Owning the Routing Table – New OSPF Attacks

Alex Kirshon, Dima Gonikman, Dr. Gabi Nakibly

Technion, CS department
Haifa, Israel

Abstract

Open Shortest Path First (OSPF) is the most popular interior gateway routing protocol on the Internet. Most of the known OSPF attacks are based on falsifying the link state advertisement (LSA) of an attacker-controlled router. These attacks may create serious damage if the attacker-controlled router is strategically located. However, these attacks can only falsify a small portion of the routing domain's topology; hence their effect is usually limited. More powerful attacks are the ones that affect LSAs of other routers not controlled by the attacker. However, these attacks usually trigger the ``fight-back'' mechanism by the victim router which advertises a correcting LSA, making the attacks' effect non-persistent.

In this work we present new attacks that exploit design vulnerabilities in the protocol specification. These new attacks can affect the LSAs of routers not controlled by the attacker while evading ``fight-back''. These attacks afford an attacker a greater power to persistently falsify large portions of the routing domain's topology. *This allows an attacker to effectively own the routing tables of the routers in the AS without actually owning the routers themselves*. This may be utilized to induce routing loops, network cuts or longer routes in order to facilitate DoS of the routing domain or to gain access to information flows which otherwise the attacker had no access to.

The main implication of this work is the new recognition that by controlling a single router the attacker can control the entire routing domain.

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 OSPF working group in the IETF. This work study version 2 of the protocol [RFC2328] which was specifically designed for IPv4 networks, hence it is practically the only version used today. Version 3 of the protocol has been standardized to accommodate IPv6 networks, in which the fundamental mechanisms of version 2 have been kept.

The OSPF is a link-state routing protocol, this means that each router advertises its links to neighboring routers and networks. A router dynamically discovers its neighbors by executing Hello protocol, in which each router broadcasts messages on the local network. Once the neighbors have been discovered the router advertises its links to them. These advertisements are termed Link State Advertisements (LSAs). An important piece of information in an LSA is the cost of each link. The cost of a link is usually statically configured by the network administrator. The LSAs are flooded throughput the AS. A router receiving an LSA from one of its neighbors resends it to its other neighbors. In this way every router compiles a database of all the LSAs of an AS. This database is identical in all routers. Using this database a router obtains a complete view of the AS topology. This allows it to employ Dijksatra's algorithm to calculate the least cost paths between it and every other advertised network or router. From these paths a next hop router is derived for each destination. This forms the router's routing table.

In this work we present 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. All the attacks exploit design vulnerabilities in the protocol specification as defined in RFC2328. It should

be emphasized that the attacks do not rely on implementation vulnerabilities; consequently *any OSPF router may be vulnerable to these attacks*. In particular, *the attacks have been successfully test against Cisco routers* (IOS 15.0(1)M – IOS's latest stable release). The attacks allow a malicious entity to persistently subvert the routing tables of some or all the routers in the AS. This subversion allows an attacker to gain control over the routing process throughout the AS thereby freely changing the routes traversed by the data packets. The subverted routes have a global effect on the AS, since they affect all IP packets no matter what transport or application layer protocols they use. Controlling the routing process in the AS can facilitate two principal objectives. The first one is denial of service. In this objective the attacker degrades the network's ability to forward traffic with a desirable quality of service. To serve this objective the attacker can leverage the following attack vectors:

Link overload – large volume of traffic is forwarded thorough a limited capacity link. This will overwhelm the link rendering it unusable.

Long routes – traffic is routed over unnecessarily long routes. On the one hand the long routes will overload the AS by consuming more network resources. On the other hand this will inevitably increase the delay experienced by the diverted traffic.

Delivery failure – traffic is routed through a router that can not forward it to the destination. Alternatively some portion of the network mistakenly believes that it is disconnected from the destination and can not route the traffic.

Routing loops – the router's routing tables are unsynchronized in such a way that traffic is routed in loops between them never reaching its destination. In addition to the fact that this is similar in effect to a delivery failure, the looped traffic consumes large amounts of network resources before being dropped.

Churn – the forwarding of the traffic is changed very rapidly resulting in a network instability and performance degradation of congestion control mechanisms.

A second potential objective of an attacker is eavesdropping. In this objective we refer to a situation where the attacker sees traffic which otherwise it would not have access to. This allows the attacker to record or even change the traffic to facilitate impersonation or main in the middle attacks. Eavesdropping

is done by diverting the traffic through a portion of the network the attacker has control over.

In this work we assume that the attacker has an ability to send LSAs to routers in the routing domain and that routers process them as valid LSAs. This assumption is usually incorrect if the attacker is located outside the routing domain since today most routing domains filter ingress OSPF packets. Hence, in this work we assume that the attacker is an insider. Namely, the attacker has gained control over a legitimate router in the AS. This can be achieved, for example, by conspiring with an authorized personnel having physical access to the router or by remotely exploiting an implementation vulnerability to achieve code execution on the router. Several such vulnerabilities have been published in the past. This allows the attacker to send OSPF packets that will be accepted and processed by other OSPF routers in the attacked AS.

In this work we make the following assumption on the attacker's capabilities:

1. **Location** – as mentioned above, we assume the attacker is located with the boundaries of the AS while having control over a legitimate router. Other than that we do not assume anything about the attacker location within the AS or the role the router has in the OSPF process (e.g., AS border router).
2. **Resources** – the attacker has bandwidth, processing and memory resources which are comparable to an average router in the AS. In particular, the attacker can not process or originate traffic in a higher rate than most other routers in the AS.
3. **Acts alone** – the attacker has only a single foothold in the AS. It does not spread throughout the AS and take over other routers. In addition, it does not collaborate with other attackers in the AS. All other routers in the AS besides the attacker are legitimate innocent routers.

There are a few past works that presented attacks that exploit design vulnerabilities of the OSPF protocol. Most of these attacks fall under one of the following attack vectors:

1. **False self LSAs** – in this attack vector the attacker send LSAs on behalf of the router it has control over. These LSAs contain false information. The attacker may falsely advertise it is connected to certain stub networks. It may also falsify the costs of real or false links to neighbors. This vector of attacks is simple and can be easily executed. However, they have limited effectiveness since the attacker can only falsify a small piece of the AS topology – its immediate neighborhood.
2. **False Hello** – in this attack vector the attacker send false Hello messages on the networks it is attached to. Using these messages the attacker can make other routers on the network believe there are link to new neighbors or existing neighbors are disconnected. Attacks in this vector have only local effect since they can only affect the routers in the local network.
3. **False phantom LSA** – in this attack vector the attacker send LSAs on behalf of a phantom router that does not really exist in the AS. In this way the attacker can have a more global effect by influencing on a large portion of the AS topology; however these false LSAs have no direct impact on the routing tables of the routers. This is because the OSPF protocol expects each link to be advertised by both its ends. Since no other router advertises a link to the phantom router, its entire advertised links are ignored.
4. **False peer LSA** – in this attack vector the attacker send LSAs on behalf of an exiting victim router in the AS which is not itself. Using this technique the attacker can have a global effect by influencing a large potion of the AS topology. It can also affect the router's routing tables since other routers advertise links to the victim router. The main drawback of this attack vector is that its effect is not persistent. The false LSA in flooded throughout the AS by other routers in the AS, therefore the victim router will also get the false LSA. Once the victim router receives the false LSA it immediately issues a correcting LSA that overrides the false one – the fight back mechanism. This reverts the effect of the attack. The attacker must again issue a false LSA. This

increases the exposure of the attacker and makes it more prone for detection.

In this work we propose novel attacks that exploit new found design vulnerabilities in the OSPF specification. As opposed to the above attack vectors, the attacks presented in this paper can persistently subvert the routing tables of the routers in the AS, while being able to have a global effect on the AS, namely falsify potions of the AS topology that are not necessarily attached to the attacking router.

Related Work

There are only a handful of works that analyze the security of the OSPF. Ref. [Wang97] discusses an attack in which an area internal router impersonates as an AS border router and advertises AS external LSAs. This can be done since there is no mechanism in the OSPF by which a router can authenticate the role other routers assume. The power of this attack is that the AS external LSAs are flooded throughput the AS (except stub areas) as opposed to other types of LSAs which are confined to a single area in which they were advertised. An attacker can take advantage of this attack and advertise links to destinations external to the LSA (IP address of Google or Facebook, for example). The advertisement can include very low cost to the destination or a longer subnet address. 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 make take a longer route. This attack has the disadvantage that it can not influence destination which are internal to the AS. A router will always prefer an AS internal router than an external one.

Ref. [Wu99] describes several attacks in which the attacker sends a false LSA on behalf of another router in the AS. All the attack variants described in [Wu99] trigger a fight-back by the victim router, make the attack effect non-persistent and force the attacker to re-launch the attack. On one hand, this can be leveraged by the attacker to make the routing process in the AS instable, but on the hand it dramatically increases the exposure of the attacker and the chances the AS administrator discovering its location.

Ref. [Jones06] surveys all the different attack vectors on OSPF. It also introduces a few novel attacks. One attack disables the fight-back mechanism by periodically injecting the false LSA (1 packet per 5 seconds). This disables the fight-back since the OSPF standard does not allow a router to send two instances of the same LSA within the time period MinLSInterval (a protocol parameters that defaults to 5 seconds). Since the standard also states that the fight-back is triggered only after the router has already processed and flooded the false LSA. This means that by receiving a false LSA every 5 seconds the victim router is unable to send a fight-back LSA. The effect of this attack is persistent, but with a relatively high cost: the attacker must flood its false LSA at a relatively high rate.

Another attack introduced in [Jones06] in which the attacker may send false Hello messages thereby changing the designated router elected in the attackers LAN or making other routers in the LAN reset their adjacency with the designated router. In both cases the routers in the LAN must re-establish their adjacencies; a process that may take tens of seconds. During this time the LAN is advertised by the router as a stub network through which no packet may be routed towards other networks in the AS. This can cause other routers in the AS to repeatedly recalculate their routing tables.

Another class of attacks discussed in [Jones06] is denial of service attacks. In this type of attacks the attacker floods the victim router while consuming its resources. This may overwhelm the victim router rendering it unable to function properly. In one attack the attacker originates large number of Hello packets destined to the victim router each with a different spoofed IP source address. Each such Hello packet makes the victim create a new entry in the Neighbors list. By overflowing this list the attacker can make sure that the victim is unable to process Hello packets from new neighbors on the LAN. In another attack the attacker overwhelms the victim with bogus LSAs. Each LSA must be saved in the LSA database until it expires (which takes 1 hour). By overflowing this database the attacker can make sure that the victim is unable to process new LSAs, thereby seriously affecting the victim's ability to adapt its routing table to changes in the AS topology.

Yet another novel attack introduced in [Jones06] is an attack in which the attacker impersonates as a AS border router and originates an AS-external

LSA of a particular popular network outside the AS in which it states that packets to this destination network must be routed though a router in a stub area (using the Forward field in the LSA). Since the AS-external LSAs are not flooded inside stub areas this causes a routing loop: routers outside the stub area will route the packets towards the stub area (according to the false LSA) while routers inside that area will route it outside the area.

The New Attacks

Disguised LSA

According to RFC 2328 Sec. 13.1 two instances of a LSA are considered identical if they have:

1) The same sequence number,

2) The same checksum value, and

3) Approximately the same age (within a 15 minutes time difference).

This is true even if the actual content of the LSAs is different!

The attacker can exploit this vulnerability by advertising an LSA with the same three fields (sequence, checksum and age) as a valid LSA being advertised by the victim router. This has the benefit that even if the victim receives the spoofed LSA a fight back is not triggered since the LSA is disguised and considered to be the same copy as the valid LSA (again, even if their contents are very different) and therefore ignores it.

However, all other routers in the AS will also consider the false LSA as a duplicate and therefore, they will not install the LSA in their LSA DB. To fix this the attacker shall disguise the LSA to the next valid instance of the LSA that the victim is expected to originate. As the attacker sends this disguised LSA it triggers the victim to originate this next valid instance of the LSA. The trigger is simply done by leveraging the fight-back mechanism. Namely, the attacker sends out a false LSA to the victim who fights back by sending out the next valid instance of the LSA. The following figure illustrates the basics of the attack:

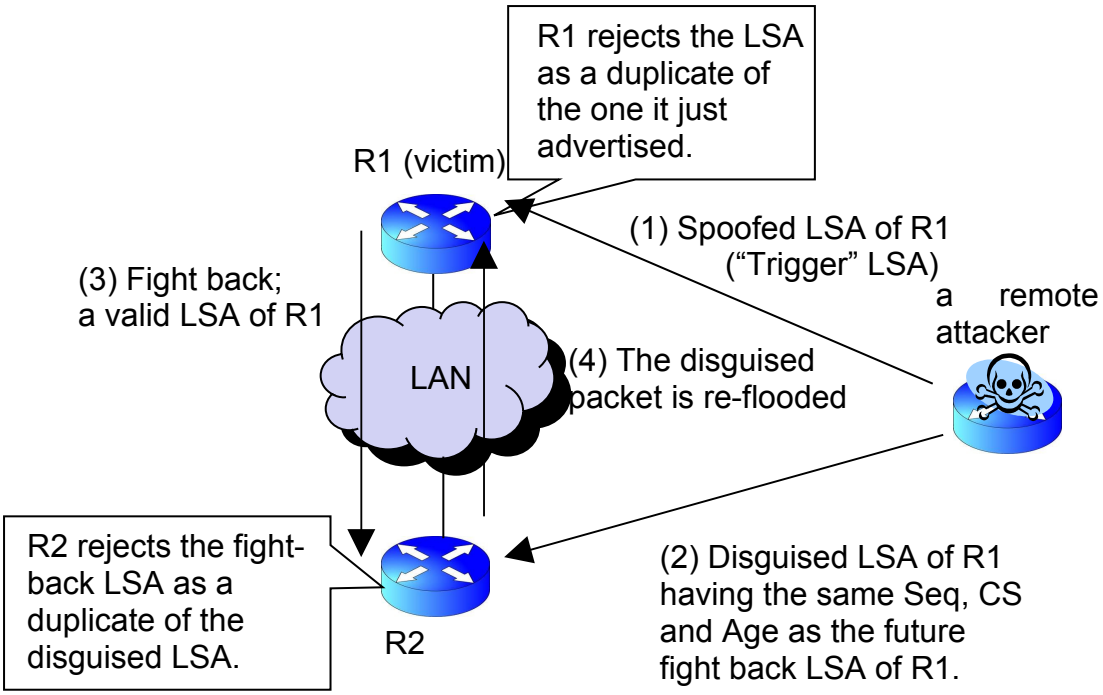

Figure 1 - Illustration of the disguised LSA

(1) The attack begins by sending a spoofed LSA of R1 to R1 itself. This will certainly trigger a fight back. Let's call this packet the "Trigger".

(2) At the same time the attacker sends a disguised LSA of R1 to R2. The disguised LSA is a specially crafted packet having the same sequence, checksum and age (+/- 15 minutes) as the future fight back LSA of R1. We later shall discuss how one can predict these three fields.

(3) R1 sends the fight back LSA. This will be received by R2, but having the same three fields as the disguised LSA the fight back will be viewed by R2 as the same copy of the LSA it just received. Hence it will not update its LSA DB and it will not re-flood the packet.

(4) R2 re-floods the disguised LSA. This will be received by R1, but having the same three fields as the fight back LSA this LSA will be viewed by R1 as the same copy of the LSA it just sent. Hence it will not update its LSA DB and it will not re-flood the packet or trigger another fight back.

After this sequence of packets R1 and R2 have in their LSA DB two different copies of the LSA of R1. This state is persistent. The routers will again be synchronized only after R1 will advertise its next LSA instance after 30 minutes (the default LSA interval).

The three fields of the disguised LSA are determined as follows. All the content of the future valid (fight back) LSA is deterministic and predetermined. This also includes the three values of Sequence, Age and Checksum. Setting the Age and Sequence values of the disguised packet is straightforward. The age should be '0' while the difference between the time the disguised packet is advertised and the time the fight back LSA is originated must be less than 15 minutes. The sequence of the disguised packet should be greater by 1 than that of the trigger packet. The checksum is a bit trickier. We can add to the disguised LSA a dummy Link entry which its fields' values will be chosen in such a way that the checksum value of the disguised LSA will be the same as the checksum value of the fight-back LSA. Since the LSA checksum is a linear combination of the all the LSA fields the fields of the dummy Link entry can be easily calculated. We are assured that such values exist with high probability since the length of the checksum is only 16 bits, while there ar 88 bits in the dummy Link entry that can be determined arbitrarily (#TOS, metric, Link ID, and Link Data). The exact value of the dummy Link entry does not matter to the attack itself. Since this link will not be bidirectional (i.e. another router will not advertise the opposite direction of that link), it will not be considered for the routing table calculation anyway.

Note that the above illustration of the attack necessitates the attacker to know the MD5 key of the links attached to the victim router. Another powerful and potential use of attack is to consecutively advertise the trigger and disguised LSA on the local LAN of the attacker rather then unicasting the packets to the victim router and his neighbors. From there the two packets are flooded throughout the AS while the routers install the disguised LSA in their DBs. As the trigger arrives at the victim router it advertises fight back LSA. The fight back LSA will be flooded to the victim's neighbors, but if these neighbors already received and installed the disguised LSA the fight back LSA will be rejected as duplicated by those neighbors and will not be re-flooded to the next neighbors. This means that we have a race between the fight back LSA and the disguised LSA. The one that arrives first to a router gets installed and the other is rejected as a duplicate. Since the disguised LSA is sent well before the fight back LSA the former has a much greater chance to "conquer" greater portions of the AS. Here is a typical map of an AS after this attack

variant is launched. The red parts indicate that the routers within them installed the disguised LSA and the blue parts indicate the locations of routers which the fight-back LSA has reached first.

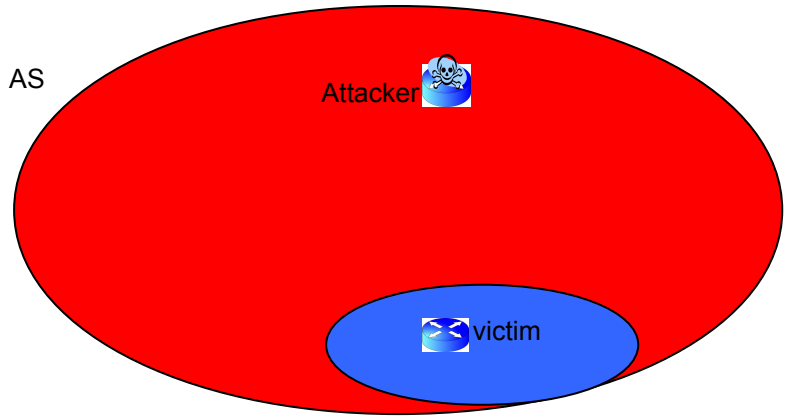

Figure 2 - AS map after the attack has been launched

As one can see, this attack is an effective tool to persistently falsify an LSA of a router not controlled by the attacker. The attacker can achieve a situation where all/most of the routers in the routing domain has a false LSA of the victim router. The attacker can repeat it for different victim routers to fully control the topology viewed by the routers in the AS and consequently their routing tables.

Remote False Adjacency

The vulnerability this attack exploit is documented in [RFC 2328 Sec. 10.8]. This section describes the procedure for sending database description packets during the adjacency setup process. A review of this section reveals that a master router can successfully complete the adjacency setup without actually seeing the messages sent by its peer – the slave router. This means that an attacker can remotely setup an adjacency with a victim router as long as the victim router plays the slave in the setup exchange. Since a neighbor of the victim must have an IP address that belongs to the subnet ID of victim's link. The attacker must impersonate as a phantom (non-existing) router in the victim's link. The victim setup an adjacency with this phantom router.

After the attack is launched and the victim router has an adjacency with the phantom router, and the victim advertises a link to the phantom router! This is

a pivotal point in the attack and its main benefit. If the attacker advertises false LSA on behalf of the phantom router that links it to the victim router, the advertised link to the phantom router would make the link between the phantom router and the LAN **bidirectional**. This means that all the other routers in the routing domain will take this link and the LSAs advertised on behalf of the phantom router into account while calculating their routing tables. This is the first attack ever to successfully create a persistent bidirectional link between a real router and a phantom one thereby making the LSAs of the phantom router be considered in the routing table calculation by all the routers in the routing domain. The attacker can now advertise arbitrary LSA on behalf of the phantom router. These LSAs will affect the routing tables of all the routers in the AS.

To successfully complete this attack the attacker must know the following pieces of information:

1. The MD5 key of the remote LAN. In most cases this is the same shared secret for all LANs in the AS.
2. The configuration parameters of the remote LAN, e.g., HelloInterval, RouterDeadInterval, etc… In most cases these are the same parameters for all LANs in the AS.

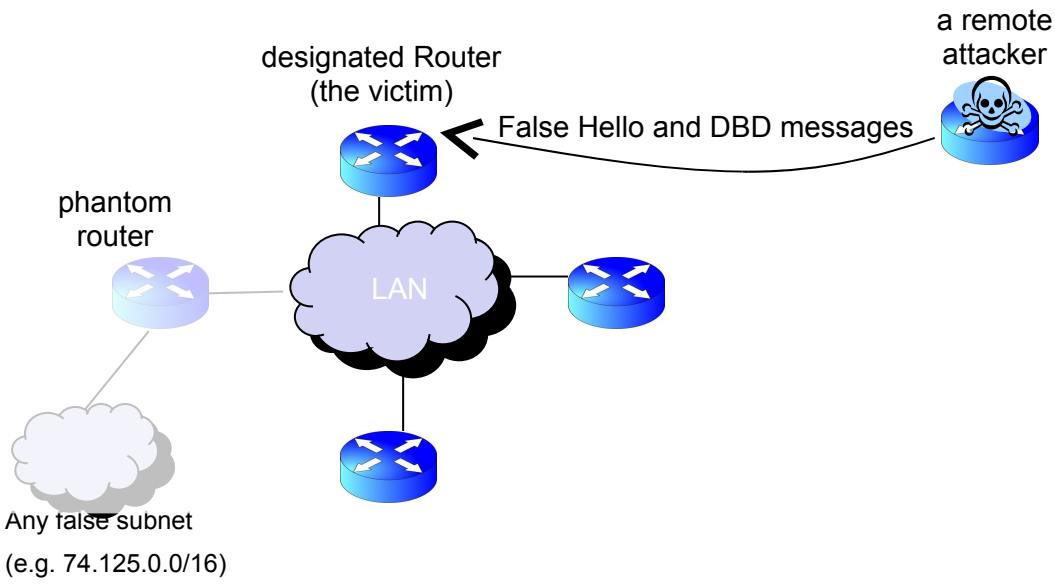

Figure 3 - Illustration of the remote adjacency attack

A potential use of the attack is to black-hole traffic to a specific subnet by having the phantom advertise the subnet IP to be black-holed. Since the attacker can create phantom router anywhere it wishes on the AS it can essentially black hole all the traffic that originate anywhere in the network which is destined to this subnet. See the following illustration.

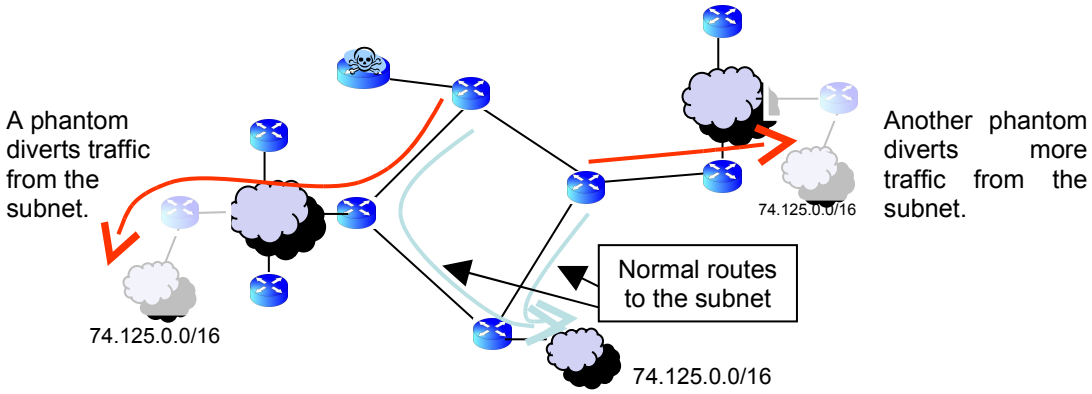

Figure 4 - Black-hole the entire taffic in the AS destined to a specific subnet

Another potential use of the attack is to locate the phantom router in a strategic "location" that allows it to create a desirable shortcut for large volumes of traffic in the AS. For example, the phantom router can be made to link to two distant networks in the AS as shown in the following illustration. This can be done by targeting two victim routers in the two networks. The two victim routers should be the designated routers of their respective neighbors.

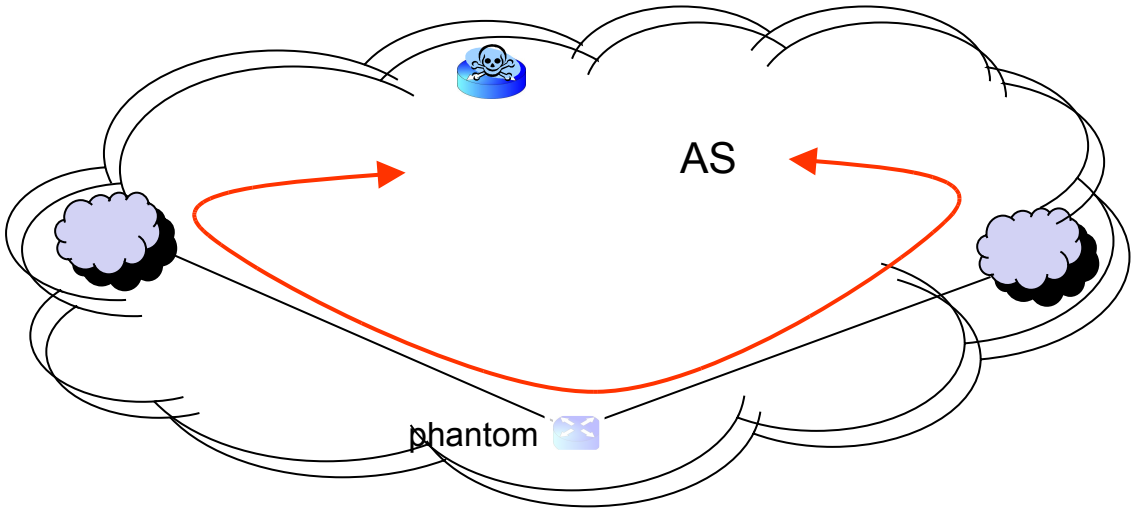

Figure 5 - The phantom router is a desirable shortcut for most of the AS traffic

The detailed sequence of the adjacency bring-up is as follows. All the packets sent by the attacker have a spoofed IP source that equals to the false IP address of the phantom. This address must be part of the local subnet of the victim's network.

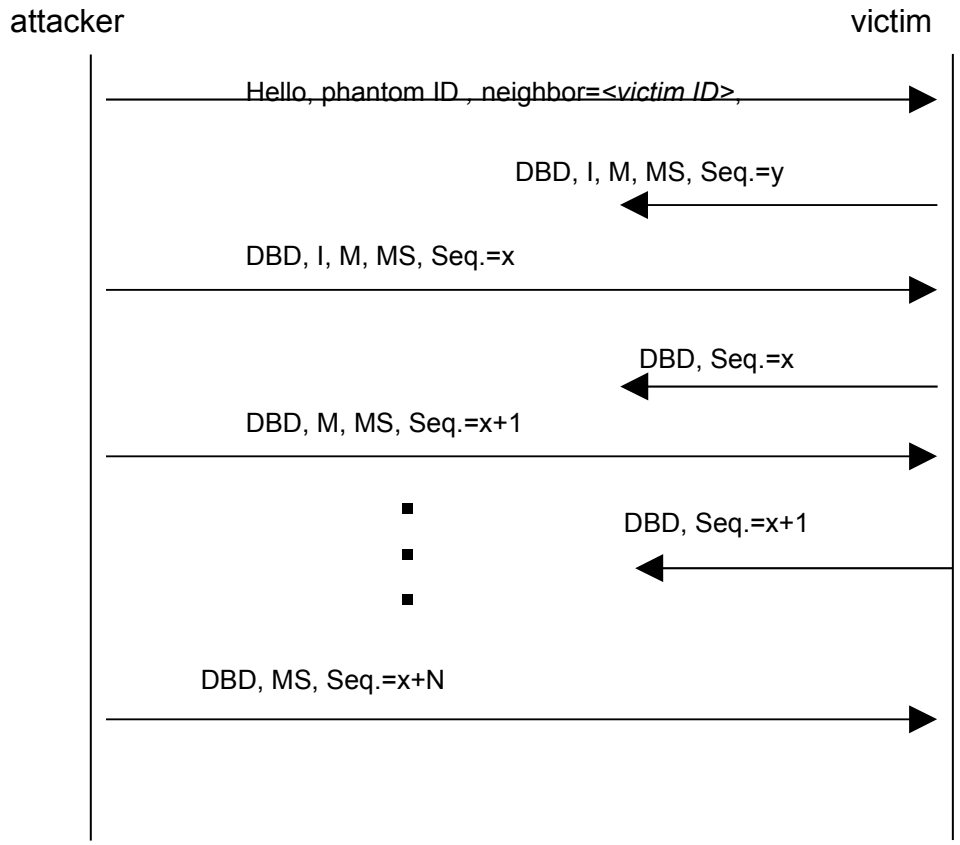

Figure 6 - The sequence of the remote fase adjacency attack

The attack starts by sending Hello packet to the victim router. Since the Hello contains in it’s neighbor a list the victim’s ID the victim enters the 2-way state. The victim is assumed to be a designated router (DR), hence it starts setting up an adjacency with the phantom and enters the ExStart state. The victim then sends a DB description (DBD) packet with his own sequence, y. This packet, as all the packets the victim sends, is not received by the attacker since it is destined to the spoofed IP address of phantom router in the victim’s subnet.

The attacker then sends its first DBD. The timing of the packet is not that important, the victim retransmits his first DBD several times every few seconds (5 secs by default). The first DBD of the attacker (actually the phantom) sets the Initialize (I), More (M), and Master (MS) bits and sets the

sequence number to an arbitrary value (x). Since the packet is crafted in such a way that the ID of the phantom is larger than the ID of the victim the victim concedes to be a slave and the phantom gets to be the master. This means that the victim adopts the sequence number of the phantom (x) and sends his next DBD only after he gets a DBD from the phantom.

The attacker then proceeds to repeatedly send DBDs with increasing sequence values. All the phantom's DBDs have no LSAs as if the LSA DB of the phantom is empty. The attacker keeps sending these DBDs to let the victim send all the LSAs in his LSA DB. The attacker doesn't really know how many DBD messages the victim will need to send all his DB content, however he can easily bound this number (10 BDBs are usually enough). In the example this bound is N. It does not matter if N is an overshoot; the victim will keep sending empty DBDs when he has no LSAs to send.

After the attacker (phantom) sends his last DBD (we assume that by now the victim also finished sending his own DB) the victim skips the Loading state since the phantom has no new LSAs, and then the victim enters the Full state. At this point the victim is fully adjacent with the phantom and updates the Network LSA of its network accordingly. Success!

The attack has a few caveats:

1. The adjacency must be continuously maintained by sending a Hello message every RouterDeadInterval. By default this value is 40 seconds.
2. Following the adjacency setup the victim floods LSAs to the phantom and expects to receive Acks from it. According to the OSPF spec if adjacent router does not respond with an Ack the victim will just retransmit the LSAs over and over endlessly. Nonetheless, we observed that a Cisco router gives up after 125 seconds and then tears down the adjacency.

The last caveat means that for Cisco routers the attack should be re-launched once every 125 seconds. However, it should be noted that if the attacker and the victim router are located in the same area, the attacker, in principal, knows each LSA the victim floods. This means that it can spoof the Ack messages

as well (it has a time window of over 120 seconds to respond with an Ack). However, we have not tested this idea in practice.

Conclusions

The two attacks are based on analysis of the OSPF specification [RFC 2328]. The attacks are successful against Cisco IOS 15.0(1)M (on a 7200-series router). This means that the vulnerabilities we described for each attack is indeed implemented as expected in Cisco IOS. The Scapy attack scripts we used are available on demand.

We believe the vulnerabilities and the attack we described in this paper is ground breaking. Up until now the common wisdom was that even if the attacker is an insider it can not persistently falsify the LSA of a router it does not control. Our work shatters this misconception. The main implication of the new attacks is that **one can control the entire routing domain from a single router.**

References

[RFC2328] J. Moy, "OSPF Version 2", IETF RFC 2328, April 1998.

[Wang97] F. Wang et. al., "Secure routing protocols: theory and practice", Technical Report, North Carolina State University, May 1997.

[Wu99] S. Wu et. al., "JiNao: Design and implementation of a scalable intrusion detection system" for the OSPF routing protocol", Journal of Computer Network and ISDN systems, 1999

[Jones06] E. Jones et. al.. "OSPF Security Vulnerability analysis", IETF draft-ietf-rpsec-ospf-vuln-02, June 2006.