Owning the Routing Table

Part II

Gabi Nakibly[1], Eitan Menahem[2], Ariel Waizel[2], Yuval Elovici[2]

[1]National EW Research & Simulation Center,
Rafael – Advanced Defense Systems Ltd.
[2]Telekom Innovation Laboratories, Ben Gurion University

Abstract

Open Shortest Path First (OSPF) is the most popular interior gateway routing protocol on the Internet. Most known OSPF attacks are based on falsifying link state advertisements (LSA) of an attacker-controlled router. These attacks may create serious damage if the attacker-controlled router is strategically located in the autonomous system (AS) topology. 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 – the router on behalf of which the attacker advertises the false LSA – which advertises a correcting LSA, making the attacks' effect non-persistent.

At Black Hat USA 2011 [BH11] and NDSS 2012 [NDSS12] we presented the first known attacks that allow an attacker to persistently falsify an LSA on behalf of a router it does not control, while evading the "fight-back" mechanism. These attacks allow to persistently poison the routing domain with false topology information.

As a sequel to that work we now push the envelope further and present an even more powerful OSPF attack that exploit a newly discovered ambiguity of the OSPF standard [RFC2328]. As the attack is launched against a victim Cisco router not only that victim does not fight back but its routing table is completely erased, effectively excluding it from the routing domain.

The new attack allows an attacker that owns just a single router within an AS to effectively own the routing tables of ALL the routers in that 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 contribution of this work is the recognition that by controlling a single router inside the AS 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 a new powerful attack that exploits a newly discovered ambiguity in the OSPF standard. This means that some OSPF implementations will be vulnerable to attack and others will not. This depends on the choices the implementer made. *The attack has been successfully tested against Cisco routers* (IOS version 15.0(1)M[1]). We note that Cisco holds about 75% of the global enterprise router market [Infonetics12]. Potentially many other implementations of OSPF may be vulnerable due to this ambiguity.

The attack allows a malicious entity that already controls a just a single router within the AS to persistently subvert the routing tables of all the routers in that 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 cannot forward it to the destination. Alternatively some portion of the network mistakenly believes that it is disconnected from the destination and cannot route the traffic.

[1] IOS's latest stable release we can get our hands on.

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-controlled router 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.

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 assumptions on the attacker's capabilities:

1. **Location** – as mentioned above, we assume the attacker is located within the boundaries of the AS while having control over a legitimate router. Other than that we assume nothing about the attacker location within the AS or the role the attacker-controlled 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 cannot 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.

Related Work

There are a few past works that presented attacks that exploit design vulnerabilities of the OSPF protocol. As we will next see, most of these attacks fall under one of the following attack vectors:

1. **False self LSAs** – in this attack vector the attacker sends LSAs only 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, it has 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 sends 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 links 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 of the attacker.
3. **False phantom LSA** – in this attack vector the attacker sends LSAs on behalf of a phantom router that does not really exist in the AS. 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 the opposite direction of the links of the phantom router, these links will be ignored by all routers during the routing table calculation.
4. **False peer LSA** – in this attack vector the attacker send LSAs on behalf of an existing victim router in the AS which it does not control. Using this technique the attacker can falsify arbitrary LSAs in the AS thereby influencing a large portion of the AS topology. The main

drawback of this attack vector is that its effect is not persistent. The false LSA is flooded throughout the AS by other routers in the AS, therefore the victim router will eventually receive the false LSA advertised on its behalf. Once the victim router receives the false LSA it immediately issues a correcting LSA that overrides the false one – this is called the "fight-back" mechanism. This fight-back LSA reverts the effect of the attack. The attacker then must issue again a false LSA. This increases the exposure of the attacker and makes it more prone to detection.

In this work we propose novel attack that exploits a design ambiguity in the OSPF specification. As opposed to the above attack vectors, the attack 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.

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 AS. 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 take a longer route. This attack has the disadvantage that it cannot 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, making the attack effect non-

persistent and forcing 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 other hand it dramatically increases the exposure of the attacker and the chances of the AS administrator to discover 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 (one packet every five 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 iteratively send the false LSA.

Another attack introduced in [Jones06] is one 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 through 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 attacks we presented in [BH11] and [NDSS12] were the first to persistently and stealthily falsify an LSA on behalf of a router the attacker does not control, while evading the "fight-back" mechanism. The most powerful attack we presented in that work was called "Disguised LSA". It exploited a vulnerability of the OSPF standard which allows two LSAs to be considered identical even if their actual payloads are different. This vulnerability allowed the attacker to send a false LSA which is considered identical to the fight-back LSA of the victim router. Consequently, the fight-back LSA is rejected as duplicate by the routers and it does not override the false LSA advertised by the attacker. The attack's major drawbacks were:

1) Most of the AS routers are poisoned but not all of them.
2) The attacker was required sending two false LSAs.
3) No effect on the routing table of the victim router's routing table is achieved.

The new attack we next present does not suffer from these drawbacks.

The New Attack

An LSA has a header of the following format:

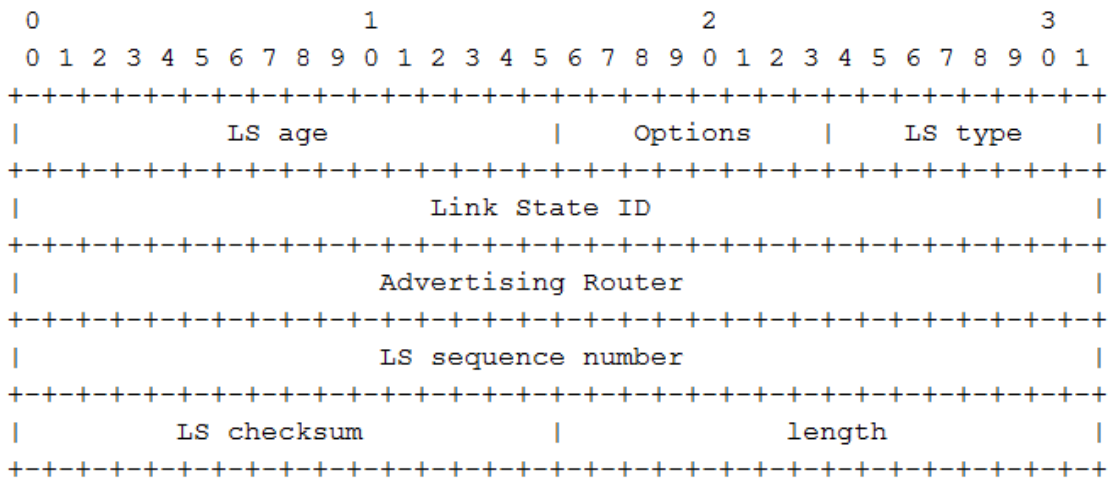

Figure 1 - LSA header format

- *LS age* – The time in seconds since the LSA was originated
- *Options* – Supported optional capabilities
- *LS type* – The type of the LSA (e.g. Router, Network, Summary,…). In the following we shall focus solely on Router LSA type.
- *Link State ID* – Identifies the portion of the AS topology that is being described by the LSA.
- *Advertising Router* – The Router ID of the router that originated the LSA.
- *LS sequence number* – Successive instances of an LSA are given successive LS sequence numbers.
- *LS checksum* – The Fletcher checksum of the complete contents of the LSA
- *Length* – The length in bytes of the LSA.

Let's take a closer look at two of the above fields and their values when it comes to Router LSAs:

- *Link State ID* – This field identifies the router that the links of which are listed in the LSA. The field equals to the router ID of that router.
- *Advertising Router* – This field identifies the router that initially advertised (originated) the LSA.

The OSPF spec dictates that only a router itself can originate its own LSA (i.e. no router is expected to originate a LSA on behalf of other routers), therefore in Router LSAs the two fields – 'Link State ID' and 'Advertising Router' – must have the exact same value. However, the OSPF spec does not specify a check to verify this equality on Router LSA reception. This enables one to send a Router LSA with different values in these two fields. In the following we shall see why this may desirable for an attacker.

According to Section 13.4 of the OSPF spec a router will fight back only if it receives a false LSA in which

> "the Advertising Router is equal to the router's own Router ID"

This means that no fight back shall be triggered by the victim router as long as the field 'Advertising Router' of a false LSA is NOT equal to the victim router's ID. This is true even if the 'Link State ID' of that LSA is equal to the victim router's ID. Namely, no fight back is triggered even if the false LSA claims to describe the links of the victim router.

Hence, the attack goes as follows. Assuming the attacker wishes to advertise a Router LSA on behalf of some victim router, R_v. It should originate an LSA for which:

- Link State ID = ID of router R_v.
- Advertising Router = any value other than the ID of router R_v.

The OSPF spec guarantees that this false LSA will not trigger fight back by R_v and all routers in the AS – including R_v – will install this false LSA in their LSA DBs.

But there should be a problem with this attack. Section 12.1 of the OSPF spec determines that a LSA is uniquely identified by the combination of the following three fields:

- LS type (this field always equals '1' for Router LSAs)
- Advertising Router
- Link State ID

Therefore, the false LSA should NOT replace the valid LSA in the LSA DBs, since those two LSAs have different identifiers (different Advertising Routers).

This means that the valid LSA is not guaranteed to be erased from the LSA DB.

Now let us turn to the ambiguity in the spec that allows the attack to succeed. According to Section 16.1 of the OSPF spec during the routing table calculation LSAs are looked up in the LSA DB while:

> "This is a lookup … based on the Vertex ID."

Here by Vertex ID the OSPF spec means the Link State ID field. This means that while a router calculates its routing table it identifies LSAs based on their Link State ID field only.

This creates an ambiguity in OSPF spec. On one hand an LSA is identified by the combination of the three fields mentioned above. On the other hand while the routing table is calculated the lookup identifier of an LSA is composed of the Link State ID field only.

This ambiguity raises the following question: which LSA will be fetched from the LSA DB during the routing table calculation – the valid LSA of the victim router or the false one advertised by the attacker? Remember, both LSAs reside side by side in the LSA DB of every router in the AS. Both LSAs have the exact same value in their Link State ID field – the Router ID of the victim router.

The OSPF spec fails to answer this question. Hence, the answer must be implementation dependent. An OSPF implementation that fetches the valid LSA during the routing table calculation is oblivious to the attack. However, an OSPF implementation that fetches the false LSA is completely vulnerable to the attack.

Evaluation of Cisco

We now turn to the most common OSPF implementation in the world: Cisco's IOS. According to a recent study [Infonetics12] Cisco holds about 75% of the global enterprise router market. To evaluate Cisco's OSPF implementation we used GNS3 emulation software with a production IOS image. The latest IOS version we got our hands on is 15.0(1)M[2]. The Scapy attack script in included in the Appendix.

[2] As of this writing, the latest stable release of IOS is 15.2(4)M2.

Our evaluation reveals that Cisco's OSPF implementation is vulnerable to the attack. Here are our main findings:

- **The false LSA replaces the valid LSA** – If the false LSA is advertised with a sequence number that is higher than the sequence number of the current valid LSA, the false LSA is not only installed in the LSA DBs of all the routers but also replaces the valid LSA in the LSA DBs. This happens in all routers in the AS including the victim router. Consequently, all routers in the AS (including the victim router) consider the false LSA during their routing table calculation. The next figure contains a screen capture of the LSA DB of the victim illustrating this.

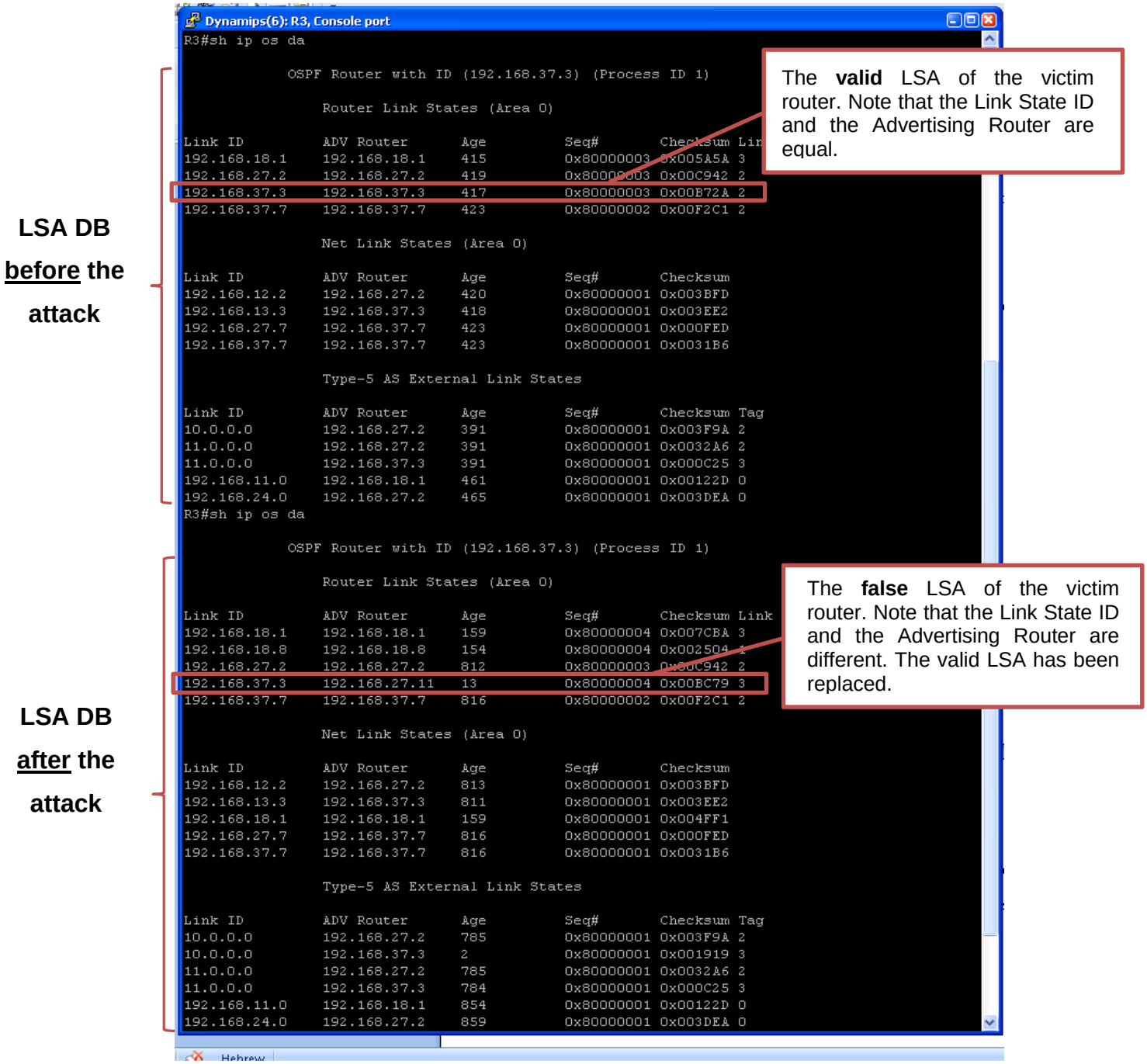

Figure 2 - The victim's LSA DB before and after the attack

- **Routing tables of all routers except the victim are poisoned** – The routing tables of all the routers except the victim router build their routing tables exactly according to the false links described in the false LSA advertised by the attacker.
- **Routing table of the victim router is erased** – The victim router does not have in its DB a LSA with an Advertising Router field that equals the victim's Router ID (recall that the valid LSA was replaced by the false LSA having a different Advertising Router value). In Cisco's OSPF

implementation this leads to a situation that the routing table calculation process does not find paths to any other router or network. Consequently, all the entries in the victim's routing table which are sourced from the OSPF process are deleted. This essentially empties the routing table. This means that, unless the victim router has been preconfigured with a static default route, it will drop all incoming IP packets unless they are destined to a router or network immediately connected to the victim router.

This erasure of the victim's routing table is permanent. Unless the attacker decides to "undo" this erasure (see next paragraph) the victim's routing table does not spontaneously recover from this. The OSPF process must be reinitialized by the administrator.

- **Undoing the attack** – If the attacker wishes, it can easily undo the effects of the attack by sending another false LSA but this time with an Advertising Router that equals the victim's Router ID. The 'undo' LSA must have a higher sequence number than that of the attack LSA that was previously sent. This will trigger a fight back by the victim which will originate a newer instance of the valid LSA which shall replace the false LSA in all the LSA DBs of all routers in the AS.

Attack Applications Examples

In the following we review a couple of the potential applications of the attack. We note that the attacker can be anywhere in the AS to successfully launch the attack.

1) **Black hole** – in this application the attacker aims to disconnect all the routers and networks of the AS from some destination network outside the AS. This will be achieved by making one of the routers in the AS a black hole for that destination.

   The attacker shall originate a false LSA which announces that the victim router is directly connected to some given destination network, let's call it net-X, that actually resides outside the AS (e.g. the IP range of google.com). Since an intra-AS router will always take precedence over an inter-AS route all router in the AS will recalculate their routing tables such that all traffic destined to net-X will be routed to the victim

router. Since the routing table of the victim router is erased following the attack the victim will not have a routing entry associated with net-X and it will drop all packets making it a black hole.

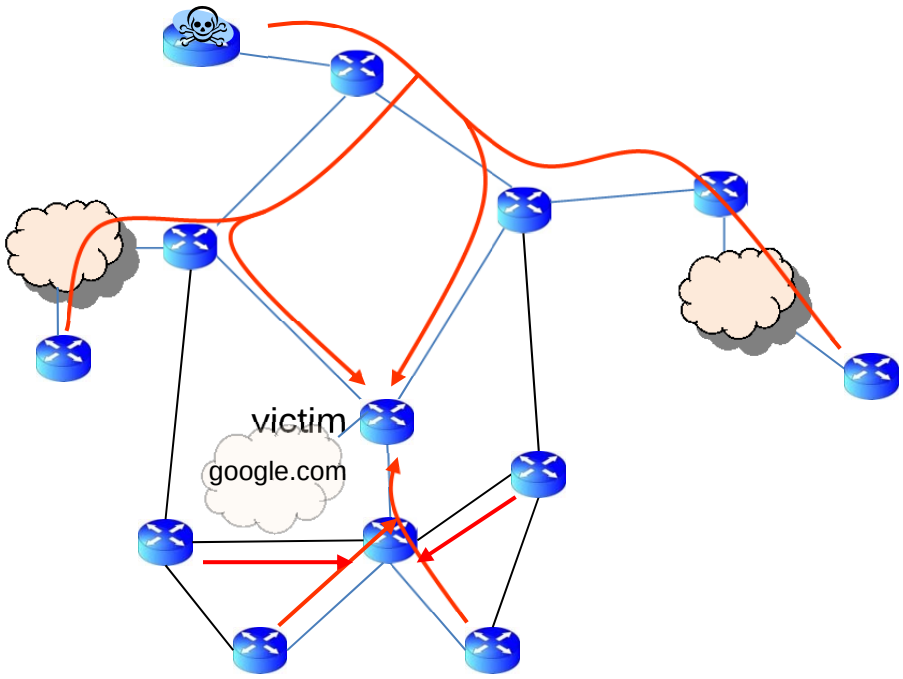

Figure 3 - Black hole of goole.com

2) **Traffic diversion** – in this application the attacker aims to divert traffic though alternative paths in the AS. This may facilitate for example a man-in-the-middle attack in which the traffic is diverted through the attacker.

   The attacker shall originate a false LSA which announces that the victim router has no links to other routers or networks in the AS. This will effectively cut off the victim from the routing process. All routers in the AS will calculate new routing tables as if the victim router has been removed from the AS. Note that this will happen despite the fact that the victim's neighboring routers continue to advertise their links to the victim[3].

   The end result is that all AS traffic shall circumvent the victim router taking alternative routes if they exist. If they don't exist, the AS is

[3] According the OSPF spec every link must be advertised by both its peers in order to take it into consideration during the routing table calculation.

partitioned. In the following figure the green line denotes the path before the attack while the red one denotes the path after the attack. In this scenario the attacker now have access to more traffic than it normally had.

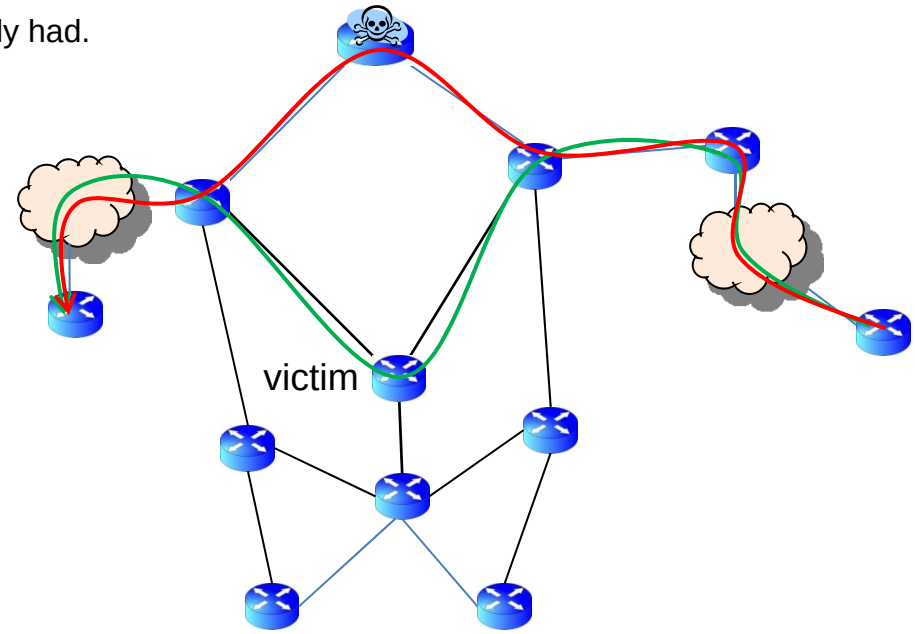

Figure 4 – Traffic diversion. Green path – before the attack, Red path – after the atatck

Conclusions

The new attack is based on analysis of the OSPF specification [RFC 2328] and revealed a newly discovered ambiguity in the spec. This ambiguity may expose many OSPF implementations to this attack. In particular, the attack is successful against Cisco routers.

As the attacks we presented at Black Hat USA ’11 [BH11] the attack we described here is novel. Up until now the common wisdom was that even if the attacker is an insider it cannot persistently falsify an LSA of a router it does not control, much less make it consider the false LSA as its own. Our work shatters this misconception. Not only that the victim router does not fight back but, in the Cisco case, its routing table is erased by the attacker effectively excluding it from the routing domain. The main implication of the new attacks is that **one can control the entire routing domain from a single router.**

References

[BH11] G. Nakibly, A. Kirshon, and D. Gonikman, "Owning the Routing Table – New OSPF Attacks", Black Hat USA, Aug. 2011.

[Infonetics12] Infonetics Research, "Enterprise Routers Quarterly Market Share, Size, and Forecasts", May 2012.

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

[NDSS12] G. Nakibly, A. Kirshon, D. Gonikman, and D. Boneh, "Persistent OSPF Attacks", NDSS, Feb. 2012.

[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

Appendix – Scapy attack script

```
attacker_source_ip = "192.168.13.1"
attacker_router_id = "192.168.18.1"
victim_destination_ip = "192.168.13.3"

victim_router_id = "192.168.37.3"
false_adv_router = "192.168.27.11"
seq_num = 0x80000004L

R3_FALSE_LSA = IP(src=attacker_source_ip, dst=victim_destination_ip) \
              /OSPF_Hdr(src=attacker_router_id) \
              /OSPF_LSUpd(lsalist=[ \
                     OSPF_Router_LSA(options=0x22, type=1, id=victim_router_id, adrouter=false_adv_router, seq=seq_num, linklist=[ \
                            OSPF_Link(id="192.168.37.7", data="192.168.37.3", type=2, metric=1), \
                            OSPF_Link(id="192.168.13.3", data="192.168.13.3", type=2, metric=1), \
                            OSPF_Link(id="192.168.50.0", data="255.255.255.0", type=3, metric=3) \
                     ])
              ])


send(R3_FALSE_LSA, iface="eth0")
```