

sniffer Packet

تحلیل گر بسته های شبکه یا آنچه که به طور معمول با نام هایی چون تحلیلگر شبکه (Network Analyzer) ، تحلیلگر پروتکل (Protocol Analyzer) یا Packet sniffer و یا آنچه در شرایطی برای شبکه های خاص با نام هایی چون Ethernet sniffer یا wireless sniffer شناخته می شود در حقیقت یک برنامه نرم افزاری یا قسمتی از سخت افزار کامپیوتر است که میتواند ترافیک شبکه یا بخشی از شبکه را رهگیری و فایل های گزارش بر این اساس تهیه نماید

هنگامی که جریان داده ها در یک شبکه از مسیری در جریان است ، Sniffer بسته های اطلاعاتی در طی این مسیر را گرفته و اگر نیاز باشد ، داده های خام آن بسته را Decode نموده و فیلد های مختلف به همراه داده های آن را از داخل بسته اطلاعاتی استخراج و نمایش میدهد و سپس تحلیل های لازم را بر اساس مشخصات و یا متود های RFC(request for comments) بر روی این اطلاعات انجام میدهد.

قابلیت های Packet Sniffing

در شبکه های LAN بسته به نوع ساختار آن Hub یا Switch ، فرد میتواند جریان داده ها بر روی بخشی از این شبکه یا Client خاص بر روی این شبکه را مانیتور نماید ، هر چند متود های وجود دارد که از دسترسی دیگر سیستم های شبکه و احاطه آنها بر روی جریان داده فوق الذکر جلوگیری می نماید. برای مثال میتوان روش ARP Spoofing را مثال زد که در واقع تکنیکی است که حمله کننده در آن ARP (Address Resolution Protocol) های جعلی را در سطح شبکه LAN ارسال نموده و قصد دارد تا MAC Address خود را به IP Address میزبانی دیگر نسبت دهد تا ارسال داده ها برای IP آدرس مذکور برای سیستم حمله کننده ارسال شود.

جهت مانیتور کردن یک شبکه حتی میتوان کلیه بسته های اطلاعاتی شبکه LAN را توسط یک سویچ به همراه یک پورت جهت مانیتورینگ (Monitoring Port) استفاده کرد که می تواند تمام بسته های ارسالی از طریق port های دیگر را هنگام اتصال یک سیستم به یکی از port های سویچ مورد نظر کپی برداری نماید.

در شبکه های بیسیم Wireless LAN ، میتوان ترافیک شبکه مورد نظر را بر روی یک و یا چندین کانال مختلف مانیتور نمود .

برخی برنامه های Sniffer هنگامی که ترافیک به صورت Multicast ارسال می شود با قرار گرفتن در مد promiscuous mode یا بی قاعده میتوانند همه ترافیک مورد نظر را دریافت نمایند (لازم به ذکر است همه ی Sniffer ها از این مد پشتیبانی به عمل نمی آورند).

در Sniffer ها اطلاعات دریافتی از داده های خام دیجیتالی ، رمز گشایی شده و به زبان قابل خواندن توسط انسان یا در اصطلاح زبان human-readable تبدیل می شوند که به کاربر این اجازه را می دهند که به راحتی اطلاعات رد و بدل شده را تحلیل نمایند. Sniffer ها در نمایش اطلاعات داده ها امکانات مختلفی را برای نمایش اطلاعات به کاربر عرضه میدارند مانند:

- نمایش ریشه خطاهای بوجود آمده

- نمایش نمودار زمانی

- بازسازی داده های TCP و UDP

برخی Sniffer ها خود میتوانند ترافیک ایجاد نموده و خود نقش دستگاه منبع را ایفا نمایند و در تست و تحلیل پروتکل های سیستم کارآمد باشند. این تست کننده ها در برخی مواقع این امکان را به کاربر میدهند تا عمدا برخی خطاها مربوط به DUT را ایجاد نمایند تا کارایی و قابلیت های سیستم در شرایط مشابه بررسی شود.

برخی از تحلیلگر ها نیز ممکن است سخت افزاری باشند ، این سخت افزار ها بسته های اطلاعاتی و یا قسمتی از آن را کپی برداری و بر روی دیسک سخت خود ذخیره می نمایند.

موارد استفاده از Packet Sniffing

موارد استفاده از Packet Sniffer ها میتواند متغییر باشد که در زیر میتوان به آنها اشاره نمود:

- تحلیل مشکلات شبکه ای
- تشخیص حمله های نفوذی
- تشخیص سوء استفاده از شبکه توسط کاربران داخلی و خارجی
- بدست آوردن اطلاعات مربوط به یک شبکه برای نفوذ به آن
- مانیتور کردن پهنای باند شبکه های WAN
- مانیتور کردن استفاده های کاربران خارجی و داخلی شبکه
- مانیتور کردن داده های موجود در جریان داده یک شبکه
- مانیتور کردن وضعیت های امنیتی شبکه WAN
- جمع آوری و گزارش آمار های مربوط به شبکه
- فیلتر سازی اطلاعات مشکوک از ترافیک شبکه
- جاسوسی بر روی شبکه های دیگر برای جمع آوری اطلاعات حساس مانند رمز های عبور (بسته به نوع رمز نگاری این داده ها)
- مهندسی معکوس داده های بر روی شبکه
- اشکال زدایی مربوط به ارتباط Client/Server بر روی شبکه
- اشکال زدایی طراحی پروتکل های شبکه

- کنترل و تایید سیستم های داخلی از نظر صحت کارکرد مانند Firewallها

برخی از Sniffer های معروف (Packet Analyzers)

- Capsa Network Analyzer
- Cain and Abel
- Carnivore (FBI)
- dSniff
- ettercap
- SkyGrabber
- snoop
- tcpdump
- Wireshark

روش های تشخیص packet sniffing در شبکه

همانگونه که اشاره گردید تشخیص این موضوع که یک فرد در یک بازه زمانی محدود و همزمان با حرکت بسته های اطلاعاتی در شبکه از یک packet sniffer استفاده می نماید ، کار مشکلی خواهد بود . با بررسی و آنالیز برخی داده ها می توان تا اندازه ای این موضوع را تشخیص داد :

- استفاده از امکانات ارائه شده توسط برخی نرم افزارها : در صورتی که مهاجمان دارای منابع محدودی باشند ممکن است از برنامه کاربردی Network Monitor برای packet sniffing استفاده نمایند . یک نسخه محدود از Network Monitor به همراه ویندوز NT و ۲۰۰۰ و یک نسخه کامل از آن به همراه SMS Server ارائه شده است . برنامه فوق ، گزینه ای مناسب برای مهاجمانی است که می خواهند در کوتاه ترین زمان به اهداف خود دست یابند چراکه استفاده از آن در مقایسه با سایر نرم افزارهای مشابه راحت تر است . خوشبختانه می توان بسادگی از اجرای این برنامه توسط سایر کاربران در یک شبکه ، آگاهی یافت . بدین منظور کافی است از طریق منوی Tools گزینه Identify Network Monitor Users را انتخاب نمود .

- بررسی سرویس دهنده DNS : در صورتی که مهاجمان از یکی از صدها نرم افزار ارائه شده برای packet sniffing استفاده نمایند ، امکان تشخیص سریع آن همانند برنامه Network Monitor وجود نخواهد داشت . توجه داشته باشید که یک روش صددرصد تضمینی به منظور تشخیص وجود یک برنامه packet sniffing در شبکه وجود ندارد ولی با مشاهده نشانه هائی خاص می توان احتمال وجود packet sniffing در شبکه را تشخیص داد . شاید بهترین نشانه وجود یک packet sniffing در شبکه به بانک اطلاعاتی سرویس دهنده DNS برگردد . سرویس

DNS دهنده وظیفه جستجو در بانک اطلاعاتی به منظور یافتن نام **host** و برگرداندن آدرس IP مربوطه را بر عهده دارد . در صورتی که مهاجمی یک **packet sniffing** را اجراء نماید که اسامی **host** را نمایش می دهد (اکثر آنان چنین کاری را انجام می دهند) ، ماشینی که فرآیند **packet sniffing** را انجام می دهد یک حجم بالا از درخواست های **DNS** را اجراء می نماید . در مرحله اول سعی نمائید ماشینی را که تعداد زیادی درخواست های **DNS lookups** را انجام می دهد ، بررسی نمائید . با این که وجود حجم بالائی از درخواست های **DNS lookup** به تنهایی نشاندهنده **packet sniffing** نمی باشد ولی می تواند به عنوان نشانه ای مناسب در این زمینه مطرح گردد . در صورتی که به یک ماشین خاص در شبکه مشکوک شده اید ، سعی نمائید یک ماشین طعمه را پیکربندی و آماده نمائید . ماشین فوق یک کامپیوتر شخصی است که کاربران از وجود آن آگاهی ندارد . پس از اتصال این نوع کامپیوترها به شبکه ، یک حجم بالای ترافیک بر روی شبکه را ایجاد نموده و به موازات انجام این کار درخواست های **DNS** را بررسی نمائید تا مشخص گردد که آیا ماشین مشکوک یک درخواست **DNS** را بر روی ماشین طعمه انجام می دهد . در صورتی که اینچنین است می توان با اطمینان گفت که ماشین مشکوک همان ماشین **packet sniffing** است .

- اندازه گیری زمان پاسخ ماشین های مشکوک : یکی دیگر از روش های متداول برای شناسایی افرادی که از **packet sniffing** استفاده می نمایند ، اندازه گیری زمان پاسخ ماشین مشکوک است . روش فوق مستلزم دقت زیاد و تا اندازه ای غیرمطمئن است . بدین منظور از دستور **Ping** ماشین مشکوک به منظور اندازه گیری مدت زمان پاسخ استفاده می شود . بخاطر داشته باشید فردی که عملیات **packet sniffing** را انجام می دهد تمامی بسته های اطلاعاتی را کپی نخواهد کرد ، چراکه حجم اطلاعات افزایش خواهد یافت . آنان با تعریف یک فیلتر مناسب، صرفاً " بسته های اطلاعاتی مورد علاقه خود را تکثیر می نمایند (نظیر آنانی که برای تأیید کاربران استفاده می گردد) . بنابراین از تعدادی از همکاران خود بخواهید که چندین مرتبه عملیات **log in** و **log out** را انجام داده و در این همین وضعیت مدت زمان پاسخ کامپیوتر مشکوک را محاسبه نمائید . در صورتی که مدت زمان پاسخ زیاد تغییر نکند ، آن ماشین احتمالاً " عملیات **packet sniffing** را انجام نمی دهد ولی در صورتی که زمان پاسخ کند گردد ، این احتمال وجود خواهد داشت که ماشین مشکوک شناسایی شده باشد.

- استفاده از ابزارهای مختص **AntiSniff** : شرکت های متعددی اقدام به طراحی و پیاده سازی نرم افزارهایی به منظور ردیابی و شناسایی **packet sniffing** نموده اند . برنامه های فوق از روش های اشاره شده و سایر روش های موجود به منظور شناسایی **packet sniffing** در یک شبکه استفاده می نمایند .

راه های مقابله با Packet Sniffing و Sniffer ها

- یکی از کاربردی ترین راه های مقابله با **Sniffing** ، استفاده از رمز نگاری در داده های رد و بدل شده در شبکه است و در واقع میتوانید داده های خود را با الگوریتم های معروف **Hash** نمایید
- استفاده از نرم افزار های **AntiSniff** که در واقع حضور هر گونه مانیتور بر روی شبکه شما را شناسایی مینمایند مانند برنامه های زیر:

- **sniffdet**

- Sniffer.Detectors

- ntop

- و در آخر میتوان با استفاده از یک Switch در شبکه هایی مانند Ethernet بسته های موجود در جریان داده های شبکه را به مقصد درست آنها راهنمایی کرد و این کار را نیز میتوان با یک پورت نظارتی و ایجاد یک Mirror انجام داد.

کتابخانه ها:

یکی از کتابخانه هایی که می توان به وسیله ی آن یک packet sniffer را در زبان C کد نویسی کرد Libpcap است.

دلیل انتخاب زبان برنامه نویسی C :

- زبان C نسبت به دیگر زبان های سطح بالا شما را به زبان ماشین نزدیکتر می کند.
- برنامه نویسی شبکه یک موضوع سرگرم کننده است ، در عین حال بسیار عمیق است و دارای سطوح بسیار است.
- برخی زبان های برنامه نویسی این انتزاع ها را پنهان می کنند . به عنوان مثال ، در زبان برن امه نویسی پایتون میتوانید کل صفحه وب را فقط با استفاد از یک خط کد بارگیری کنید . در C اینگونه نیست، در C اگر بخواهید یکصفحه وب را بارگیری کنید ، باید بدانید که همه چیز چگونه کار می کند. شما باید سوکت ها را بشناسید و در برنامه نویسی شبکه با زبان C ، هیچ چیز پنهان نیست.
- C یک زبان عالی برای یادگیری برنامه نوی سی شبکه است. این فقط به این دلیل نیست که همه جزئیات را می بینیم ، بلکه همچنین به این دلیل است که سیستم عامل های محبوب همه از هسته های نوشته شده در C استفاده می کنند. هیچ زبان دیگری همان دسترسی سطح اول را به شما نمی دهد . در C ، همه چیز تحت کنترل شما است - شما می توانید ساختار داده های خود را به طور دقیق تنظیم کنید. شما می توانید ، دقیقا حافظه را مطابق میل خود مدیریت کنید.
- استفاده از زبان برنامه ن ویسی C در همه جا وجود دارد. تقریبا هر پشته شبکه در زبان C برنامه نویسی شده است . این موضوع برای ویندوز ، لینوک س و macOS هم صدق می کند.

My repository:

<https://github.com/hadi1376tm/computer-network-project>

: (port sniffer) phase 1

در این فاز ابزار ما باید باز یا بسته بودن port های host مشخص شده در آرگومانهای خط فرمان را تشخیص دهد.

کاربر میتواند در سه حالت این ابزار را فراخوانی کند؛

- به صورت sniffer برای تمامی port های یک host ،
 - صرفا برای port های مشهور و رزرو شده
 - در نهایت در قالب یک پرسمان برای بررسی فراهم بودن سرویسهای لایه کاربرد.
- سرویسهای قابل پرسمان:

- HTTP
- TLS
- SMPT
- FTP
- TELNET
- SSH

پارامترهایی که کاربر باید بتواند مشخص کند:

زمان انتظار برای بررسی حالت هر port

بازه port ها برای بررسی

:Run commands

```
$gcc phase1.c -o phase1
$./phase1
```

توابع:

```
int socket_creation()
```

```
bool socket_connect(int fileDescriptor, char *host, uint16_t port)
```

```
void remove_cr(char* str)
```

```
void range_port_scan(char *hostname , int start , int end)
```

```
void try_connecting(char* hostname , int port)
```

```
void ask_options(char* hostname)
```

```
void ask_port_service(char* hostname)
```

: (ping)phase 2

Ping معیاری برای احتساب فاصله‌ی زمانی اتصال بین دو دستگاه در شبکه است. از طریق میزان پینگ یک سرویس اینترنتی، می‌توان دریافت که چقدر طول خواهد کشید که یک بسته اطلاعات از سمت دستگاه هوشمند مانند کامپیوتر به سمت سرور رفته و دوباره از آن مسیر برگردد. پینگ درواقع تأخیر زمانی است که در هنگام انجام هر کار Online اتفاق می‌افتد، از کلیک بر روی لینک، تا استریم کردن یک ویدئو.

اما Ping صرفاً ابزاری برای اندازه‌گیری «تأخیر» نیست. هر شبکه از تعداد متعددی دستگاه و سرور تشکیل شده است. هرکدام از این دستگاه‌ها می‌توانند داده‌ای را که آن را «Packet» می‌نامند به دستگاه هوشمندی که مقصد است ارسال کند. دستگاه مقصد بسته‌ی ارسالی که نیاز به پاسخ دارد را دریافت می‌کند و در جواب یک Packet به دستگاه اول ارسال می‌کند.

این مراحل برای هر پینگ تکرار می‌شود. پینگ دستوری است که این امکان را فراهم می‌کند مدت زمان رفت و برگشت بسته‌ها را بین دستگاه‌ها، اندازه بگیرد. استفاده از دستور پینگ برای چک کردن وضعیت سرورها بسیار مفید است. اگر یکی از سرورها down شود و یا کند عمل کند، می‌تواند مانع اتصال دو دستگاه به یکدیگر شود. در این وضعیت Packet Loss اتفاق خواهد افتاد و همان‌طور که از نامش مشخص است، Packet Loss زمانی اتفاق می‌افتد که بسته یا بسته‌هایی از داده به مقصدشان نمی‌رسند.

Ping همچنین کمک می‌کند تا بسته‌هایی که از بین رفته‌اند دیده شوند. برای مثال فرض کنید دستور پینگ را اجرا کرده‌اید و با پاسخ‌های مختلف روبرو شده‌اید. مثلاً Request timed out یعنی Packet توسط رایانه مقصد دریافت نشده است، یا پاسخ‌های سرور به دستگاه اول ارسال نشده است. یا ممکن است در طول مسیر ارسال و دریافت از بین رفته باشد.

: phase 3

یکی از ابزارهای جالب و کاربردی شبکه traceroute می‌باشد. این ابزار تمام hop های بین ما و host مد نظرمان را در صورت امکان پیدا می‌کند. این ابزار می‌تواند مبتنی بر پروتکل های متعددی پیاده سازی شود. این پروژه با پروتکل TCP پیاده سازی شده است.

ابزار **تریس روت** به وسیله افزایش مقدار ttl برای هر خوشه ارسال شده از بسته‌ها کار می‌کند. سه بسته فرستاده شده نخست تی‌تی‌ال مقدار یک دارند(بر این دلالت می‌کند که این بسته‌ها به وسیله میزبان به جای

میزبان بعدی ارجاع نشوند). خوشه بسته‌های بعدی مقدار تی‌تی‌ال دو دارند و به همین ترتیب مقدار تی‌تی‌ال خوشه‌های بسته‌ها زیاد می‌شود. وقتی یک بسته به یک میزبان می‌رسد، به‌طور عادی یک عدد از مقدار تی‌تی‌ال آن کم می‌شود، و پس از آن بسته به مقصد بعدی ارجاع داده می‌شود. ولی هنگامی که یک بسته با مقدار تی‌تی‌ال یک به میزبانی برسد، میزبان بسته را دور می‌ریزد و پیغام ICMP با محتوای از حد زمانی تجاوز شد به فرستنده بازپس می‌فرستد. ابزار تریس روت از این نوع بازگشت بسته (رد کردن بسته) استفاده می‌کند تا لسیتهی از میزبان‌هایی که بسته با مسیریابی منتقل شده تا به مقصد برسد را تولید کند. در ضمن سه مقدار زمانی (از فاصله تا میزبان) برای هر یک از میزبان‌هایی که بسته باید طی کند با واحد میلی‌ثانیه بازگردانده می‌شود.

:Run commands

```
$gcc TCP_traceroute.c -o TCP_traceroute
$sudo ./TCP_traceroute <host name or IP>
```

توابع:

```
int main (int argc, char *argv[])
```

```
char* reverse_dns_lookup(char *ip_addr)
```

: (Host discovery)phase 4

در این فاز ابزار ما با کمک **arp** اقدام با استفاده از ابزار **scapy** به شناسایی **host** های موجود در شبکه می‌کند. شیوه عملکرد به این صورت است که کاربر بازه **ip** نیازمند بررسی را در اختیار ابزار قرار می‌دهد و ابزار با استفاده از **arp request** اقدام به بررسی زنده بودن **ip** یا **host** می‌کند و نتیجه را چاپ می‌کند.

پارامترهایی که کاربر باید بتواند مشخص کند:

- **Ip** شروع و **Ip** پایان یا آدرس به صورت **mask**
- حداکثر زمان انتظار برای جواب

:ARP

زمانی که **Host 1** قصد برقراری ارتباط با سایر دستگاه های شبکه را دارد به دو مورد برای برقراری ارتباط نیاز دارد مورد اول آیپی **Host 2** و مورد دوم مک آدرس **Host 2** می‌باشد **Host 1**. برای پیدا کردن مک مقصد

در مرحله اول arp cache خود را چک میکند در صورتی که مک آدرس در arp cache وجود داشته باشد بسته را ارسال می کند. اما اگر مک آدرس در arp cache وجود نداشته باشد Host 1 یک arp request ارسال می کند و در پیام ارسالی به جای مک مقصد از مک FF:FF:FF:FF:FF:FF استفاده می کند زمانی که بسته به سوئیچ می رسد، سوئیچ با دیدن این بسته مک Host 1 را روی پورت متصل به Host 1 یادداشت می کند و بسته را روی تمام پورت های خود Broadcast می کند و از تمام Host های شبکه سوال می کند که ”اگر آییی شما با آییی پکت ارسالی مطابقت دارد، لطفا مک دستگاه خودتون را ارسال کنید“.

:Run commands

```
$sudo python3 phase4.py
```

توابع:

```
check_IP_range(choice)
```

```
--main--
```