and stealthily falsifying an LSA of a router the attacker does not control.
4. LSA content – An LSA holds only a small part of the topology; only the links to its immediate neighbors. Therefore, in order for an attacker to significantly influence a router's view of the AS topology and consequently influence its routing table it must falsify many LSAs of many routers in the AS.

[1] OSPF computes the packet integrity tag as MD5(data||key||pad||length) where || denotes concatenation. While this integrity method is now known to be insecure, we do not use this fact in our attacks. OSPF does not use HMAC for historical reasons.

5. Bidirectional links – Only if a link is advertised by both its ends will it be taken into account during the routing table calculation. An attacker advertising a non-existing link to another router will not influence the routing tables since that other router will never advertise a link back to the attacker.

3 Previous Attacks on OSPF

There are a few past works that present attacks exploiting design vulnerabilities of the OSPF protocol. In the following we focus on previously published attacks that falsify LSAs. All the attacks we list assume the attacker is an insider which possesses the secrets of its directly attached links.

The most common attack vector aimed at falsifying LSAs is the one in which the attacker falsify the LSA of the router it controls. It is a very convenient attack vector since a "fight-back" will never be triggered. However, this is a very limiting attack vector since only one LSA can be falsified. Wang et al. [11] present one example of such an attack in which the attacker impersonates a router that resides on the border of the AS while advertising an LSA with links to destinations outside the AS. The result is that some or all the traffic destined to those destinations will be attracted to the attacker. This way the attacker can black-hole the traffic, eavesdrop on it, or just divert it through a longer route. This attack has the disadvantage that it can not influence traffic to destination internal to the AS. A router will always prefer an AS internal route than an external one. In addition, this attack can only attract traffic to the attacker. No real control of the routing tables is achieved.

Another attack vector is one in which the attacker sends out false LSAs on behalf of routers it does not control. Wu et al. [12] describe several such attacks (e.g. Seq++ and MaxSeq). All the attack variants described in [12] trigger a "fight-back" by the victim router reverting the attacks' effects. This can be leveraged by the attacker to make the routing process in the AS unstable. However, the attacks do not enable an attacker to persistently and stealthily falsify the view the routers' have on the AS topology. The attacks also dramatically increase the exposure of the attacker and the chances of being discovered.

Jones et al. [6] introduce the first and only known attack which evades the "fight-back" mechanism. The attack exploits vulnerability in the OSPF specification which mutes a victim router from originating a correcting LSA if it receives its own LSA at a high rate (at least 1 packet per 5 seconds). It is evident that the attack dramatically increases the chances of the attacker's detection.

Another attack vector is one in which the attacker sends out false LSA on behalf of a phantom router – a router that does not exist on the AS. This attack vector will not trigger a "fight-back". On the other hand, it will not influence the routing tables because of the bidirectional requirement; no real router will advertise a link back to a phantom router. In [6] such attacks are discussed, but their sole purpose is to overflow the routers' LSA databases.

4 The New Attacks

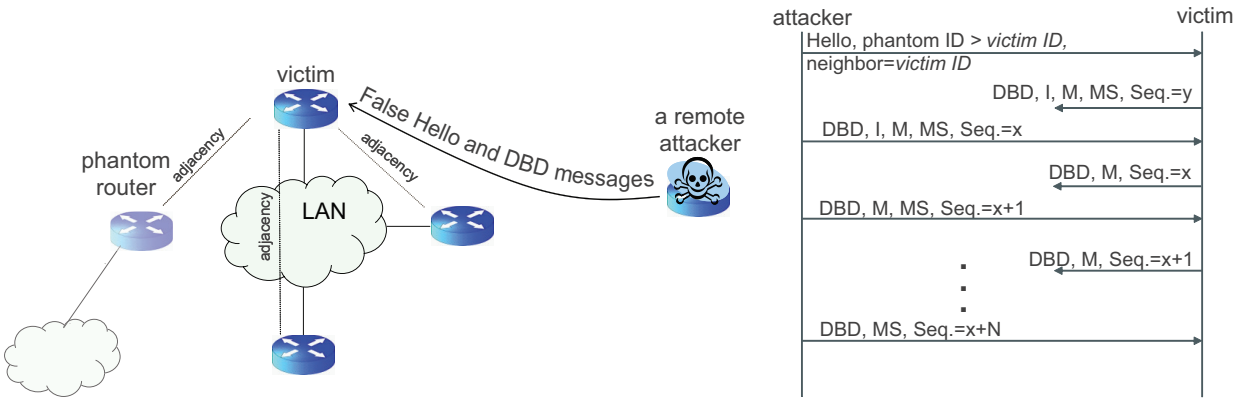
We now present two new attacks on OSPF. The first, called *Remote False Adjacency*, enables an attacker to fool a remote router into advertising a non-existing link in its LSA. This attack assumes that routers in the AS are configured with the same secret keys on all links. The second attack, called *Disguised LSA*, is more powerful and enables an attacker to fully control the entire content of an LSA of a remote router. This attack makes no assumptions about the secret keys of the AS links. We describe each attack in turn.

4.1 Remote False Adjacency

Section 10.8 of the OSPF spec [9] describes the procedure for sending database description packets during the adjacency set up process. A careful review of the section shows that a master router can successfully complete the adjacency set up without ever seeing messages sent by its peer – the slave router. This means that an attacker who controls one router can send spoofed OSPF messages to a remote victim router and confuse the victim into setting up an adjacency to a phantom (non-existent) router on the victim's local network (see Figure 3(a)). The attack is successful even though the attacker cannot see messages that the victim sends to the phantom router. Figure 3(a) illustrates the locations of the attacker, victim and phantom routers.

Since OSPF adjacencies can only be established with routers on the same subnet, the attacker must impersonate a phantom router located on the victim's local network. Moreover, the victim router should be the designated router of its local network to ensure it is willing to set up an adjacency with the phantom router.

After the attack is launched and the victim router is adjacent to the phantom router the victim advertises on behalf of the local network an LSA containing a link to the phantom router. This is the crux of the attack and its



(a) The victim router is fooled into setting up an adjacency with the phantom (non-existent) router

(b) The sequence of attack messages

Figure 3. The remote false adjacency attack

main benefit. Assuming the attacker advertises on behalf of the phantom router a link from the phantom back to the local network the bidirectional requirement will be met. Thereby the non-existing link will be taken into consideration by all other routers on the AS during their routing table calculation. This is the first published attack to successfully create a persistent bidirectional link between a real router or a network and a phantom router.

The attack sequence is shown in Figure 3(b) and proceeds as follows. In all the attack steps the attacker sends packets that appear to be coming from the phantom router and are destined to the victim. More precisely, the source IP address is always set to the address of the phantom router, a fictitious address in the subnet of the victim's local network. The destination IP address is set to the IP address of the victim's interface attached to that network.

The attack begins by sending a Hello message to the victim router while claiming to have previously received the victim's Hello messages. The attacker chooses for the phantom an ID that is numerically larger than the ID of the victim. Since the victim is assumed to be a designated router it starts setting up an adjacency with the phantom by immediately sending a DB description (DBD) message with an arbitrary sequence number. This message and all other messages sent by the victim are not received by the attacker since they are destined to the IP address of the phantom router.

Next, the attacker sends its first DBD message. In that message the attacker (masquerading as the phantom) claims to be the master of the exchange and suggests a different sequence number. The phantom is elected to be the master since it has a higher ID. Consequently, the victim adopts the sequence number suggested by the phantom. The attacker proceeds by repeatedly sending DBDs with increasing sequence numbers. We note that while constructing the DBD messages sent by the phantom the attacker need not see the contents of the DBD messages sent by the victim. For the sake of simplicity the attacker sends empty DBDs having no summary LSAs. To successfully complete the protocol and establish the adjacency the attacker must end the DBD exchange only after the victim sends out all summaries of its LSA database. Since the attacker does not receive the victim's DBD messages it does not know when the victim is finished, but fortunately this is not a problem. Even if the attacker continues sending DBD messages after the victim is done the victim will simply respond with empty DBD messages. Hence, the attacker need only upper bound the number of DBD messages needed by the victim to send its database content. The upper bound does not need to be tight and can be arbitrarily large[2]. After the attacker (phantom) sends its last DBD message the victim will not request LSAs from the phantom since it considers its database empty (the phantom's DBD messages were all empty). At this point the victim successfully ends the adjacency set up. From this point onwards the victim will advertise a link to the phantom router on behalf of its network.

[2]Since the attacker is assumed to reside on the victim's AS the databases of the two routers should be the same. As a result, the attacker has a fairly precise approximation for the number of DBD messages the victim will need in order to convey its database content.

Attack consequences. The attack can be exploited to black-hole traffic destined to a specific subnet. This can be done by having the phantom router advertise a link to the subnet to be black-holed. This will attract traffic to the phantom from near-by routers. Since the attacker can create phantom routers anywhere it wishes on the AS it can essentially black hole all the traffic destined to that subnet while sourced from anywhere on the AS. This application is illustrated in Figure 4(a). The normal routes to the subnet (10.10.0.0/16) are depicted by the solid lines. The dotted lines indicate the diverted routes to the phantom routers.

Another potential use of the attack is to place the phantom router in a strategic "location" on the AS allowing it to appear as a desirable shortcut for large volumes of traffic. For example, the same phantom router can be linked to two distant networks on the AS as shown in Figure 4(b). This can be done by targeting two victim designated routers of those two networks while using the same phantom router ID.

4.1.1 Caveats and Assumptions

The remote false adjacency attack has the following caveats:

1. The false Hello and DBD messages are remotely unicasted directly to the victim. Therefore, the attacker must know the secret authentication key of the victim's local network. Since the attacker knows only the secret keys of the links directly attached to it the attacker must assume that all links on the AS have the same secret. As we noted above this assumption is indeed true in many cases, however it does not have to be so.

2. The adjacency must be continuously maintained by sending a Hello message every time interval defined by the victim's RouterDeadInterval parameter. This parameter has a default value of 40 seconds [9]. If the victim does not receive a Hello message within that time interval it will tear down the adjacency.

3. Following the adjacency setup, the victim floods LSAs to the phantom and expects to receive LSA acknowledgments in return. According to [9] if an adjacent router does not respond with an acknowledgment to an LSA the router will indefinitely retransmit the LSA. Nonetheless, we observed that a Cisco router gives up after 125 seconds of retransmissions and tears down the adjacency. Since the attacker also receives every LSA sent by the victim, the attacker can spoof the acknowledgment messages: for each LSA it receives it has 125 seconds to send an acknowledgment.

4.2 Disguised LSA

Section 13.1 of the OSPF spec [9] states that two instances of an LSA are considered identical if they have the same values in the following three fields:

Sequence Number, Checksum, and Age.

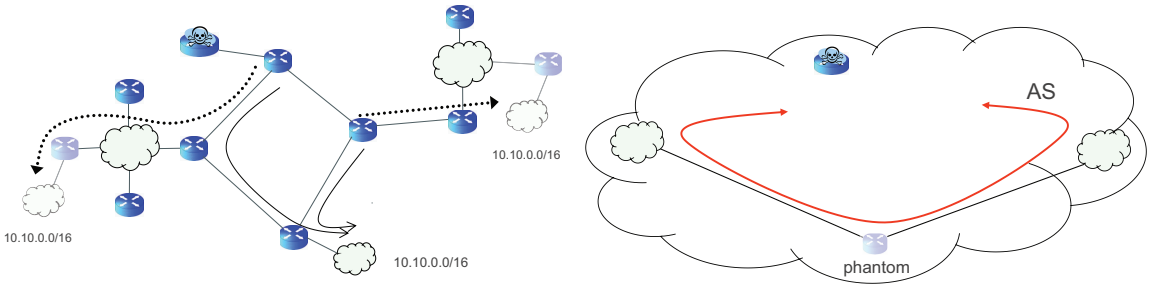
In fact, two LSAs are considered identical even if their Age fields differ by up to 15 minutes (and the Sequence Number and Checksum fields are the same). The key point is that the spec considers these two LSAs to be the same even if the actual advertised links in the LSAs differ.

A naive exploitation of this feature is to advertise an LSA with false links on behalf of a victim router while having the same values of the above three fields as the valid LSA advertised by the victim[3]. We call this false LSA a *disguised LSA*. When the victim receives the disguised LSA it will not fight back since it will consider it to be an identical copy of the last instance it advertised. Unfortunately, all other routers on the AS will also consider the disguised LSA as a duplicate and will not install it in their LSA database.

A better approach for the attacker is to advertise a disguised LSA that matches a recently generated LSA that has yet to be installed by all routers on the AS. On the one hand, the victim will consider the disguised LSA a duplicate of the fresh instance it just generated and will not activate the "fight-back" mechanism. On the other hand, other routers who have not yet received the new valid LSA will treat the disguised LSA as a new valid instance and install it in their databases. Once they receive the true valid LSA they will reject it as a duplicate.

One implementation of this approach is for the attacker to wait for a new valid instance advertised by the victim router. Once the LSA is received the attacker will flood the disguised LSA to its neighbors (rather then the valid LSA). This implementation enables the attacker to poison routers, i.e. make them install the disguised LSA, on the part of the AS that is farther from the victim router. Figure 5(a) illustrates the effect of this attack. The dotted arrows depict the propagation of the valid LSA until it reaches the attacker. The solid arrows depict the propagation of the disguised LSA. The shaded

[3]Note that the attacker must still make sure that the payload of the false LSA will be checksummed correctly. We explain how the attacker can achieve this latter in this section.



(a) Dashed lines show diverted traffic intended for the victim network (10.10.0.0/16). Traffic is diverted to the semi-transparent phantom routers.

(b) Black-holing most AS traffic by creating a phantom router adjacent to two remote networks. All traffic between these two networks will be routed incorrectly through the phantom.

Figure 4. Potential use cases for the remote false adjacency attack

routers are poisoned ones. These routers installed the disguised LSA since they received it before the valid LSA. It should be noted that the two LSAs are in a race. The first LSA to be received by a router is installed and continues to be flooded. The second LSA to arrive is rejected as a duplicate.

Figure 5(a) shows that this implementation is rather restricted. In most scenarios the attacker is able to poison only a small part of the routers on the AS. This is because the valid LSA is originated well before the disguised LSA.

A better implementation is to send the disguised LSA so it arrives at the victim immediately after it originates the next valid LSA. This way the disguised LSA starts the race well before the valid LSA does. At first this may seem impractical due to the precise timing requirements needed. However, observe that the attacker can exploit the "fight-back" mechanism to deliberately trigger the generation of a valid instance of an LSA. Just before sending the disguised LSA the attacker floods a false LSA which is *not* disguised. Its purpose is to intentionally trigger a "fight-back" LSA, i.e., the next valid instance of the LSA. We call this LSA a *trigger LSA*. As the trigger LSA is flooded throughout the AS it is installed by the routers. Immediately following it, the disguised LSA is received and is installed over the trigger LSA since it is disguised to the next valid instance[4]. The victim will first receive the trigger LSA and immediately send the next instance of the LSA. The disguised LSA which follows will be rejected by it as a duplicate of the instance it just originated. As a result, many more routers can be poisoned by the attacker. Figure 5(b) illustrates the effect of the attack. The solid arrows depict the propagation of the trigger and the disguised LSA until they reach the victim. The dotted arrow depicts the propagation of the "fight-back" LSA. The shaded routers are poisoned ones. These routers received the disguised LSA before the "fight-back" LSA. It is evident that using this technique a router can poison many more routers as compared to the previous technique. However, not all routers can be poisoned as evident form the figure. The relative locations of the attacker and the victim determine which routers will be poisoned.

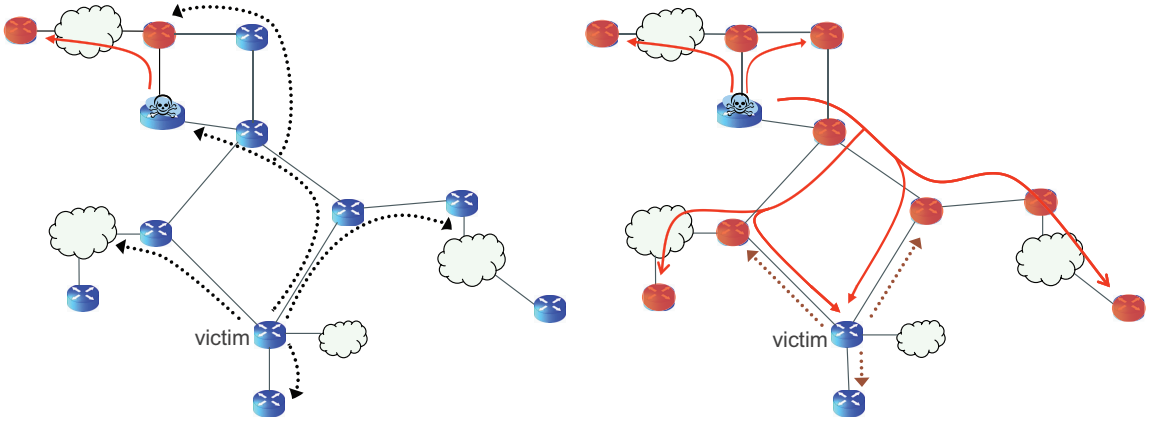
One difficulty is that the attacker needs to craft an LSA disguised to a future instance it has not yet seen. Fortunately, all three relevant fields of the future instance – Age, Sequence Number, and Checksum – are predictable:

1. Age – For all practical purposes the attack ensures that the "fight-back" LSA will be originated within 15 minutes of the origination of the disguised LSA. Hence, the Age field of the disguised LSA can be simply set to 0.

2. Sequence Number – The OSPF spec [9] defines the "fight-back" LSA to have a sequence number value that is larger than the trigger LSA by one. Hence, the disguised LSA must have that sequence number value.

3. Checksum – We note that the entire payload as well as the header of the next valid instance is deterministic and predictable. Hence, its checksum value is predictable as well. The attacker can now add to the false links it wishes to advertise a dummy link entry. The value of this entry will be set in such

[4]We note that the disguised LSA must not follow the trigger LSA too closely. In [9] it is defined that an instance of an LSA can be installed only if sufficient time has elapsed since its predecessor was received. This time is defined by the MinLSArrival parameter which has a default value of 1 second.



(a) Attacker waits for a valid LSA instance

(b) Attacker triggers a valid LSA instance

Figure 5. The effects of the disguised LSA attack: shaded routers are poisoned

a way that the entire disguised LSA will be checksummed to the desirable value. Since a link entry has 64 free bits and the checksum field is only 16 bits long, we are assured that such a value exists. The value can be either directly calculated or found by an exhaustive search within a few seconds.

We emphasize that the above attack makes no assumption on the secret keys of the links on the AS since the attacker need only send LSAs on its directly attached links.

5 Effectiveness of the Attacks

In this section we show that the attacks outlined above are practical and highly effective. We first show that the attacks work against Cisco's IOS. We then show that the attacks are effective against a real-world AS topology.

5.1 Validating the Attacks

The two attacks from Section 4 were discovered by a careful security analysis of the OSPF standard [9]. As a first step we validated that the attacks work against real-world routers. To do so, we tested the attacks on the most common commercial implementation of OSPF: Cisco's IOS.

We used a network simulation software called GNS3 [2] that builds on top of a Cisco IOS emulator called Dynamips [1]. We emulated a Cisco 7200 router with a commercial image of IOS's latest stable release – 15.0(1)M. Both attacks were successful. We briefly describe the exact steps of the attacks.

Remote false adjacency. We created using the simulation software a local network attached to two routers. We chose as the victim the router that OSPF elected as the designated router. We also set up another network which is not attached directly to either one of those two routers. On that network we attached a virtual PC from which the attack was launched. All the attack packets had an arbitrary false source address that matched the subnet of the local network of the victim. We first sent a Hello message to the victim while claiming to have seen a previous Hello from the victim itself. We chose for the phantom router an ID that is larger than that of the victim. Note that the chosen phantom ID and the spoofed phantom IP address do not have to be the same. Afterwards, we sent a sequence of 10 empty database description messages with 2 seconds interval between consecutive messages. The messages have incrementing sequence numbers starting at an arbitrary value. In all messages we set the MS flag to indicate the phantom is the master of the exchange. At this point we verified that the victim has setup an adjacency with the phantom router by issuing the command 'show ip ospf neighbor' on the victim router and making sure the phantom router is shown on that list of neighbors while having the Full state. To maintain the adjacency we repeated the same Hello message we sent at the beginning of the attack every 39 seconds.

AS num.	ISP name	Num. of routers	Average degree
1221	Telstra	115	1.3
3967	Exodus	80	1.8
6461	Abovenet	145	2.6

Table 1. The list of AS topologies used for the attacks' evaluation

ISP \ # phantoms	1	2	3	4
Telstra	39%	57%	66%	70%
Exodus	47%	62%	70%	74%
Abovenet	36%	50%	60%	65%

Table 2. Percentage of black-holed routers pairs when multiple phantom routers are setup

Disguised LSA. We used the same AS topology as described above. We first predicted what will be the content of the "fight-back" LSA that will be sent by the victim and its checksum value. Using this value we found the value of the dummy LSA to insert at the end of the disguised LSA. This was done by an exhaustive search. The search took 1-2 seconds on a standard PC. We then launched the attack by locally advertising on the network of the attacker a trigger LSA and after 1 second the disguised LSA. The trigger was sent with a sequence number which is higher than the current valid instance of the LSA. The disguised LSA was sent with a sequence number that is larger than that of the trigger LSA by one. To verify the success of the attack we issued the command '`show ip ospf database`' on the routers of the AS. This command displays the contents of the LSA database of the router. For each router we checked that the disguised LSA was installed and not the "fight-back". As noted at the previous section, depending on the relative locations of the attacker and the victim some of the routers indeed installed the disguised LSA and some installed the valid LSA.

5.2 Real-World Impact of the Attacks

To evaluate the effectiveness of the new attacks on a real-world autonomous system, we simulated the attacks on actual ISP topologies, as inferred by the RocketFuel project [10, 7] (RocketFuel is an ISP topology mapping engine). We measured the attack's effect on each published AS topology. The topologies we used are listed on Table 1.

Remote false adjacency. We first evaluate the effect of the attack in which the same phantom router connects to two remote links to appear as a non-existing shortcut on the AS topology (depicted in Figure 4(b)). The most significant factors that affect the magnitude of the attack's effect are the location of the target routers to which the phantom router sets up adjacencies. These locations can be chosen by the attacker with no restriction. The effect of the attack is measured by the percentage of pairs of routers on the AS for which the shortest path between them is diverted through the phantom router. The traffic between these pairs of routers will be black-holed. To evaluate the effect of the attack we connected the phantom router to each possible pair of target routers on the AS and each case measured the number of pair of routers which were black-holed.

Table 2 shows that a single phantom router connected to wisely chosen target routers can black-hole up to 39%, 47%, and 36% of the pairs of the routers in Telstra, Exodus and Abovenet, respectively. The results differ between the ISPs due the link density of their AS topology. In ASs with lower link densities it is easier for the attacker to attract traffic since fewer alternative paths exist. Nonetheless, in all three cases the results indicate that by launching a single attack a sophisticated attacker may persistently black hole between $1/3$ to $1/2$ of the router pairs in an AS. Table 2 shows the portion of black-holed router pairs for each ISP if the attacker were to setup multiple phantom routers in strategic location on the AS. The results show that by leveraging only a small number of phantom routers an attacker can black-hole the majority of router pairs on the AS.

We also evaluated an application of the attack that black-holes traffic to a particular destination (depicted in Figure 4(a)). Here effectiveness is measured by the percentage of routers on the AS for which traffic originating from them is black-holed. Table 3 shows the number of phantom routers an attacker needs to set up in order to black-hole the traffic on the AS to specific destinations. In this experiment we consider a particular destination to be black-holed if traffic from more than 85% of routers on the AS is diverted to a phantom router. The table shows that most destinations on every AS can be black-holed (separately) by an attacker while utilizing only a single phantom router. Only 3 to 4 phantom routers are sufficient in order to persistently black-hole any destination on the AS the attacker chooses.

ISP \ # phantoms	1	2	3	4
Telstra	87%	93%	98%	100%
Exodus	49%	96%	100%	100%
Abovenet	76%	96%	100%	100%

Table 3. Percentage of destinations that can be black-holed when multiple phantom routers are setup

Disguised LSA. As noted in Section 4.2 the disguised LSA and the "fight-back" LSA are in a race, hence the attacker may not be able to poison all routers on the AS. We simulated the attack on the three ISP topologies for all possible locations of attacker-victim pairs. For each case we simulated the propagation of the trigger, disguised and "fight-back" LSAs throughout the AS and measured the percentage of routers which were poisoned, i.e., installed the disguised LSA. Table 4 summarizes the parameter values taken in the simulations. Note that the flooding and propagation times are considerably smaller than the time elapsed between the originations of the trigger LSA and the disguised LSA. This means that in some cases that the "fight-back" LSA will be triggered before the disguised LSA is originated. However, the "fight-back" LSA can not be flooded throughout the AS due to the fact that MinLSArrival interval has not elapsed since the trigger LSA was received by the routers. Hence, the neighboring routers of the victim will drop the "fight-back" LSA and the victim will have to retransmit it after 5 seconds; long after the disguised LSA is originated.

For each possible attacker location on the AS we measure the average percentage of routers the attacker succeeded to poison, i.e., install the disguised LSA. The average is calculated over all possible victim routers. Figure 6 depicts the cumulative distribution of the average percentage of poisoned routers per attacker location for all three AS topologies. It is evident from all three curves that from almost any location on the AS topology a router is able to poison more than 90% of the routers on the AS. This result demonstrates very well the power of this attack to falsify large portions of the routers' views of the AS topology, thereby effectively controlling their routing tables.

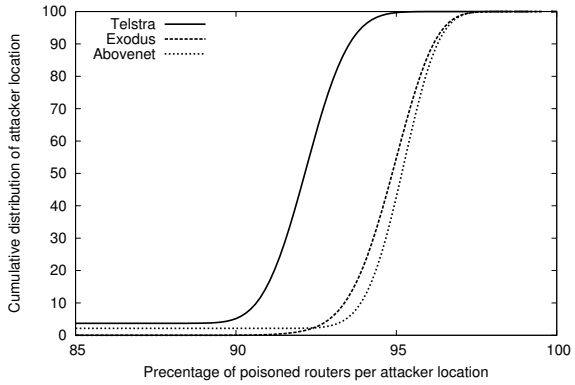


Figure 6. Cumulative distribution of the average percentage of poisoned routers per attacker location

6 Mitigation Measures

Having demonstrated the attacks, we now turn to mitigation techniques. The two attacks, remote false adjacency and disguised LSA, exploit different vulnerabilities of the OSPF spec. We briefly summarize the core vulnerabilities and then describe methods by which routers can prevent the attacks without protocol changes. We also discuss potential protocol changes that can strengthen OSPF against these and other attacks.

Disguised LSA. The attack exploits two fundamental features of the OSPF specifications:

1. a router considers two LSAs to be identical even if their contents is not, and
2. the entire contents of a future LSA is predictable.

The first is the core weakness that makes the attack possible. The second makes the attack far worse since it enables the attacker to send the disguised LSA before the next valid LSA and consequently poison more routers. As discussed in Section 4.2 the attack can be successful without the predictability vulnerability (but with much fewer poisoned routers).

Remote false adjacency. The attack exploits a common real-world operational error as well as a protocol weakness:

1. same secret key used for integrity on all links, and
2. the master can complete the adjacency protocol without seeing any of the slave's messages.

The fact that many ASs install the same secret key on all links stems from the fact that OSPF has no built-in key establishment mechanism. Operators are expected to manually install secret link keys on all routers. It is

Parameter	Value [msec]	Description
MinLSArrival	1000	The minimum time that must elapse between reception of new LSA instances at a particular router during flooding. This is the default value of the OSPF spec as well as of Cisco's IOS. This is also the time interval between the originations of the trigger LSA and the disguised LSA.
Flooding time	35	This is the time it takes from a reception of a new LSA at a router until it is flooded out. The value is based on measurements reported by Shaikh at al. [3]. We assume this is also the time it takes the victim to originate a "fight-back" once it receives the trigger LSA.
Propagation time	10	The time it takes for an OSPF message to be transmitted and propagated over a link to a neighboring router. For simplicity reasons we assume the same value applies to all links on the AS.
RxmtInterval	5000	The time between LSA retransmissions. This is the default value of the OSPF spec as well as of Cisco's IOS.

Table 4. Time values taken for the simulation of the disguised LSA

therefore not surprising that many ASs opt for a single key for all routers on the AS.

6.1 Protocol-Compliant Measures

We begin with a few mitigation measures that require no protocol changes and are fully compliant with the OSPF spec. These mitigations are intended to defeat the specific attacks described in this paper.

Disguised LSA. We do not see protocol-compliant mitigation for the core weakness. Nonetheless, we can mitigate the predictability vulnerability by adding to valid LSAs a dummy advertised link with random values. This dummy link acts as a nonce and randomizes the LSA's checksum. Routers will install the LSA with the dummy link in their databases. However, the dummy links will not effect the routing table calculation due to the bidirectional requirement. The disadvantage of this measure is that the LSA databases on all the routers will be larger.

Remote false adjacency. The attack is ineffective if the AS ensures that different links use independent secret keys for packet integrity and the secret key is known only to routers on the link. This can be difficult to enforce operationally since OSPF has no built it key management capabilities and key management must be done manually[5].

[5]We also mention that a modern design of OSPF could potentially use digital signatures rather than per-link MACs as this would greatly simplify the key management problem.

Alternatively, routers can employ anti source-IP spoofing measures on OSPF packets. Many ISPs already employ ingress filtering at their customer-facing routers. These measures could potentially be extended – at some cost – to all links on the AS. This will prevent the spoofed Hello and DBD messages from reaching the victim.

6.2 Backwards Compatible Protocol-Changing Measures

We next describe mitigation measures that update the OSPF spec. Since ASs cannot deploy protocol changes to all their routers at once, we constrain ourselves to changes that are backwards compatible. That is, an upgraded router should be able to interoperate with a current generation router. If the victim router is upgraded then it should be robust to the attacks in this paper.

Disguised LSA. To address the core weakness we propose to extend the LSA database by also storing a cryptographic hash (e.g. SHA-256) of the installed LSA. The hash is computed over the entire LSA, including the advertised links, but excluding the Age field. A router determines if two LSAs are identical by first examining the three fields – Age, Sequence Number and Checksum. If their values are different one of the LSAs is considered newer according to the current OSPF spec. If their values are the same, the router proceeds to consider their hashes. If the hashes are also equal the two LSAs are considered identical. If they are different, the LSA which was last received is considered newer. The three fields are still being compared for backwards-compatibility reasons; an LSA considered newer by the

current spec will also be considered newer while applying this.

The above change will make the disguised LSA non-persistent. The victim will consider the disguised LSA newer than the "fight-back" LSA (rather than identical) and it will originate a newer "fight-back" LSA. If the disguised LSA is sent only after the attacker receives the valid instance, then the other routers will consider the disguised LSA newer and will flood it throughout the AS eventually reaching the victim which shall originate a "fight-back" LSA.

Remote false adjacency. Clearly the master must prove to the slave that it has seen at least one message from the slave. The obvious solution in which the slave adds some random nonce to its DBD messages that would need to be echoed by the master in its DBD messages is not backwards compatible.

We propose a different solution that is backwards compatible. Once the slave receives a DBD message from the master it would send its next DBD message with probability p. The slave would send a replay of its previous DBD packet with the complementary probability. In the latter case the master must resend its last DBD message thinking it has not been received by the victim. In the former case the master must send its next DBD message. The slave can distinguish between the two responses of the master using the message's sequence number. If the slave replays its last DBD message while the master responds with its next DBD message, the slave can deduce that the master does not see its messages and stop the adjacency set up. The disadvantage of this proposal is lengthening of the exchange by an additive factor of $1-p$ messages. Another disadvantage is that the slave cannot know for sure whether the master indeed sees its messages. This is true since an attacker can correctly guess a step of the slave with probability $1-(1-p)q$, where q is the probability the attacker sends its next DBD message (rather then sending the previous one).

7 Conclusions

We presented two powerful attacks on OSPF: remote false adjacency and disguised LSA. We validated that both attacks work on widely deployed routers and demonstrated the effectiveness of the attacks on real-world AS topologies. We proposed a number of mitigation method by which routers can defend themselves against these attacks. Some of our proposed defenses require small updates to the OSPF spec.

The wide range of attacks against the OSPF protocol found by previous works and our own suggests that a rigorous security analysis using formal verification tools is needed. We leave this for future work.

References

[1] Cisco IOS emualtor. http://dynagen.org/.

[2] Graphical network simulator. http://www.gns3.net.

[3] A. S. Albert and A. Greenberg. Experience in black-box ospf measurement. In *In ACM SIGCOMM Internet Measurement Workshop (IMW)*, pages 113–125, 2001.

[4] R. Coltun and et. al. OSPF for IPv6. IETF RFC 5340, July 2008.

[5] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[6] E. Jones and O. L. Moigne. OSPF Security Vulnerabilities Analysis. Internet-Draft draft-ietf-rpsec-ospf-vuln-02, IETF, June 2006.

[7] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring link weights using end-to-end measurements. In *In ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.

[8] V. Manral and et. al. Issues with existing cryptographic protection methods for routing protocols. IETF RFC 6039, October 2010.

[9] J. Moy. OSPF version 2. IETF RFC 2328, April 1998.

[10] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with RocketFuel. In *Proceedings of the ACM SIGCOMM*, August 2002.

[11] F. Wang, B. Vetter, and S. F. Wu. Secure Routing Protocols: Theory and Practice. Technical report, North Carolina State University, May 1997.

[12] S. F. Wu and et. al. Jinao: Design and implementation of a scalable intrusion detection system for the ospf routing protocol. *ACM Transactiom on Computer Systems, VoL*, 2:251–273, 1999.