مفاهیم شبکه

مروری بر مفاهیم شبکه:

برای تحلیل و فهم روشهائی که یک نفوذگر با بکارگیری آنها با شبکه حمله می کند، باید یک دانش پایه از تکنولوژی شبکه داشته باشیم. درک مکانیزم حملات ممکن نیست مگر آنکه حداقل اصول TCP/IP را بدانیم.

عاملی که تمام شبکه های مختلف را به صورت موفقیت آمیز به هم پیوند زده است، تبعیت همه آنها از مجموعه پروتکلی است که تحت عنوان TCP/IP در دنیا شناخته می شود. دقت کنید که عبارت خلاصه شده TCP/IP می تواند به دو موضوع متفاوت اشاره داشته باشد:

مدل TCP/IP: این مدل یک ساختار چهار لایه ای برای ارتباطات گسترده تعریف می نماید که آنرا در ادامه بررسی می کنیم.

پشتهٔ پروتکلهای TCP/IP:[1] پشتهٔ TCP/IP مجموعه ای شامل بیش از صد پروتکل متفاوت است که برای سازماندهی کلیه اجزاء شبکهٔ اینترنت به کار می رود.

TCP/IP بهترین پروتکل شبکه بندی دنیا نیست! پروتکلهای بهینه تر از آن هم وجود دارند؛ ولیکن فراگیرترین و محبوبترین تکنولوژی شبکه بندی در دنیای کامپیوتر محسوب می شود. شاید بزرگترین حسن TCP/IP آن باشد که بدون پیچیدگی زیاد، بخوبی کار می کند! اینترنت بر اساس TCP/IP بنا شده و بیشتر حملات نیز مبتنی بر مجموعهٔ پروتکلهای TCP/IP هستند.

---

[1] TCP/IP Protocol Stack

طراحی شبکه ها و اصول لایه بندی

برای طراحی یک شبکهٔ کامپیوتری، مسائل و مشکلات بسیار گسترده و متنوعی وجود دارد که باید به نحوی حل شود تا بتوان یک ارتباط مطمئن و قابل اعتماد بین دو ماشین در شبکه برقرار کرد. این مسائل و مشکلات همگی از یک سنخ نیستند و منشأ و راه حل مشابه نیز ندارند؛ بخشی از آنها توسط سخت افزار و بخش دیگر با تکنیکهای نرم افزاری قابل حل هستند. به عنوان مثال نیاز برای ارتباط بی سیم بین چند ایستگاه در شبکه، طراح شبکه را مجبور به استفاده از مدولاسیون آنالوگ در سخت افزار مخابراتی خواهد کرد ولی مسئلهٔ هماهنگی در ارسال بسته ها از مبدأ به مقصد یا شماره گیری بسته ها برای بازسازی پیام و اطمینان از رسیدن یک بسته، با استفاده از تکنیکهای نرم افزازی قابل حل است. بهمین دلیل برای طراحی شبکه های کامپیوتری، باید مسائل و مشکلاتی که برای برقراری یک ارتباط مطمئن، ساده و شفاف بین دو ماشین در شبکه وجود دارد، دسته بندی شده و راه حلهای استاندارد برای آنها ارائه می شود. در زیربخشی از مسائل طراحی شبکه ها عنوان شده است:

اولین موضوع چگونگی ارسال و دریافت بیتهای اطلاعات بصورت یک سیگنال الکتریکی، الکترومغناطیسی یا نوری است، بسته به اینکه آیا کانال نتقال سیم مسی، فیبرنوری، کانال ماهواره ای یا خطوط مایکروویو است. بنابراین تبدیل بیتها به یک سیگنال متناسب با کانال انتقال یکی از مسائل اولیهٔ شبکه به شمار می رود.

مساله دوم ماهیت انتقال است که می تواند به یکی از سه صورت زیر باشد:

Simplex: ارتباط یک طرفه (یک طرف همیشه گیرنده و طرف دیگر همیشه فرستنده).

Half Duplex: ارتباط دو طرفهٔ غیرهمزمان (هر دو ماشین هم می توانند فرستنده یا گیرنده باشند ولی نه بصورت همزمان، بلکه یکی از طرفین ابتدا ارسال می کند، سپس ساکت می شود تا طرف مقابل ارسال داشته باشد)

Full Duplex: ارتباط دو طرفه همزمان (مانند خطوط مایکروویو)

مساله سوم مسئله خطا و وجود نویز روی کانالهای ارتباطی است بدین معنا که ممکن است در حین ارسال داده ها بر روی کانال فیزیکی تعدادی از بیتها دچار خرابی شود؛ چنین وضعیتی که قابل اجتناب نیست باید تشخیص داده شد و داده های فاقد اعتبار دو ریخته شود مبدأ آنها را از نو ارسال کند.

با توجه به اینکه در شبکه ها ممکن است مسیرهای گوناگونی بین مبدأ و مقصد وجود داشته باشد؛ بنابراین پیدا کرن بهترین مسیر و هدایت بسته ها، از مسائل طراحی شبکه محسوب می شود. در ضمن ممکن است یک پیام بزرگ به واحدهای کوچکتری تقسیم شده و از مسیرهای مختلفی به مقصد برسد بنابراین بازسازی پیام از دیگر مسائل شبکه به شمار می آید.

ممکن است گیرنده به دلایلی نتواند با سرعتی که فرستنده بسته های یک پیام را ارسال می کند آنها را دریافت کند، بنابراین طراحی مکانیزمهای حفظ هماهنگی بین مبدأ و مقصد از دیگر مسائل شبکه است.

چون ماشینهای فرستنده و گیرندهٔ متعددی در یک شبکه وجود دارد مسائلی مثل ازدحام، تداخل و تصادم در شبکه ها بوجود می آید که این مشکلات بهمراه مسائل دیگر باید در سخت افزاز و نرم افزار شبکه حل شود.

طراح یک شبکه باید تمام مسائل شبکه را تجزیه و تحلیل کرده و برای آنها راه حل ارائه کند ولی چون این مسائل دارای ماهیتی متفاوت از یکدیگر هستند، بنابراین طراحی یک شبکه باید بصورت «لایه به لایه» انجام شود. به عنوان مثال وقتی قرار است یک شبکه به گونه ای طراحی شود که ایستگاهها بتوانند انتقال فایل داشته باشند، اولین مسئله ای که طراح باید به آن بیندیشد طراحی یک سخت افزار مخابراتی برای ارسال و دریافت بیتها روی کانال فیزیکی است. اگر چنین سخت افزاری طراحی شود، می تواند بر اساس آن اقدام به حل مسئلهٔ خطاهای احتمالی در داده ها نماید؛ یعنی زمانی مکانیزمهای کنترل و کشف خطا مطرح می شود که قبل از آن سخت افزار مخابرهٔ داده ها طراحی شده باشد. بعد از این دو مرحلهٔ طراحی، باید مکانیزمهای بسته بندی اطلاعات، آدرس دهی ماشینها و مسیریابی بسته ها طراحی شود. سپس برای بقیه مسائل نظیر آدرس دهی پروسه ها و چگونگی انتقال فایل راه حل ارائه شود.

طراحی لایه ای شبکه به منظور تفکیک مسائلی است که باید توسط طراح حل شود و مبتنی بر اصول زیر است:[1]

[1] طراحي لايه اي شبكه را مي توان با برنامه نويسي ماجولار مقايسه كرد، بدين نحو كه روالهاي حل يك مسئله با اجزاي كوچكتري شكسته مي شود و براي آن زيربرنامه نوشته مي شود. در توابع صدا زنندهٔ اين زيربرنامه ها، جزئيات دروني آنها اهميت

- هر لایه وظیفه مشخصی دارد و طراح شبکه باید آنها را به دقت تشریح کند.
- هر گاه سرویسهایی که باید ارائه شود از نظر ماهیتی متفاوت باشد، لایه به لایه و جداگانه طراحی شود.
- وظیفه هر لایه باید با توجه به قراردادها و استانداردهای جهانی مشخص شود.
- تعداد لایه ها نباید آنقدر زیاد باشد که تمیز لایه ها از دیدگاه سرویسهای ارائه شده نامشخص باشد و نه آنقدر کم باشد، که وظیفه و خدمات یک لایه، پیچیده و نامشخص شود.
- در هر لایه جزئیات لایه های زیرین نادیده گرفته می شود و لایه های بالایی باید در یک روال ساده و ماجولار از خدمات لایهٔ زیرین خود استفاده کنند.
- باید مرزهای هر لایه به گونه ای انتخاب شود که جریان اطلاعات بین لایه ها، حداقل باشد.

برای آنکه طراحی شبکه ها سلیقه ای و پیچیده نشود سازمان جهانی استاندارد[1] (ISO)، مدلی هفت لایه ای برای شبکه ارائه کرد، به گونه ای که وظایف و خدمات شبکه در هفت لایهٔ مجزا تعریف و ارائه می شود. این مدل هفت لایه ای،OSI[2] نام گرفت. هر چند در شبکهٔ اینترنت از این مدل استفاده نمی شود و بجای آن یک مدل چهار لایه ای

ندارد بلکه فقط نحوهٔ صدا زدن آنها و پارامترهای مورد نیاز ورودی به زیربرنامه و چگونگی برگشت نتیجه به صدا زننده، مهم است.

[1] International Standard Organization
[2] Open System Interconnection

به نام TCP/IP تعریف شده است، ولیکن بررسی مدل هفت لایه ای OSI، بدلیل دقتی که در تفکیک و تبیین مسائل شبکه در آن وجود دارد، با ارزش خواهد بود. پس از بررسی مدل OSI، به تشریح مدل TCP/IP خواهیم پرداخت.

مدل هفت لایه ای OSI از سازمان استاندارد جهانی ISO

در این استاندارد کل وظایف و خدمات یک شبکه در هفت لایه تعریف شده است:

لایه ۱- لایه فیزیکی	Physical Layer
لایه ۲- لایه پیوند داده ها	Data Link Layer
لایه ۳- لایه شبکه	Network Layer
لایه ٤- لایه انتقال	Transport Layer
لایه ٥- لایه جلسه	Session Layer
لایه ٦- لایه ارائه (نمایش)	Presentation Layer
لایه ۷- لایه کاربرد	Application Layer

از لایه های پایین به بالا، سرویسهای ارائه شده (با تکیه بر سرویسی که لایه های زیرین ارائه می کنند) پیشرفته تر می شود.

این مدل به منظور تعریف یک استاندارد جهانی و فراگیر ارائه شد و گمان می رفت که تمام شبکه ها بر اساس این مدل در هفت لایه طراحی شوند، به گونه ای که در دهۀ هشتاد سازمان ملی علوم در آمریکا عنوان کرد که در آینده فقط از این استاندارد حمایت خواهد کرد، ولی در عمل، طراحان شبکه به این مدل وفادار نماندند.

در ادامه به اختصار وظائف هر لایه در مدل OSI را تعریف خواهیم کرد.

لایه فیزیکی

وظیفه اصلی در لایهٔ فیزیکی، انتقال بیتها بصورت سیگنال الکتریکی و ارسال آن بر روی کانال می باشد. واحد اطلاعات در این لایه بیت است و بنابراین این لایه هیچ اطلاعات از محتوای پیام ندارد و تنها بیتهای ۰ و ۱ را ارسال یا دریافت می کند پارامترهایی که باید در این لایه مورد نظر باشند عبارتند از: ظرفیت کانال فیزیکی و نرخ ارسال[1]، نوع مدولاسیون، چگونگی کوپلاژ با خط انتقال، مسائل مکانیکی و الکتریکی مانند نوع کابل، باند فرکانسی و نوع رابط (کانکتور) کابل.

در این لایه که تماماً سخت افزاری است، مسایل مخابراتی در مبادلهٔ بیتها، تجزیه و تحلیل شده و طراحی های لازم انجام می شود. طراح شبکه می تواند برای طراحی این لایه، از استانداردهای شناخته شدهٔ انتقال همانند RS-232 و RS-422 و RS-423 و ... که سخت افزار آنها موجود است، استفاده کند. این لایه هیچ وظیفه ای در مورد تشخیص و ترمیم خطا ندارد.

لایهٔ پیوند داده ها

وظیفهٔ این لایه آن است که با استفاده از مکانیزمهای کشف و کنترل خطا، داده ها را روی یک کانال انتقال که ذاتاً دارای خطا است، بدون خطا و مطمئن به مقصد برساند. در حقیقت می توان وظیفهٔ این لایه را بیمهٔ اطلاعات در مقابل خطاهای احتمالی دانست؛

[1] Channel Capacity and Bit Rate

زیرا ماهیت خطا به گونه ای است که قابل رفع نیست ولی می توان تدابیری اتخاذ کرد که فرستنده از رسیدن یا نرسیدن صحیح اطلاعات به مقصد مطلع شده و در صورت بروز خطا مجدداً اقدام به ارسال اطاعات کند؛ با چنین مکانیزمی یک کانال دارای خطا به یک خط مطمئن و بدون خطا تبدیل خواهد شد.

یکی دیگر از وظایف لایهٔ پیوند داده ها آن است که اطلاعات ارسالی از لایهٔ بالاتر را به واحدهای استاندارد و کوچکتری شکسته و ابتدا و انتهای آن را از طریق نشانه های خاصی که Delimiter نامیده می شود، مشخص نماید. این قالب استاندارد که ابتدا و انتهای آن دقیقاً مشخص شده، فریم نامیده می شود؛ یعنی واحد اطلاعات در لایهٔ دو فریم است.

کشف خطا که از وظایف این لایه می باشد از طریق اضافه کردن بیتهای کنترل خطا مثل بیتهای Parity Check و Checksum و CRC انجام می شود.

یکی دیگر از وظایف لایهٔ دوم کنترل جریان یا به عبارت دیگر تنظیم جریان ارسال فریم ها به گونه ای است که یک دستگاه کند هیچ گونه فریمی را به خاطر آهسته بودن از دست ندهد. از دیگر وظایف این لایه آن است که وصول داده ها یا عدم رسید داده ها را به فرستنده اعلام کند.

یکی دیگر از وظایف این لایه آن است که قراردادهایی را برای جلوگیری از تصادم سیگنال ایستگاههایی که از کانال اشتراکی استفاده می کنند، وضع کند چرا که فرمان

ارسال داده بر روی کانال مشترک از لایهٔ دوم صادر می شود. این قراردادها در زیر لایه ای به نام MAS[1] تعریف شده است.

وقتی یک واحد اطلاعاتی تحویل یک ماشین متصل به کانال فیزیکی در شبکه شد، وظیفهٔ این لایه پایان می یابد. از دیدگاه این لایه، ماشینهائی که به کانال فیزیکی متصل نمی باشند، در دسترس نیستند. کنترل سخت افزار لایهٔ فیزیکی به عهدهٔ این لایه است.

فراموش نکنید که وظایف این لایه نیز با استفاده از سخت افزارهای دیجیتال انجام می شود.

لایهٔ شبکه

در این لایه اطلاعات به صورت بسته هایی سازماندهی می شود و برای انتقال مطمئن تحویل لایهٔ دوم می شود. با توجه به آنکه ممکن است بین دو ماشین در شبکه مسیرهای گوناگونی وجود داشته باشد، لذا این لایه وظیفه دارد هر بسته اطلاعاتی را پس از دریافت به مسیری هدایت کند تا آن بسته بتواند به مقصد برسد. در این لایه باید تدابیری اندیشیده شود تا از ازدحام (یعنی ترافیک بیش از اندازهٔ بسته ها در یک مسیریاب یا مرکز سوئیچ) جلوگیری شده و از ایجاد بن بست ممانعت بعمل بیاورد.

هر مسیرباب می تواند به صورت ایستا و غیرهوشمند بسته ها را مسیریابی کند. همچنین می تواند به صورت پویا و هوشمند برای بسته ها مسیر انتخاب نماید. در این لایه تمام

[1] Medium Access Sublayer

ماشینهای شبکه دارای یک آدرس جهانی و منحصر به فرد خواهند بود که هر ماشین بر اساس این آدرسها اقدام به هدایت بسته ها به سمت مقصد خواهد کرد.

این لایه ذاتاً «بدون اتصال»[1] است یعنی پس از تولید یک بستهٔ اطلاعاتی در مبدأ، بدون هیچ تضمینی در رسیدن آن بسته به مقصد، بسته شروع به طی مسیر در شبکه می کند. وظائف این لایه به سیستم نامه رسانی تشبیه شده است؛ یک پاکت محتوی نامه پس از آنکه مشخصات لازم بر روی آن درج شد، به صندوق پست انداخته می شود، بدون آنکه بتوان زمان دقیق رسیدن نامه و وجود گیرنده نامه را در مقصد، از قبل حدس زد. در ضمن ممکن است نامه به هر دلیلی گم شود یا به اشتباه در راهی بیفتد که مدتها در سیر بماند و زمانی به گیرندهٔ آن برسد که هیچ ارزشی نداشته باشد.

در این لایه تضمینی وجود ندارد وقتی بسته ای برای یک ماشین مقصد ارسال می شود آن ماشین آمادهٔ دریافت آن بسته باشد و بتواند آنرا دریافت کند. در ضمن هیچ تضمینی وجود ندارد وقتی چند بستهٔ متوالی برای یک ماشین ارسال می شود به همان ترتیبی که بر روی شبکه ارسال شده، در مقصد دریافت شوند. همچنین ممکن است که وقتی بسته ای برای یک مقصد ارسال می گردد، به دلیل دیر رسیدن از اعتبار ساقط شده و مجدداً ارسال شود و هر دو بسته (جدید و قدیم) به هم برسند. این مسائل در لایهٔ بالاتر قابل حل خواهد بود.

---

[1] Connectionless

هر چند وظائف این لایه می تواند بصورت نرم افزاری پیاده شود ولی برای بالاتر رفتن سرعت عمل شبکه، می توان برای این لایه یک کامپیوتر خاص طراحی نمود تا در کنار سخت افزار لایه های زیرین، بسته ها را روی شبکه رد و بدل کند.

لایهٔ انتقال

در این لایه بر اساس خدمات لایهٔ زیرین، یک سرویس انتقال بسیار مطمئن و «اتصال گرا»[1] ارائه می شود. تمام مشکلاتی که در لایهٔ شبکه عنوان شد در این لایه حل و فصل می شود:

- قبل از ارسال بسته ها، نرم افزار این لایهٔ اقدام به ارسال یک بسته ویژه می نماید تا مطمئن شود که ماشین گیرنده آمادهٔ دریافت اطلاعات است.
- جریان ارسال اطلاعات شماره گذاری شده تا هیچ بسته گم نشود یا دوبار دریافت نشود.
- ترتیب جریان بسته ها حفظ می شود.
- در این لایه پروسه های مختلفی که بر روی یک ماشین واحد اجرا شده اند، آدرس دهی می شوند به نحوی که هر پروسه بر روی یک ماشین واحد، به عنوان یک هویت مستقل داده های خود را ارسال یا دریافت نماید.

واحد اطلاعات در این لایه قطعهٔ[2] است. از وظائف دیگر این لایه می توان به موارد زیر اشاره کرد:

---

[1] Connection Oriented
[2] Segment

- تقسیم پیامهای بزرگ به بسته های اطلاعات کوچکتر
- بازسازی بسته های اطلاعاتی و تشکیل یک پیام کامل
- شماره گذاری بسته های کوچکتر جهت بازسازی
- تعیین و تبیین مکانیزم نامگذاری ایستگاه هایی که در شبکه اند.
- وظائف این لایه (و لایه های بعدی) با استفاده از نرم افزار پیاده سازی می شود و فقط بر روی ماشینهای نهایی (Hosts) وجود دارد و مراکز سوئیچ به وظائف این لایه احتیاجی ندارند (مگر در موارد خاص).

لایهٔ جلسه

وظیفهٔ این لایه فراهم آوردن شرایط یک جلسه (نشست) همانند ورود به سیستم از راه دور[1]، احراز هویت طرفین، نگهداری این نشست و توانایی از سرگیری یک نشست در هنگام قطع ارتباط می باشد. وظایف این لایه را می توان در موارد زیر خلاصه کرد: برقراری و مدیریت یک جلسه، شناسایی طرفین، مشخص نمودن اعتبار پیامها، اتمام جلسه، حسابداری مشتری ها[2]

لایهٔ ارائهٔ (نمایش)

در این لایه معمولا کارهایی صورت می گیرد که اگر چه بنیادی و اساسی نیستند ولیکن به عنوان نیازهای عمومی تلقی می شوند. مثل: فشرده سازی فایل[3]، رمزنگاری[4] برای

[1] Remote Login
[2] Accounting
[3] Data Compression
[4] Encryption

ارسال داده های محرمانه، رمزگشایی[1]، تبدیل کدها به یکدیگر (وقتی که دو ماشین از استانداردهای مختلفی برای متن استفاده می کنند؛ مثل تبدیل متون EBCDIC به ASCII و بالعکس)

لایهٔ کاربرد

در این لایه، استاندارد مبادلهٔ پیام بین نرم افزاهائی که در اختیار کاربر بوده و به نحوی با شبکه در ارتباطند، تعریف می شود. لایهٔ کاربرد شامل تعریف استانداردهایی نظیر انتقال نامه های الکترونیکی، انتقال مطمئن فایل، دسترسی به بانکهای اطلاعاتی راه دور، مدیریت شبکه و انتقال صفحات وب است.

در مدل لایه ای شبکه، وقتی یک برنامهٔ کاربردی در لایهٔ آخر اقدام به ارسال یک واحد اطلاعات می نماید، سرآیند لازم به آن اضافه شده و از طریق صدا زدن توابع سیستمی استاندارد به لایهٔ زیرین تحویل داده می شود. لایهٔ زیر نیز پس از اضافه کردن سرآیند لازم، آنرا به لایهٔ پایین تحویل می دهدو این روند تکرار می شود تا آن واحد اطلاعات روی کانال فیزیکی ارسال شود. در مقصد پس از دریافت یک واحد اطلاعات از روی خط فیزیکی، تحویل لایهٔ بالاتر شده و در هر لایه پس از تحلیل و پردازش لازم، سرآیند اضافه شده را حذف و به لایهٔ بالاتر تحویل می دهد. در شکل (۱-۲) روند حذف و اضافه شدن سرآیند در هر لایه به تصویر کشیده شده است.

---

[1] Decryption

مدل چهار لایه ای TCP/IP

همانگونه که اشاره شد این مدل یک ساختار چهار لایه ای برای شبکه عرضه کرده است. شکل (۲-۲) این مدل را به تصویر کشیده است. اگر بخواهیم این مدل چهار لایه ای را با مدل OSI مقایسه کنیم، لایهٔ اول از مدل TCP/IP یعنی لایهٔ دسترسی به شبکهٔ تلفیقی از وظائف لایهٔ فیزیکی و لایهٔ پیوند داده ها از مدل OSI خواهد بود. لایهٔ دوم از مدل TCP/IP معادل لایهٔ سوم از مدل OSI یعنی لایهٔ شبکه است. لایهٔ سوم از مدل TCP/IP همنام و معادل با لایهٔ چهارم از مدل OSI یعنی لایهٔ انتقال خواهد بود. لایه پنجم (جلسه) و لایهٔ ششم (ارائه) از مدل OSI در مدل TCP/IP وجود ندارند و وظائف آنها در صورت لزوم در لایهٔ چهارم از مدل TCP/IP ادغام شده است. لایهٔ هفتم از مدل OSI معدل بخشی از لایهٔ چهارم از مدل TCP/IP است. در شکل (۳-۲) دو مدل TCP/IP و OSI با هم مقایسه شده اند.

در ادامه چهار لایهٔ مدل TCP/IP را بررسی خواهیم کرد.

زیربنای اینترنت ساختار چهار لایه ای TCP/IP است. در این کتاب یاد خواهید گرفت که حملات نفوذگران نیز در یکی از این چهار لایه شکل می گیرد؛ لذا ماهیت و مکانیزمهای حمله و همچنین ابزار و هدف حمله وابسته به لایه ای است که مورد حمل قرار می گیرد.

شکل (۱-۲) روند حذف و اضافه شدن سرآیند در هر لایه

لایه ها	نامهای معادل در برخی از کتب
لایه کاربرد Application layer	• لایه سرویسهای کاربردی
لایه انتقال Transport layer	• لایه ارتباط میزبان به میزبان (Host to Host) • لایه ارتباط عناصر انتهایی (End to End Connection)
لایه شبکه Network layer	• لایه اینترنت • لایه ارتباطات اینترنت
لایه واسط شبکه Network interface	• لایه میزبان به شبکه (Host to network) • لایه رابط شبکه

شکل (۲-۲) مدل چهار لایه ای TCP/IP

شکل (۳-۲) مقایسهٔ دو مدل TCP/IP و OSI

لایهٔ اول از مدل TCP/IP : لایهٔ واسط شبکه

در این لایه استانداردهای سخت افزار، نرم افزار های راه انداز[1] و پروتکلهای شبکه تعریف می شود. این لایه درگیر با مسائل فیزیکی، الکتریکی و مخابراتی کانال انتقال، نوع کارت شبکه و راه اندازه های لازم برای نصب کارت شبکه می باشد. در شبکهٔ اینترنت که می تواند مجموعه ای از عناصر غیرهمگن و نامشابه را به هم پیوند بزند

[1] Device Driver

انعطاف لازم در این لایه برای شبکه های گوناگون و ماشینهای میزبان فراهم شده است. یعنی الزام ویژه ای در بکارگیری سخت افزار ارتباطی خاص، در این لایه وجود ندارد. ایستگاهی که تصمیم دارد به اینترنت متصل شود بایستی با استفاده از پروتکلهای متعدد و معتبر و نرم افزار راه انداز مناسب، به نحوی داده های خودش را به شبکه تزریق کند. بنابراین اصرار و اجبار خاصی در استفاده از یک استاندارد خاص در این لایه وجود ندارد. تمام پروتکلهای MAN LAN در این لایه قابل استفاده است.

یک ماشین میزبان می تواند از طریق شبکهٔ محلی، فریمهای اطلاعاتی را به زیر شبکه تزریق کند به این نحو که بسته های راه دور[1] را که مقصدشان خارج از شبکهٔ محلی است، به مسیریاب از پیش تعریف شده، هدایت نماید. شبکه های محلی از طریق یک یا چند مسیریاب می توانند به اینترنت متصل شوند. بنابراین یک بسته اطلاعاتی که از لایه بالاتر جهت ارسال به یک مقصد، به لایه اول در مدل TCP/IP تحویل می شود، نهایتاً در قسمت «فیلد داده»[2] از فریم شبکه محلی قرار می گیرد و مسیر خود را آغاز می نماید؛ پروتکلهائی که در لایهٔ اول از مدل TCP/IP تعریف می شوند، می توانند مبتنی بر ارسال رشتهٔ بیت[3] یا مبتنی بر ارسال رشتهٔ بایت[4] باشند.

---

[1] Distant Packet

[2] Data Field/Payload

[3] Bit oriented در اینجا کوچکترین واحد اطلاعات که می تواند بطور مستقل ارسال شود یک بیت خواهد بود.

[4] Byte oriented در اینجا کوچکترین واحد اطلاعات که می تواند بطور مستقل ارسال شود یک بایت خواهد بود.

لایهٔ دوم از مدل TCP/IP : لایهٔ شبکه

این لایه در ساده ترین عبارت وظیفه دارد بسته های اطلاعاتی را که از این به بعد آنها را بسته های IP می نامیم، روی شبکه هدایت کرده و از مبدأ تا مقصد به پیش ببرد. در این لایه چندین پروتکل در کنار هم وظیفه مسیریابی و تحویل بسته های اطلاعاتی از مبدأ تا مقصد را انجام می دهند. کلیدی ترین پروتکل در این لایه، پروتکل IP نام دارد. برخی از پروتکلهای مهم که یک سری وظایف جانبی برعهده دارند عبارتند از: BOOTP,IGMP,ICMP,RIP,RARP,ARP و .... . این پروتکلها را به اختصار توضیح خواهیم داد ولی بیشترین تلاش ما در کالبدشناسی پروتکلIP خواهد بود.

همانگونه که اشاره شد در این لایه یک واحد اطلاعاتی که بایستی تحویل مقصد شود، دیتاگرام نامیده می شود. پروتکل IP می تواند یک دیتاگرام را در قالب بسته های کوچکتری قطعه قطعه کرده و پس از اضافه کردن اطلاعات لازم برای بازسازی، آنها را روی شبکه ارسال کند.

لازم است بدانید که در این لایه برقرای ارتباط بین مبدأ و مقصد بروش «بدون اتصال» خواهد بود و از اسال یک بستهIP روی شبکه، عبور از مسیر خاصی را تضمین نمی کند. بعنی اگر دو بسته متوالی برای یک مقصد یکسان ارسال شود هیچ تصمینی در به ترتیب رسیدن آنها وجود ندارد، چون این دو بسته می توانند از مسیرهای متفاوتی به سمت مقصد حرکت نماید. در ضمن در این لایه پس از آنکه بسته ای روی یکی از کانالهای ارتباطی هدایت شد، از سالم رسیدن یا نرسیدن آن به مقصد هیچ اطلاعی بدست نخواهد

آمد، چرا که در این لایه، برای بسته های IP هیچ گونه پیغام دیافت یا عدم دریافت[1] بین عناصر واقع بر روی مسیر، رد و بدل نمی شود؛ بنابراین سرویسی که در این لایه ارائه می شود نامطمئن است و اگر به سرویسهای مطمئن و یا اتصال گرا نیاز باشد د رلایه بالاتر این نیاز تامین خواهد شد.

در این لایه مسیریابها بایستی از شرایط توپولوژیکی و ترافیکی شبکه اطلاعاتی را کسب نمایند تا مسیریابی بروش پویا انجام شود. همچنین در این لایه باید اطلاعاتی درباره مشکلات یا خطاهای احتمالی در ساختار زیرشبکه بین مسیریابها و ماشیهای میزبان، مبادله شود. یکی دیگر از وظائف این لایه ویژگی ارسال «چند پخشی»[2] است یعنی یک ایستگاه قادر باشد به چندین مقصد گوناگون که در قالب یک گروه سازماندهی شده اند، بسته یا بسته هائی را ارسال نماید.

لایهٔ سوم از مدل TCP/IP: لایهٔ انتقال

این لایه ارتباط ماشینهای انتهایی (ماشینهای میزبان) را در شبکه برقرار می کند یعنی می تواند بر اساس سرویسی که لایه دوم ارائه می کند یک ارتباط اتصال گرا و مطمئن[3]، برقرار کند. البته در این لایه برای عملیاتی نظیر ارسال صوت و تصویر که سرعت مهمتر از دقت و خطا است سرویسهای بدون اتصال سریع و نامطمئن نیز فراهم شده است.

---

[1] Ack/Nack
[2] Multicast
[3] Reliable

در سرویس مطمئنی که در این لایه ارائه می شود، مکانیزمی اتخاذ شده است که فرستنده از رسیدن نو یا عدم رسید صحیح بسته به مقصد با خبر شود. در مورد سرویسهای مطمئن و نامطمئن بعداً بحث خواهد شد. این لایه از یکطرف با لایهٔ شبکه و از طرف دیگر با لایهٔ کاربرد در ارتباط است. داده های تحویلی به این لایه توسط برنامهٔ کاربردی و با صدا زدن توابع سیستمی تعریف شده در «اواسط برنامه های کاربردی» (API[1]) ارسال یا دریافت می شوند.

لایهٔ چهارم از مدل TCP/IP: لایهٔ کاربرد

در این لایه بر اساس خدمات لایه های زیرین، سرویس سطح بالایی برای خلق برنامه های کاربردی ویژه و پیچیده ارائه می شود. این خدمات در قالب، پروتکلهای استانداردی همانند موارد زیر به کاربر ارائه می شود: شبیه سازی ترمینال[2]، انتقال فایل یا FTP، مدیریت پست الکترونیکی، خدمات انتقال صفحات ابرمتنی و دهها پروتکل کاربردی دیگر. در پایان این قسمت بایستی خاطر نشان کنیم که ارسال یک واحد اطلاعاتی از لایه چهارم پس از انجام پردازشهای لازم در لایه های زیرین به نحو مناسبی روی زیر شبکه تزریق شده و نهایتاً در ماشین مقصد، تحویل یک برنامه کاربردی خاص خواهد شد.

[1] Application Program Interface
[2] TEINET/Teminal Emulation

لایهٔ اینترنت (IP)

جوهرهٔ اینترنت به گونه ای شکل گرفته است که مجموعه ای از شبکه های خودمختار[1] را به همدیگر وصل می نماید. هیچگونه ساختار حقیقی و رقابتی نمی توان برای اینترنت متصور شد. این نکته را بایستی یادآور شویم که در قسمت «زیرشبکه» از شبکهٔ اینترنت تعدادی از خطوط ارتباطی با پهنای باند (نرخ ارسال) بسیار بالا و مسیریابهای بسیار سریع و هوشمند، برای پیکرهٔ شبکهٔ جهانی اینترنت یک «ستون فقرات»[2] تشکیل داده است. شبکه های منطقه ای و محلی پیرامون این ستون فقرات شکل گرفته و ترافیک دادهٔ آنها به نحوی از این ستون فقرات خواهد گذشت. ستون فقرات در شبکهٔ اینترنت که با سرمایه گذاری عظیمی در آمریکا، اروپا و قسمتهایی از اقیانوسیه و آسیا ایجاد شده است. حجم بسیار وسیعی از بسته های اطلاعاتی را در هر ثانیه حمل می کنند و اکثر شبکه های منطقه ای و محلی یا ارائه دهندگان سرویسهای اینترنت[3] به نحوی با یکی از گروه های این ستون فقرات در ارتباطند.

به گونه ای که در بخش قبلی اشاره شده قراردادی که حمل و تردد بسته های اطلاعاتی و همچنین مسیریابی صحیح آنها را از مبدأ به مقصد، مدیریت و سازماندهی می نماید پروتکل IP[4] نام دارد. در حقیقت پروتکل IP که روی تمام ماشینهای شبکه اینترنت وجود دارد بسته های اطلاعاتی را (بسته IP) از مبدإ تا مقصد هدایت می نماید، فارغ از

---

[1] Autonomous
[2] Backbone
[3] Internet Service Provider(ISP)
[4] Internet protocol

آنکه آیا ماشینهای مبدأ و مقصد روی یک شبکه هستند یا چندین شبکهٔ دیگر بین آنها واقع شده است.

ساده ترین تعریف برای پروتکل IP روی شبکهٔ اینترنت بصورت زیر خلاصه می شود:

لایهٔ IP یک واحد از داده ها را از لایهٔ بالاتر تحویل می گیرد؛ به این واحد اطلاعات معمولاً یک «دیتاگرام» گفته می شود.امکان دارد طول این دیتاگرام بزرگ باشد، در چنین موردی لایهٔ IP آنرا به واحدهای کوچکتری که هر کدام «قطعه»[1] نام دارد شکسته و با تشکیل یک بستهٔ IP به ازای هر قطعه، اطلاعات لازم برای طی مسیر در شبکه را به آنها اضافه می کند و سپس آنها را روی شبکه به جریان می اندازد؛ هر مسیریاب با بررسی و پردازش بسته ها، آنها را تا مقصد هدایت می کند. هر چند طول یک بسته IP می تواند حداکثر 64Kbyte باشد و لیکن در عمل عموماً طول یک بسته ها حدود ۱۵۰۰ بایت است. (این قضیه به دلیل آنست که اکثر شبکه های محلی دنیا اعم از Bus، حلقه، ستاره، طول فریمی نزدیک به یک تا چند کیلو بایت دارند) پروتکل IP مجبور است هنگام قطعه قطعه کردن یک دیتاگرام، برای کل آن یک شمارهٔ مشخصه و برای هر قطعه یک شمارهٔ ترتیب در نظر بگیرد تا آن دیتاگرام بتواند در مقصد برای تحویل به لایهٔ بالاتر یعنی لایهٔ انتقال بازسازی شود.

[1] Fragment

(مجدداً تأکید می کنیم که در این مبحث، دیتاگرام[1] یک واحد اطلاعات است که به صورت یک جا از لایهٔIP به لایهٔ انتقال تحویل داده می شود یا بالعکس لایه انتقال آنرا جهت ارسال روی شبکه به لایهٔIP تحویل داده و ممکن است شکسته شود).

در کنار پروتکل IP چندین پروتکل دیگر مثل RIP,RARP,ARP,ICMP و ... تعریف شده که پروتکل IP را در عملکرد بهتر، مسیریابی صحیح، مدیریت خطاهای احتمالی یا کشف آدرسهای ناشناخته کمک می کنند.

تواناییهایی که پروتکل IP چندین پروتکلهای جانبی آن عرضه می کنند این امکان را فراهم آورده است که تمام شبکه ها و ابزارهای شبکه ای (مثل ماشینهای میزبان، مسیریابها، پلها، و ...) فارغ از نوع ماشین و نوع سخت افزار و حتی با وجود تفاوت در سیستم عامل مورد استفادهٔ آنها، بتوانند بسته های IP را با یکدیگر مبادله کنند. پروتکل IP ساختاری استاندارد دارد و به هیچ سخت افزار یا سیستم عامل خاص وابسته نیست.

بعنوان اولین گام در شناخت پروتکل IP لازم است قالب یک بستهٔIP را کالبد شکافی کره و در گامهای بعدی چگونگی آدرس دهی ماشینها و انواع کلاسهای آدرس در شبکهٔ اینترنت را معرفی نموده و نهایتاً به روشهای مسیریابی و همچین تعریف پروتکلهای وابسته به IP بپردازیم.

[1] اصطلاح دیتاگرام در ادبیات شبکه های کامپیوتری به معنای متفاوت و در موارد متعدد استفاده شده است. لذا به مورد استفادهٔ آن دقت داشته باشید.

قالب یک بستهٔ IP

شکل (٤-٢) قالب یک بسته IP را نشان می دهد. یک بستهٔIP از دو قسمت سرایند و قسمت حمل داده تشکیل شده است. مجموعهٔ اطلاعاتی که در سرآیند بستهٔIP درج می شود توسط مسیریابها مورد استفاده و پردازش قرار می گیرد.

دقت کنید که برای تحلیل برخی از مکانیزمها و تاکتیکهای حمله، مجبور هستید با فیلدهای متعدد بستهٔ IP آشنا باشید؛ زیرا برخی از این فیلدها مورد سوء استفادهٔ نفوذگران قرار می گیرند. در فصل نهم یاد خواهید گرفت که هر گاه برخی از این فیلدها بصورت عمدی و حساب شده دستکاری شود، منجر به اختلال در ماشین نهایی خواهد شد.

شکل (٤-٢) قالب یک بستهٔ IP

فیلد Version: اولین فیلد در سرآیند یک بستهٔ IP که چهار بیت است نسخهٔ پروتکل IP که این بسته بر اساس آن سازمانهای و ارسال شده است را تعیین می کند. در حال حاضر تمام شبکه ها و مسیریابها از نسخهٔ شماره ٤ پروتکل IP پشتیبانی می کنند. اگر چه امروزه نسخهٔ شمارهٔ٦ پروتکل IP به نامهای IPng یا Ipv6 معرفی و در حال بررسی و نصب است ولیکن بسیاری از مسیریابها در شبکه های دنیا هنوز برای پذیرش این

پروتکل آمادگی ندارند و به نظر می رسد که تا سال ۲۰۰۵ نگاش جدید جهانی نشود. عددی که در حال حاضر در این فیلد قرار می گیرد ٤ یا $(0100)_B$ است.

فیلد IHL[1]: این فیلد هم چهار بیتی است و طول کل سرآیند بسته را بر مبنای کلمات ۳۲ بیتی مشخص می نماید. بعنوان مثال اگر در این فیلد عدد ۱۰ قرار گرفته باشد بدین معناست که کل سرآیند ۳۲۰ بیت معادل چهل بایت خواهد بود. اگر به ساختار یک بستهٔ IP دقت شود به غیر از فیلد Options که اختیاری است، وجود تمام فیلدهای سرآیند الزامی می باشد. در حقیقت این فیلد بعنوان یک اشاره گر مرز بین سرآیند و داده ها را مشخص می کند.

فیلد Type of service: این فیلد هشت بیتی است و توسط آن ماشین میزبان (یعنی ماشین تولید کنندهٔ بسته IP) از مجموعهٔ زیر شبکه (یعنی مجموعهٔ مسیریابهای بین راه تقاضای سرویس ویژه ای برای ارسال یک دیتاگرام می نماید. بعنوان مثال ممکن است یک ماشین میزبان بخواهد دیتاگرام صدا یا تصویر برای ماشین مقصد ارسال نماید؛ ر چنین شرایطی از زیرشبکه تقاضای ارسال سریع و به موقع اطلاعات را دارد نه قابلیت اطمینان صد در صد، چرا که اگر یک یا چند بیت از داد های ارسالی در سیر دچار خرابی شود تاثیر چندانی در کیفیت کار نخواهد گذاشت ولی اگر بسته های حاوی اطلاعات صدا یا تصویر به سرعت و سرموقع تحویل نشود اشکال عمده بوجود خواهد آمد. در چنین مواقعی ماشین میزبان از زیرشبکه تقاضای سرویس سریع (ولاجرم غیرقابل

[1] IP Header Length

اطمینان) می نماید. در برخی از محیط های دیگر مثل ارسال نامهٔ الکترونیکی یا مبادلهٔ فایل انتظار اطمینان صد درصد از زیرشبکه وجود دارد و سرعت تاثیر چندانی بر کیفیت کار ندارد. اکثر مسیریابهای تجاری فیلد Type of Service را نادیده می گیرند و اهمیتی به محتوای آن نمی دهند.

فیلد Total Length: در این فیلد ۱۶ بیتی عددی قرار می گیرد که طول کل بستهٔ IP را که شامل مجموع اندازه سرآیند و ناحیه داده است، تعیین می کند. مبنای طول بر حسب بایت است و بنابراین حداکثر طول کل بستهٔ IP می تواند ۶۵۵۳۵ بایت باشد.

فیلد Identification: همانگونه که قبلاً اشاره شد برخی از مواقع مسیریابیها یا ماشینهای میزبان مجبورند یک دیتاگرام را به قطعات کوچکتر بشکنند و ماشین مقصد مجبور است آنها را بازسازی کند، بنابراین وقتی یک دیتاگرام واحد شکسته می شود باید مشخصه ای داشته باشد تا در هنگام بازسازی آن در مقصد بتوان قطعه های آن دیتاگرام واحد در مشخص می کند. کلیهٔ بسته های IP که با این شماره وارد می شوند قطعه های مربوط به یک دیتاگرام بوده و باید پس از گردآوری قطعه ها، آن را مجدداً بازسازی کرد. بعنوان مثال اگر در این فیلد عدد ۱۶۵۲ قرار بگیرد تمام بسته های IP که مشخصهٔ ۱۶۵۲ دارند قطعه های مربوط به یک دیتاگرام هستند و پس از دریافت کل قطعه ها باید بازسازی شوند. البته برای حفظ ترتیب، هر قطعه گذشته از یک شمارهٔ مشخصه بایستی دارای شمارهٔ ترتیب نیز باشد تا بتوان آنها را طبق این شماره مرتب و بازسازی کرد.

فیلد Fragment Offset: این فیلد در سه بخش سازماندهی شده است:

الف) بیت DF[1]: با یک شدن این بیت در یک بستهٔIP هیچ مسیریابی حق ندارد آن را قطعه قطعه کند، چرا که مقصد قادر به بازسازی دیتاگرام های تکه تکه شده نیست. اگر این بیت به ۱ تنظیم شده باشد و مسیریاب نتواند آنرا به دلیل بزرگی اندازه، انتقال بدهد لاجرم حذف خواهد شد.

ب) بیت MF[2]: این بیت مشخص می کند که آیا بستهٔ IP آخرین قطعه از یک دیتاگرام محسوب می شود یا باز هم قطعه های بعدی وجود دارد. در آخرین قطعه از یک دیتاگرام بیت MF صفر خواهد بود و در بقیه الزاماً ۱ است.

ج) Fragment offset: این قسمت که سیزده بیتی است در حقیقت شماره ترتیب هر قعطه در یک دیتاگرام شکسته شده محسوب می شود. با توجه به سیزده بیتی بودن این فیلد، یک دیتاگرام حداکثر می تواند به ۸۱۹۲ تکه تقسیم شود. نکته بسیار مهم در مورد این فیلد آن است که اندازهٔ هر قطعه باید ضریبی از ۸ باشد یعنی به استنای قطعهٔ آخر، اندازهٔ بقیهٔ قطعه ها بایستی بگونه ای انتخاب شود که ضریبی از ۸ بایت باشد؛ مثلاً اگر در فیلد آفست مقدار ۷ قرار بگیرد نشان می دهد که محل قرارگرفتن قطعهٔ جاری در دیتاگرا بازسازی شده در موقعیت بایت پنجاه و ششم (۵۶=۸* ۷) خواهد بود. به عنوان مثالی دیگر فرض کنید مسیریابی مجبور است یک دیتاگرام به طول ۵۰۰ بایت را قطعه قطعه کند به گونه ای که اندازهٔ هر قطعهٔ زیر ۱۵۰۰ بایت باشد. در چنین موردی نمی تواند اندازهٔ هر قطعه را ۱۲۵۰ بایت در نظر بگیرد چرا که ضریبی از ۸ نیست ولی اندازهٔ

[1] Don’t Fragment
[2] More Fragment

۱۲۸۰ مناسب است. در این حالت مسیریاب، دیتاگرام را به سه بستهٔ ۱۲۸۰ بایتی و یک بستهٔ ۱۱۶۰ بایتی می شکند. در این مثال فرض کندی مسیریاب شمارهٔ۲۳۲۲ را به عنوان مشخصهٔ دیتاگرام انتخاب کرده است؛ بنابراین برای هر یک از چهار قطعهٔ دیتاگرام، فیلد آفست و مشخصه به صورت زیر خواهد بود.

شماره قطعه	Identification	Fragment Offiset	بیت MF	آدرس محل قرار گرفتن قطعه در دیتاگرام	طول هر قطعه
قطعه شماره۱	2322	0	1	8*0=0	۱۲۸۰
قطعه شماره۲	2322	160	1	8*160=1280	۱۲۸۰
قطعه شماره۳	2322	320	1	8*320=2560	۱۲۸۰
آخرین قطعه	2322	480	0	8*480=3840	۱۲۸۰

ممکن است یک دیتاگرام واحد از یک ماشین میزبان روی زیرشبکه تزریق شود و در طول مسیر به مسیریابی برسد که به دلیلی مجبور به شکستن آن به قطعات کوچکتر شود. عمل شکستن یک دیتاگرام ممکن است در هر جای زیرشبکه اتفاق بیفتد ولیکن عمل بازسازی فقط در ماشین مقصد انجام می شود.

در فصل هفتم با مکانیزمی آشنا می شوید که بر اساس آن نفوذگر سعی می کند تلاش خود برای حمله به یک سیستم را مخفی نگه دارد. این مکانیزم مبتنی بر بسته های قطعه قطعه شدهٔ IP است. (مبحث Frag Router را بخوانید.)

در فصل نهم (حملات Dos) خواهید دید که نفوذگر با دستکاری تعمدی در فیلد Fragment Offest حملاتی را برای اختلال در ماشین گیرندهٔ نهایی بسته، تدارک می بیند.

فیلد Time To Live: این فیلد هشت بیتی در نقش یک شمارندهٔ، طول عمر بسته را مشخص می کند. طول عمر یک بسته بطور ضمنی به زمانی اشاره می کند که یک بسته IP می تواند بر روی شبکهٔ سرگردان باشد. حداکثر طول عمر یک بسته، ۲۵۵ خواهد بود که به ازای عبور از هر مسیریاب[۱] از مقدار این فیلد یک واحد کم می شود. هر گاه یک بستهٔ IP به دلیل بافر شدن در حافظه یک مسیریاب زمانی را معطل بماند، به ازای هر ثانیه یک واحد از این فیلد کم خواهد شد. به محض آنکه مقدار این فیلد به صفر برسد بستهٔIP در هر نقطه از مسیر باشد حذف شده و از ادامه سیر آن به سمت مقصد جلوگیری خواهد شد. (البته معمولاً یک پیام هشدار به ماشینی که آن بسته را تولید کرده باز پس فرستاده خواهد شد.) اگر چه بزرگترین عددی که در فیلد طول عمر بسته قرار می گیرد ۲۵۵ است ولی در عمل مقداری که سیستمهای عامل در این فیلد قرار می دهند چیزی حدود ٦٤ است. (البته می توان مقدار پیش فرض آن را عوض کرد).

این فیلد برای پاکسازی زیرشبکه از بسته های IP که به هر دلیل در یک مسیر بسته می چرخند بسیار حیاتی است وگرنه پس از مدتی کل زیرشبکه از بسته های آشغال پرخواهد شد. بسته های سرگردان گاهاً به این دلیل بوجود می آیند که جداول مسیریابی

[۱] در ادبیات شبکه به عبور بسته از یک مسیریاب یک جهش یا Hop گفته می شود.

در بعضی از مسیریابها آلوده به اطلاعات نادرست[1] شده اند. سرگردانی یک بسته در زیرشبکه مسئلهٔ غیرممکنی نیست و گاهی اتفاق می افتد.

در فصل ششم با مجموعه ای از مکانیزمها و ابزار پویش (مثل traceroute , Cheops و Firewalk) آشنا می شوید که همگی به نحوی از فیلد TTL در بستهٔ IP استفاده کرده اند.

فیلد Protocol: دیتاگرامی که در فیلد داده از یک بستهٔ IP حمل می شود با ساختمان دادهٔ خاص از لایهٔ بالاتر تحویل پروتکل IP شده تا روی شبکه ارسال شود. بعنوان مثال ممکن است این داده ها را پروتکل TCP در لایه بالاتر ارسال کرده باشد و یا ممکن است این کار توسط پروتکل UDP انجام شده باشد. بنابراین مقدار این فیلد شمارهٔ پروتکلی است که در لایه بالاتر تقاضای ارسال یک دیتاگرام کرده است؛ بسته ها پس از دریافت در مقصد باید به پروتکل تعیین شده تحویل داده شود.

فیلد Header Checksum : این فیلد که شانزده بیتی است به منظور کشف خطاهای احتمالی در سرآیند هر بستهIP استفاده می شود. برای محاسبهٔ کد کشف خطا، کل سرایند بصورت دو بایت، دو بایت با یکدیگر جمع می شود. نهایتاً حاصل جمع به روش «مکمل یک»[2] منفی می شود و این عدد منفی در این فیلد از سرآیند قرار می گیرد. در هر مسیریاب قبل از پردازش و مسیریابی ابتدا صحت اطلاعات درون سرآیند بررسی می شود. دقت کنید که فیلد Checksum در هر مسیریاب باید از نو محاسبه و مقدار دهی

[1] Corrupt
[2] One's Complement

شود زیرا وقتی یک بستهٔ IP وارد یک مسیر می شود حداقل فیلد TTL از آن بسته عوض خواهد شد.

فیلد Sourrce Address: هر ماشین میزبان در شبکه اینترنت یک آدرس جهانی و یکتای ۳۲ بیتی دادر. بنابراین هر ماشین میزبان در هنگام تولید یک بسته IP باید آدرس خودش را در این فیلد قرار بدهد. بحث آدرسها در اینترنت یکی از مسائل بسیار مهمی است که در بخشی مجزا به آن خواهیم پرداخت. (به این آدرس از این به بعد، «آدرسIP» می گوییم.

فیلد Destination Address: در این فیلد آدرس ۳۲ بیتی مربوط به ماشین مقصد که باید بسته IP تحویل آن بشود، قرار می گیرد.

فیلد اختیاری Options: در این فیلد اختیاری می توان تا حداکثر ٤٠ بایت قرار داد. بدلیل بی اهمیت بودن این فیلد از توضیح در مورد آن پرهیز می کنیم.

فیلد Payload: در این فیلد داده های دریافتی از لایهٔ بالاتر قرار می گیرد.

مبحث آدرسها در اینترنت و اینترانت

همانگونه که در مباحث قبلی بدان اشاره کردیم پروتکل اینترنت در ارتباطات بین شبکه ای از آدرسهای منحصر به فرد و یکتای ۳۲ بیتی بهره می برد. (هر چند که در نسل بعدی پروتکل اینترنت که تا سال ۲۰۰۵ همه گیر خواهد شد این آدرسها ۱۲۸ بیتی می شوند.) هر ابزار شبکه اعم از ماشینهای میزبان، مسیریابها و چاپگرهای شبکه در اینترنت با یک آدرس IP شناسائی می شوند.

در ادامه این فصل باید موارد زیر را بررسی و مطالعه کنیم: قالب هر آدرس IP چگونه سازماندهی می شود؟ کلاسهای مختلف آدرس های IP به چه منظور و چگونه سازماندهی می شوند؟ چگونه آدرسهای IP به آدرسهای سخت افزاری لایه فیزیکی تبدیل خواهد شد و قراردادهای نمایش آدرسهای IP چگونه هستند؟ یک مسیریاب چگونه می تواند از یک آدرس چهاربایتی، محل دقیق یک ماشین را بین دهها میلیون ماشین متصل به شبکه پیدا نماید؟

آدرسهای IP درون یک عدد دودویی ۳۲ بیتی درج می شوند ولیکن برای سادگی نمایش به چهار قسمت هشت بیتی[1]تقسیم و بصورت چهار عدد دهدهی که با نقطه از هم جدا شده اند، نوشته می شوند؛ یعنی معادل دهدهی هر یک از بایتهای آدرس بصورت مجزا نوشته شده و هر عدد با یک علامت ● از دیگری تفکی می شود. بعنوان مثال آدرس زیر یک آدرس IP معتبر می باشد که در قالب چهار قسمت دهدهی نوشته شده است:

34.21.255.1

این آدرس بصورت زیر در فیلد آدرس از یک بستهٔ IP تنظیم می شود:

پرارزشترین بایت یعنی اولین بایت سمت چپ از آدرس IP، کلاس آدرس را مشخص می کند و از این رو دارای اهمیت ویژه ای است. ولی قبل از آنکه کلاسهای آدرس را تشریح نماییم باز هم روی این نکته تکیه می کنیم که وقتی یک ماشین میزبان به شبکه

[1]Octet

اینترنت متصل می شود بایستی آدرس IP آن منحصر به فرد و یکتا باشد. در حقیقت هر ماشین روی شبکه با یک آدرس یکتا هویت پیدا می کند.

برای اطمینان از یکتا بودن آدرسهای IP برای ارتباطات عمومی، مرکز InterNIC[1] کنترل و نظارت بر روی آدرس های IP را بر عهده گرفته است.

IANA[2] قدرت اجرائی برای اختصاص آدرسهای IP منحصر به فرد را فراهم کرده است. هرچند شبکه های خصوصی که به اینترنت وصل نیستند می توانند از آدرسهای IP دلخواه استفاده کنند ولی اگر این شبکه ها زمانی بخواهند به اینترنت وصل شوند دوگانگی آدرسهای غیریکتا و نهایتاً تناقض و اشکال در مسیریابی[3] رخ خواهد داد؛ به همین دلیل پیشنهاد شده است که حتی شبکه های خصوصی نیز برای اختصاص آدرس به ماشینهای میزبان از مرکز InterNIC مجوز بگیرند و از آدرسهای معتبر و اختصاصی استفاده کنند.

کلاسهای آدرس IP

از آنجا که TCP/IPبرای شبکه های با مقیاس بزرگ طراحی شده است لذا نمی توان انتظار داشت که فضای ۳۲ بیتی آدرس که حدود چهار میلیارد و سیصد میلیون (4,294,967,295) آدرس را در اختیار می گذارد، بدون هیچ نظم و سیاق خاص به ماشینهای شبکه اختصاص داده شود. این کار همانند آن خواهد بود که تمام آپارتمانها و

[1] Internet Network Information Center
[2] Internet Assighed Number Authority
[3] Conflict

منازل در کل جهان با شماره های ده رقمی مشخص شود بدون آنکه هیچ ضابطه ای در شماره گذاری آنها رعایت شده باشد. آنگاه منزلی با شماره ۱۰٦٥٤۳۲۳۹۰ چگونه پیدا می شود!

آدرسهای پستی ساختاری سلسله مراتبی به صورت زیر دارند، به گونه ای که هر منزل در هر کجای دنیا قابل آدرس دهی است و به راحتی پیدا می شود:

شماره/ کوچه/ خیابان/ ناحیه/ شهر/ کشور

فلسفهٔ کلاسهای آدرس IP به همین منظور است:

آدرس ماشین/ آدرس زیرشبکه/ آدرس شبکه

با توجه به آنکه اینترنت مجموعه ای از شبکه های متصل شده به هم می باشد، آدرس دادن به ماشینهای میزبان بهتر است ۳۲ بیت آدرس IP به قسمتهای زیر تقسیم شود:

الف) آدرس شبکه

ب) آدرس زیر شبکه (در صورت لزوم)

ج) آدرس ماشین میزبان

آدرسهای IP در پنج کلاس E,D,C,B,A معرفی شده اند که شما بایستی آنها را بدقت بشناسید و تحلیل کنید. در زیر قالب کلاسهای پنج گانهٔ آدرس IP مشخص شده است:

آدرسهای کلاس A: قالب ۳۲ بیتی آدرس در کلاس A به صورت زیر است:

در کلاس A، پرارزشترین بیت از آدرس، مقدار صفر دارد و این بیت، کلاس A را از دیگر کلاسها متمایز می کند؛ ۷ بیت بعدی «مشخصه آدرس شبکه» و سه بایت باقیمانده،

آدرس ماشین میزبان را تعیین می کند. بنابراین در کلاس A بایت پرارزش در محدوده صفر تا ۱۲۷ تغیی رمی کند. چون تا ۲٤ بیت می توان حدود هفده میلیون ماشین میزبان را آدرس دهی کرد، می توان به این نتیجه رسید که آدرسهای کلاس A بایستی برای آژانسهای ستون فرات اینتنت یا شبکه ها بسیار عظیم مثل NSFNet یا ARPANet اختصاص داده شده باشد. مشخصهٔ شبکه در این کلاس بهیچوجه نمی تواند اعداد صفر یا ۱۲۷ انتخاب شود چرا که این دو عدد در شبکه معنای دیگری خواهند داشت و بعداً به آن اشاره خواهیم کرد. بنابراین تعداد شبکه هائی که در جهان می توانند از کلاس A استفاده کنند ۱۲٦ تا خواهد شد که بسیار کم است. امروزه اختصاص آدرسهای کلاس A غیرممکن است چرا که همهٔ آنها توسط پیشگامان شبکه سالها قبل تملیک شده اند.

وقتی به یک آدرس IP که در قالب دهدهی نوشته شده است نگاه می کنید براحتی می توانید کلاس آنرا تشخیص بدهید. اگر عدد سمت چپ آدرس، بین صفر تا ۱۲۷ باشد، آن آدرس از کلاسA خواهد بود:

آدرسهای کلاس B: قالب ۳۲ بیتی آدرس در کلاس B به صورت زیر است:

هر گاه دو بیت پرارزش از آدرس IP مقدار 10 داشته باشد آن آدرس از کلاس b خواهد بود. ۱٤ بیت باقیمانده از ۲ بایت سمت چپ، آدرس شبکه را تعیین می کند و دو بایت اول از سمت راست (۱٦ بیت آدرس ماشین میزبان خواهد بود. در آدرسهای کلاس B، تعداد ۱٦۳۸۲ (2^{14}-2) شبکهٔ گوناگون قابل تعریف خواهد بود و هر شبکه می تواند

٦٥٥٣٤(2^{16}-2) ماشین میزبان تعریف نماید. اختصاص آدرسهای کلاس B برای شبکه های بسیار عظیم مناسب است. امرزوه عملاً نمی توان آدرس کلاس B گرفت چرا که تقریباً همهٔ آنها آن تخصیص داده شده اند. اگر آدرس IP به صورت دهدهی نوشته شود و عدد سمت چپ آن بیش ۱۲۸ تا ۱۹۱ باشد، آن آدرس کلاس B خواهد بود:

کلاس C مناسب ترین و پرکارترین کلاس از آدرس های IP است. همانگونه که از شکل مشخص است در این کلاس، سه بیت پرارزش دارای مقدار 110 است و ۲۱ بیت بعدی از سه بایت سمت چپ برای تعیین آدرس شبکهٔ مورد نظر بکار رفته است. بنابراین در این کلاس می توان حدود دو میلیون شبکه را در جهان آدرس دهی کرد و هر شبکه می تواند تا ۲٥٤ عدد ماشین میزبان تعریف نماید. برای تشخیص آدرسهای کلاس C به عدد سمت چپ از آدرس IP که به صورت دهدهی نوشته شده است نگاه کنید. اگر عدد بین ۱۹۲ تا ۲۲۳ بود آن آدرس از کلاس C خواهد بود:

در این کلاس، چهار بیت پرارزش داری مقدار 1110 است و ۲۸ بیت باقیمانده از کل آدرس برای تعیین آدرسهای «چند مقصده»[1] (آدرسهای گروهی) است. از این آدرسها برای ارسال یک دیتاگرام به طور همزمان برای چندین میزبان کاربرد دارد و بمنظور عملیات رسانه ای و چند بخشی بکار می رود.

[1] Multicast

آدرس کلاس E: فعلاً این دسته از آدرسها که پنج بیت پرارزش آنها در سمت چپ 11110 است کاربرد خاصی ندارند و برای استفاده در آینده بدون استفاده رها شده اند. البته گاهی بصورت آزمایشی از این آدرسها استفاده شد ولی تاکنون جهانی نشده اند.

آدرسهای خاص

در بین تمام کلاسهای آدرس IP پنج گروه از آدرسها، معنای ویژه ای دارند و با آنها نمی توان یک شبکهٔ خاص را تعریف و آدرس دهی کرد. این پنج گروه آدرس عبارتند از:

الف) آدرس 0.0.0.0 : هر ماشین میزبان که از آدرس IP خودش مطلع نیست این آدرس را بعنوان آدرس خودش فرض می کند. البته از این آدرس فقط به عنوان آدرس مبداء و برای ارسال یک بسته می توان استفاده کرد و برندهٔ بسته نمی تواند پاسخی به مبداء بسته برگرداند.[1]

ب) آدرس 0.HostID: این آدرس زمانی به کار می رود که ماشین میزبان، آدرس مشخصهٔ شبکه ای که بدان متعلق است را نداند. در این حالت در قسمت NetID مقدار صفر و در قسمت HostID شمارهٔ مشخصهٔ ماشین خود را قرار می دهد.

ج) آدرس 255.255.255.255: برای ارسال پیامهای فراگیر برای تمام ماشینهای میزبان بر روی شبکهٔ محلی که ماشین ارسال کننده به آن متعلق است.

---

[1] استفاده از این آدرس مانند آنست که در آدرس فرستندهٔ یک بسته پستی نوشده شود: «خودم». بسته می تواند به مقصد برسد ولی پاسخی نخواهد داشت.

د) آدرس NerID.255: برای ارسال پیامهای فراگیر برای تمام ماشینهای یک شبکهٔ راه دور که ماشین میزبان فعلی متعلق به آن نیست. آدرس شبکهٔ مورد نظر در قسمت NetID تعیین شده و تمام بیتهای قسمت مشخصهٔ ماشین میزبان ۱ قرار داده می شود. البته بسیار از مسیریابها برای مصون ماندن شبکه از مزاحمتهای بیرونی، چنین بسته هایی را حذف می کنند.

ه) آدرس 127 x.x.yy.zz: این آدرس بعنوان «آدرس بازگشت» شناخته می شود و آدرس بسیار مفیدی برای اشکال زدایی از نرم افزار می باشد. به عنوان مثال اگر بسته ای به آدرس 127.0.0.1 اراس شود، بسته برای ماشین تولید کنندهٔ آن بر خواهد گشت؛[1] در این حالت اگر نزم افزارهای TCP/IP درست و بدون اشکال نصب شده باشد فرستندهٔ بسته باید آنرا مجدداً دریافت کند. همچنین از این آدرس می توان برای آزمایش برنامه های تحت شبکه، نصب آنها بر روی ماشینهای میزبان استفاده کرد.

آدرسهای زیرشبکه

در ادامهٔ بحث بایستی مسئله زیرشبکه را در خصوص آدرس دهی ها مطرح نمائیم. مبحث را با یک مثال آغاز می نمائیم:

فرض کنید دانشگاه توانایی آدرس دهی ۲۵۵ ایستگاه را در شبکه دارد. در نظر بگیرید که دانشگاه دارای یک شبکه محلی واحد و یکپارچه برای کل دانشگاه نیست بلکه دارای

[1] این آدرس همانند آنست که فرستندهٔ یك بستهٔ پستی آدرس دقیق خودش را به عنوان گیرندهٔ آن درج نماید. بنابراین با آدرس 0.0.0.0 تفاوت ذاتی دارد.

هشت شبکهٔ محلی مجزا است که برای هر دانشکده تهیه دیده شده است. هر کدام از این شبکه ها که می تواند توپولوژی متفاوتی داشته باشد، از طریق مسیرباب به هم متصل شده اند و طبعاً برای ارتباط بین شبکه های هر دانشکده باید مسیربابی صورت گیرد. از دیدگاه بیرونی کل مجموعهٔ شبکه های محلی دانشگاه با یک آدرس مشخصه یعنی 211.11.121.0 شناخته می شود و مسیریابهای بیرونی هیچ دانشگاه یک زیرشبکه نامیده می شود.) بنابراین باید روشی وجود داشته باشد تا از طریق آدرسهای کلاس C (یا هر کلاس دیگر) بتوان زیرشبکه ها را نیز مشخص کرد تا مسیریابهای داخلی نیز قادر باشند زیر شبکه های مختلف را شناسایی و تفکیک کنند.

این مسئله برای آدرسهای کلاس B و A بسیار ضروری و اجتناب ناپذیر می نماید چرا که نمی توان انتظار داشت که یک موسسه که آدرس کلاس B با قابلیت تعریف حدود ٦٦ هزار ماشین میزبان ثبت کرده است فقط یک شبکهٔ یکپارچه داشه باشد بلکه چنین موسسه ای ممکن است دارای صدها زیر شبکهٔ کوچک و بزرگ باشد.

برای آنکه بتوان زیرشبکه ها (Sunetworks) را تفکیک کرد جدای از قسمت آدرس شبکه که کل شبکهٔ دانشگاه شما را مشخص می کند بایستی در قسمت مشخصهٔ ماشین میزبان نیز به گونه ای زیر شبکه ها مشخص شوند. این کار از طریق مفهومی به نام «الگوی زیر شبکه یا Subnet Mask» انجام می شود.

شما بانگاه اول به اولین عدد سمت چپ متوجه خواهید شد که این آدرس از چه کلاسی است ولی هنوز موارد مبهمی وجود دارد: آیا شبکه ای که آدرس آنرا پیش رو دارید فقط

یک شبکه است یا خودش زیرشبکه بندی شده است؛ یعنی از چند شبکهٔ محلی متصل بهم تشکیل شده است؟

این اطلاعات برای شبکه های مبتنی بر TCP/IP که قابلیت مسیریابی دارند بسیار مهم است، چرا که هر ماشین میزبان بایستی قادر به درک این مطلب باشد که آیا یک ماشین مقصد با آدرس خاص و مشخص، بر روی شبکهٔ محلی خودش واقع است یا آنکه آن آدرس متعلق به زیر شبکه دیگری است. بر اساس این اطلاعات ماشین میزبان تصمیم می گیرد که آیا انتقال اطلاعات باید مستقیماً بر روی شبکهٔ محلی انجام شود یا آنکه باید از طریق یک مسیریاب روی شبکه ای دیگر ارسال شود.

تمام ماشینهای میزبان برای تشخیص محل مقصد یک بستهٔ IP در شبکه احتیاج به یک مشخصهٔ دیگر دارند وآن «الگویزیر شبکه» نامیده می شود.

الگوی زیر شبکه یک عدد ۳۲ بیتی دودویی است که برای ماشین میزبان نقش یک مقایسه گر را بازی می کند تا با استفاده از آن بتواند تشخیص دهد که آیا مقصد روی همین شبکه محلی است که خودش به آن تعلق دارد یا روی شبکه دیگری است.

فرآیند استفاده از «الگوی زیر شبکه» را با استفاده از مثال قبل ولی با آدرس کلاس B آموزش می دهیم: فرض کنید شما کاربردی روی یک ایستگاه در شبکه دانشگاه خودتان هستید، آدرس IP متعلق به ماشین شما بصورت زیر اختصاص داده شده است:

131.55.213.73

با یک نگاه متوجه می شوید که آدرس از کلاس B است که مشخصهٔ شبکهٔ آن معادل 131.55.0.0 و مشخصهٔ ماشین شما 0.0.213.73 است: ولی هنوز نمی دانید شبکه ای که مشخصهٔ آن معادل 131.55 است آیا زیر شبکه دارد یا خیر؟

فرض کنید که دانشگاه شما با آدرس شبکه 131.55.0.0، می خواهد حداکثر دارای ۲۵٤ زیرشبکه باشد ، بهمین دلیل فرض کرده است که در فیلد مشخصهٔ ماشین میزبان (Host ID) که در کلاس B دو بایت سمت راست را شامل می شود، بابت دوم آن به عنوان مشخصهٔ مربوط به زیر شبکه تعریف شود یعنی فیلد دوبایتی مربوط به مشخصهٔ ماشین میزبان به دو بخش تقسیم شده است:

الف) مشخصهٔ زیر شبکه

ب) مشخصهٔ ماشین میزبان

ماشین شما تصمیم دارد بسته ای را برای ماشین میزبان با آدرس IP معادل 131.55.108.75 بفرستد؛ ماشین از کجا می تواند بفهمد که مقصد روی همین شبکه محلی که شما بدان متعلق هستید واقع است یا آنکه به شبکهٔ محلی در یک دانشکدهٔ دیگر متعلق است . دانست این موضوع بسیار با اهمیت خواهد بود چرا که اگر ماشین میزبان مورد نظر روی شبکهٔ دیگری باشد بسته باید با آدرس فیزیکی «مسیریاب پیش

فرض»[1] روی کانال ارسال شود. بنابراین تمام ماشینهای روی شبکه بایستی از وضعیت زیرشبکه ها مطلع باشند.

با توجه به آنچه که در بالا اشاره شد دومین بایت از سمت راست بعنوان مشخصهٔ زیر شبکه آن اختصاص داده شده است و بهمین دلیل هر ماشین برای دانستن آنکه آیا ماشین مقصد در شبکه محلی خودش واقع است یا در خارج از شبکه قرار دارد باید قسمت «مشخصهٔ شبکهٔ» و «مشخصهٔ زیرشبکه» از آدرس IP خودش را با همین مشخصه ها از آدرس مقصد مقایسه نماید. اینجاست که یک الگوی ۳۲ بیتی تعریف می شود که یک عدد ۳۲ بیتی و در این مثال بصورت 25.255.255.0 است:

هر گاه ماشین بخواهد یک آدرس IP را تحلیل کند. الگوی فوق را با آدرس IP خودش AND می کند (با اینکار در حقیقت Host ID خودش را صفر می نماید) سپس مجدداً الگو را با آدرس IP مقصد AND می کند (مشخصهٔ ماشین مقصد هم صفر می شود) حال نتیجه دو مرحله را با هم مقایسه می نماید. اگر نتیجه دو مرحله یکسان بود، هم مشخصهٔ شبکهٔ محلی قرار دارند. در صورت عدم تساوی، ماشین مبدأ به این نتیجه می رسد که مقصد مورد نظر روی شبکه محلی خودش نیست و آن بسته بایستی به آدرس فیزیکی مسیریاب پیش فرض ارسال شود.

---

[1] Default Gateway

پروتکل ICMP[1]

پروتکل IP، پروتکلی «بدون اتصال»[2] و «غیرقابل اعتماد»[3] است! بدون اتصال بدین معنا که مسیریاب هر بسته را بدون هیچگونه هماهنگی با مقصد بسته یا مسیریاب بعدی ارسال می نماید، بدون آنکه بتواند اطلاعی از وجود یا عدم وجود مقصد داشته باشد. در ضمن هر مسیریاب پس از ارسال یک بسته آن را فراموش می کند و منتظر «پیام دریافت بسته»[4] از گیرندهٔ آن نخواهد ماند. اگر یک بستهٔ IP با خطا به مقصد برسد و یا اصلاً به مقصد نرسد این پروتکل هیچ اطلاعی در مورد سرنوشت آن به فرستندهٔ بسته نمی دهد. دلایل مختلفی برای نرسیدن یک بسته به مقصد وجود دارد: ممکن است «زمان حیات» بسته قبل از رسیدن به مقصد منقضی شود؛ ممکن است مسیریاب بسته را به مسیری اشتباه هدایت کند؛ ممکن است در هنگام قطعه قطعه کردن بسته و ارسال آنها، یکی از قطعات دچار خطا شود یا به هر دلیلی به مقصد نرسد بنابراین کل دیتاگرام قابل بازسازی نخواهد بود؛ ممکن است مقصد بسته آمادگی دریافت بسته را نداشته باشد یا اصلاً وجود خارجی نداشته باشد. در هنگام بروز هر گونه خطا، پروتکل IP به فرستندهٔ بسته هیچ اطلاعی در مورد سرنوشت آن نخواهد داد.

عدم گزارش خطا به تولید کنندهٔ یک بسته منجر به تکرار خطا و حمل بیهوده و زائد بسته هایی می شود که محکوم به فنا و حذف در شبکه هستند. به عنوان مثال عدم

---

[1] Internet Control Message Protocol
[2] Connectionless
[3] Unreliable
[4] Acknowledgement Message

گزارش در مورد آماده نبودن مقصد برای دریافت بسته باعث خواهد شد که فرستندهٔ آن اقدام به ارسال بسته های دیگر کند در حالی که این کار بی ثمر خواهد بود و فقط بار ترافیک شبکه را افزایش می دهد و حتی می تواند منجر به بروز «ازدحام» شود.

پروتکل ICMP در کنار پروتکل IP، برای بررسی انواع خطا و ارسال پیام برای مبدأ بسته در هنگام بروز اشکالات ناخواسته استفاده می شود. در حقیقت ICMP یک سیستم گزارش خطا است که بر روی پروتکل IP نصب می شود تا در صورت بروز هر گونه خطا به فرستندهٔ بسته پیام مناسب را بدهد تا آن خطا تکرار نشود. در واقع ICMP وظیفه ای در قبال وقوع خطا ندارد بلکه فقط پیامی که بیانگر بروز خطا و نوع آن است به فرستندهٔ بر می گرداند. این پروتکل اشکالات موجود را در قالب یکسری پیام گزارش می کند که این پیام خود در یک بستهٔ IP قرار می گیرد که از جانب یک مسیریاب یا ماشین مقصد به آدرس فرستنده باز می گردد. در زیر چگونگی قرار گرفتن یک پیام ICMP درون یک بستهٔIP نشان داده شده است:

با توجه به آنکه پیام ICMP خود درون یک بسته IP جاسازی می شود بنابراین فیلد Protocol در سرآیند بستهٔ IP باید با شمارهٔ مشخصهٔ پروتکل ICMP (یعنی ۱) تنظیم شود. دقت کنید که خود بسته های ICMP نیز ممکن است دچار خطا شوند که برای این گونه خطا پیامی ارسال نخواهد شد. شکل کلی و قالب پیام ICMP در زیر مشخص شده است:

فیلد TYPe: در این فیلد عددی قرار می گیرد که بیانگر نوع پیام می باشد و ساختار فیلدهای Parameters و Data بسته به عددی که در این فیلد قرار می گیرد متفاوت خواهد بود.

فیلد Code: گاهی خود نوع پیام به چند «زیر نوع» دیگر تقسیم می شود که کد «زیر نوع» در این فیلد قرار می گیرد.

فیلد Checksum: محتوای این فیلد برای سنجش اعتبار و سلامت بستهٔ ICMP مورد استفاده قرار می گیرد. تمام بستهٔ ICMP بصورت دوبایت دوبایت جمع شده و نهایتاً از مکمل ۱ حاصل جمع، عددی ۱۶ بیتی بدست می آید که درون این فیلد قرار می گیرد.

در ادامه نوع و ساختار پیامهای ICMP را توضیح می دهیم:

پیام Destination Unreachable: این پیام زمانی صادر می شود که زیر شبکه یا یک مسیریاب نتواند آدرس مقصد را تشخیص بدهد یا به هر دلیلی بسته توسط ماشین میزبان تحویل گرفته نشود. (مثلاً بدلیل بزرگ بودن اندازهٔ بسته ها و عدم اجازه به مسیریاب برای شکستن آن)

پیام Time Exceeded: این پیام زمانی صادر می شود که مهلت قانونی یک بسته منقضی شده باشد و یک مسیریاب مجبور شود آنرا حذف کند؛ در چنین حالتی این پیام به آدرس فرستنده بسته IP برای آگاهی ارسال خواهد شد.

پیام Parameter Problem: این پیام زمانی صادر خواهد شد که مقداری نامعتبر در یکی از فیلدهای سرآیند در بستهIP قرار گرفته باشد و مسیریاب قادر به تشخیص و

تفسیر سرآیند این بسته IP نباشد. بعنوان مثال در فیلد Version از بسته IP عدد ۵ قرار گرفته باشد و یا Checksum با سرآیند تناقض داشته باشد.

پیام Source Quench: این بسته زمانی برای یک ماشین میزبان ارسال می شود که از آن خواسته شود حجم ارسال بسته هایش را کاهش بدهد چرا که در غیر اینصورت ازدحام پیش خواهد آمد. در مجموع هرگاه از یک ماشین میزبان پس از طی مدت مشخصی این پیام را دریافت نکرد می تواند سرعت تولید بسته ها را به حالت اول برگرداند.

پیام Redirect: این پیام زمنی صادر می شود که یک مسیریاب احساس کند بسته یا بسته هایی که برای او ارسال شده است در مسیر صحیح نیستند و احتمالاً اشکالی در مسیریابی وجود دارد. این پیام می تواند برای هشدار خطاهای احتمالی موثر باشد. فرض کنید به مسیریاب R1 بسته ای ارسال شده و او با بررسی جدول مسیریابی آنرا به مسیریاب R2 بسته ای ارسال شده و او با بررسی جدول مسیریابی آنرا به مسیریاب R2 فرستاده تا او آنرا به مقصد X برساند. حال اگر R2 با مقایسهٔ الگوی زیرشبکه به این نتیجه رسید که خود او و فرستنده آن بسته در یک شبکه واقعند با ارسال این پیام به فرستنده اعلام میکند اگر از این به بعد بسته هایش به جای اینکه به R1 ارسال شود به R2 داده شود، زودتر به مقصد خواهد رسید؛ ضمناً آدرس IP خودش را نیز در فیلد Gateway Internet Address قرار می دهد.

پیامهای Echo Request, Echo Reply : پیام Echo Request وقتی صادر می شود که یک مسیریاب بخواهد بداند آیا ماشین خاص شبکه قابل دسترس و موجود است یا خیر. در پاسخ به دریافت Echo Request، مقصد با ارسال پیام Echo Reply به آن پاسخ مـی دهد. با این پرسش و پاسخ، یک ماشین می تواند از قابل دسترس بودن یک مسیریاب یا ماشین میزبان در شبکه مطلع شود.

بدلیل اهمیت بسیار ویژهٔ و حساس این دو پیام در تحلیل برخی از حملات، ساختار بستهٔ آنها را معرفی می کنیم:

8: برای مشخص کردن پیام Echo Request 0: برای مشخص کردن پیام echo Reply

ابتدا پیام Echo Request به سمت ماشین مقصد ارسال می شود. ماشینی که آنرا دریافت کند، آدرسهای مبدا و مقصد را عوض کرده و Type آنرا از ۸ به صفر تغییر داده، پس از محاسبهٔ مجدد کد کشف خطا، آنراباز می گرداند. فیلدهای Identifier و Sequence Number برای پیشگیری از اشتباه در همخوانی و تطاق پیامهای رفت و برگشت است تا مبدأ بداند یک پاسخ مربوط به کدام تقاضای اوست. به فرآیند رفت بستهٔ Echo و بازگشت پاسخ، عمل Ping گفته می شود و کاربرد زیادی دارد.

پیامهای Timestamp Reply و Timestamo Request: این دو پیام دقیقاً شبیه دو پیام تعریف شده در قبل هستند با این تفاوت که دریافت کنندهٔ آن، زمان دریافت و زمان ارسال بسته را نیز در پاسخ به آن اضافه خواهد کرد. بنابراین ارسال کننده پیام Timestamp Request پس از دریافت پاسخ نه تنها از قابل دسترس بودن مقصد با خبر می شود بلکه زمان رفت و برگشت یک بسته را نیز می تواند تخمین بزند و به کمک آن جداول مسیریابی و همچنین کارآئی شبکه را اندازه گیری نماید.

در پروتکل ICMP چهار پیام دیگر نیز وجود دارد که با استفاده از آنها یک ماشین میزبان می تواند آدرس IP شبکه محلی خود را در هنگامی که چندین شبکه محلی از

آدرسهای IP مشترک استفاده می کند پیدا نماید. برای بدست آوردن اطلاعات جزئی تر و دقیق در مورد وظایف و پیامهای پروتکل ICMP به RFC-792 مراجعه نمائید.

پروتکل ARP[1]

نکته ظریفی که در مورد شبکه اینترنت وجود دارد آن است که اگر چه تمام ماشینهای میزبان و ابزارهای شبکه ای از آدرس IP منحصر به فرد و یکتا است استفاده می کنند ولیکن یک بسته IP فقط در لایه شبکه قابل شناسائی و تحلیل است. یک بسته IP قبل از ارسال روی کانال از لایه اول یعنی لایهٔ فیزیکی عبور می کند و ضمن اضافه شدن اطلاعات لازم و تشکیل یک فریم، روی کانال فیزیکی ارسال می شود. بعبارت روشنتر بسته IP قبل از ارسال درون فیلد داده از فریمی قرار می گیرد که بعداً در لایه اول تشکیل می شود؛ لایه اول وظیفه ای در قبال مسیریابی و کاهائی از این قبیل ندارد و فقط با آدرسهای فیزیکی کار می کند. بعنوان مثال اگر ماشین شما بخواهد بسته ای را برای ماشین که روی شبکه محلی خودتان واقع است بفرستد، در لایه اول الزاماً بایستی آدرس فیزیکی ماشین شما (مبداء) و آدرس فیزیکی ماشین طرف مقابل (مقصد) معین باشد. (این آدرسها بصورت سخت افزاری در کارت شبکه درج شده است) عدم دانستن آدرسهای فیریکی عملاً مساوی عدم توانایی برای ارتباط خواهد بود چرا که روی کانال انتقال آدرسهای IP بی معنا هستند.

[1] Address Resolution Protocol

حال فرض کنید ماشین شما می خواهد بسته ای را برای ماشین دیگر ارسال کند که روی شبکه فعلی شما نیست. در این حالت هم لایه اول یک فریم برای ارسال روی کانال فیزیکی تشکیل می دهد و نیاز به آدرس MAC از مقصد دارد؛ آدرس فیزیکی مقصد چیست؟

در لایه اول هر گاه بسته ای قرار است به خارج از شبکه ارسال شود آدرس فیریکی مقصد، آدرس مسیریاب پیش فرض شما خواهد بود. بنابراین آدرسهای MAN مقوله ای جدا هستند و آدرسهای IP مقوله ای دیگر!

هر ماشینی در اینترنت، گذشته از آن که بایستی آدرسهای IP خودش و مقصدش را بشناسد، نیازمند به دانستن آدرسهای فیزیکی ماشینهائی که مستقیماً با او در ارتباطند، هم هست. بعنوان مثال شبکه اترنت که در تمام دنیا شناخته شده است از آدرسهائی استفاده می کند که منحصر به فرد و ٤٨ بیتی (٦ بایتی) است. بنابراین کامپیوتری که به یک کارت اترنت مجهز است گذشته از آن که بایستی یک آدرس IP منحصر به فرد داشته باشد یقیناً دارای یک آدرس ٤٨ بیتی یکتاست که این آدرس یکتا در کارخانه سازندهٔ آن تنظیم شده است. بنابراین وقتی پروتکل IP می خواهد یک بسته اطلاعاتی را روی شبکه بفرستد باید به نحوی آدرس فیزیکی اول ماشینی که با آن بایستی ارتباط برقرار کند را بداند؛ این ماشین می تواند مسیریاب پیش فرض او باشد یا می تواند آدرس فیزیکی مقصد روی همین شبکه محلی باشد.

حال فرض کنید ایستگاهی آدرس IP ماشینی را که می خواهد با آن ارتباط برقرار کند، می داند ولی آدرس فیزیکی او را نمی داند. چه کاری می تواند انجام بدهد؟ باید از پروتکل ARP بهره ببرد! در این پروتکل فرض بر آن است که تمام ماشینهای روی یک شبکه محلی آدرس IP خود را می دانند.

برای روشن شدن وظیفهٔ پروتکل ARP به شکل (۵-۲) نگاه کنید. در این شکل فرض کنید سه شبکه در دانشگاه شما نصب شده است. شبکهٔ محلی اول در دانشکدهٔ کامپیوتر با آدرس کلاس، به شمارهٔ 192.31.65.0 و شبکهٔ دوم در دانشکده برق با آدرس کلاس C به شمارهٔ 192.31.60.0 به همدیگر متصل شده اند. هر ماشین در شبکهٔ اترنت یک آدرس ٤٨ بیتی یکتا دارد. مسیریابها در شکل مشخص شده اند و ارتباط دو شبکه اترنت را با FDDI برقرار می کنند. شبکهٔ FDDI از طریق یک خط اختصاصی به شبکهٔ جهانی اینترنت متصل شده است . هر مسیریاب به دو شبکهٔ متفاوت متصل شده و به عنوان عضوی از هر دو شبکه دارای دو آدرس IP مجزا می باشد، که هر یک از آنها در یکی از شبکه های محلی تعریف شده است.

حال فرض کنیدکه ماشینی مایل است به آدرس خاصی مثلاً 192.31.65.5 بستهٔ IP بفرستد. در لایهٔ شبکه یک بستهٔ IP با مشخصات لازم ساخته می شود و در قسمت آدرس مقصد مقدار 192.31.65.5 قرار می گیرد. از دیدگاه لایه شبکه پس از تشکیل بستهٔIP، کار تمام است ولیکن از دیدگاه لایه اول که بایستی آن بسته را روی کانال ارسال

کنـد دانسـتن آدرس فیزیکـی (آدرس MAC) ماشـین مقصـدی کـه آدرس IP آن 192.31.65.5 است، حیاتی است.

وظیفه پروتکل ARP در اینجا آن است که یک «بسته فراگیـر»[1] روی کـل شـبکه محلـی منتشر کند که این بسته در حقیقت سوال می کند:

«کسی که آدرس IP او 192.31.65.5 است، آدرس فیزیکی او چیست؟»

با توجه به آنها بسته های فراگیر توسط تمام ماشینهای روی شـبکهٔ محلـی دریافـت مـی شود، ماشینی که آدرس IP خودش را درون این بسته می بیند، بـدان پاسـخ مـی دهـد و آدرس فیزیکی خود را برای ارسال کنندهٔ آن بسته می فرستد. پس از آنکه آدرس فیزیکی مقصد بدست آمد، فریم اترنت ساخته شده بر وری کانال منتقل می شود.

به این نکته توجه داشته باشید که هـر ماشـین بـر روی شـبکهٔ محلـی از پروتکـل ARP حمایت می کند و این پروتکل عملیات پرسش و پاسخ را برای هر ماشـین کـه تقاضـای ارسال بستهٔ IP دارد، انجام می دهد. (فعال بـودن ARP بـر روی تمـام ماشـینهای شـبکهٔ محلی الزامی است.)

برخلاف پروتتکل ICMP که روی پروتکل ARP مستقیماً بر روی لایهٔ فیزیکی عمل می کند؛ یعنی پس از آنکه بستهٔ ARP ساخته شد، درون فیلد داده از فریم لایهٔ فیزیکی قـرار گرفته و روی کانال ارسال می شود. در شکل (٦-٢) چگـونگی سـاخته شـده یـک پیـام

[1] Broadcast

ARP به تصویر کشیده شده است . در شکل (۷-۲) ساختار درونی بستهٔARP تشریح شده است.

Hardware Type	
Protoco Type	
Hardware Address Length	Protocol Address Length
Operation Code	
Source Hardware Address	
Source IP Address	
Destination Hardware Address	
Destination IP Address	

شکل (۷-۲) ساختار بسته های ARP

Hardware Type: شمارهٔ مشخصهٔ نوع سخت افزار کارت شبکه که در لایهٔ اول وظیفهٔ انتقال اطلاعات روی کانال فیزیکی را برعهده دارد. این شماره ها در جدول (۱۳-۳) مشخص شده اند.

Protocol Type: نوع پروتکلی که در لایهٔدوم از آن استفاده می شود. برای شبکه های مبتنی بر TCP/IP این شماره ۲۰٤۸ است.

Hardware Address Length: طـول آدرسـهای IP کـه در پروتکـل TCP/IP مقـدار ٤ است.

Operation Coke (Opcode): ۱ برای ARP request و ۲ برای ARP Reply

Source Hardware Address: آدرس فیزیکی مبدأ

Source IP Address: آدرس IP ماشین مبدأ

Destination Hardware Address: آدرس فیزیکی ماشین مقصد

Destination Ip Address: آدرس IP ماشین مقصد

برای بالا بردن سرعت پروتکل ARP، وقتی برای یکبار آدرس فیزیکی متناظر بـا آدرس IP از یک ایستگاه بدست آمد، پروتکل ARP این دو آدرس را در جدولی درون حافظـهٔ اصلی که ARP Cache نامیده می شود ذخیره می کند تا اگر مجدداً بـه ایـن آدرس نیـاز شد به سرعت در اختیار قرار بگیرد. ساختار هر رکورد از این جدول بصورت زیر است:

IF index	Physical Address	IP Address	Type

IF Index: شماره پورت سخت افزاری متناظر با آن کارت شبکه

Physical Address: آدرس سخت افزاری کارت شبکه

IP Address: آدرس IP متناظر با آدرس سخت افزاری

Type: مقداری که در این فیلد قرار مـی گیـرد وضـعیت هـر رکـورد را در ایـن جـدول مشخص می کند:

مقدار ۱: یعنی این رکورد باید بطور متناوب به هنگام شود. دقت کنید که ARP Cache هر دقیقه یکبار «بهنگام سازی» می شود. مقدار ٤: بدین معناست که این رکورد ثابت و بدون تغییر است و نباید بهنگام شود. مقدار ۱: یعنی رکورد چون بهنگام نشده از اعتبار ساقط است.

نکته آخری که در مورد پروتکل ARP بایستی توضیح بدهیم آن است که در مسیریابها نیز برای شناسائی آدرس ایستگاههای یک شبکهٔ محلی متصل به آنها به همین روش عمل می شود. برای جزئیات دقیقتر پروتکل ARP به REC-826 مراجعه کنید.

برخی از مکانیزمهای استراق سمع در شبکه مبتنی بر ARP شکل می گیرند. یک ماشین که براساس ARP مورد حمله قرار می گیرد در حقیقت بستهٔ دروغین و جعلی مشابه شکل (۷-۲) دریافت می کند که در آن آدرس IP یک ماشین بدورغ مقداری غلط ذکر شده است. این مقادیر اشتباه در حافظهٔ نهان (ARP Cache) درج می شود.

لایهٔ انتقال[1] در شبکهٔ اینترنت

پروتکل IP وظیفهٔ هدایت و مسیریابی بسته های اطلاعاتی را از یک ماشین میزبان به ماشینی دیگر برعهده دارد و مشکلاتی که در طی مسیر ممکن است برای یک بستهٔ IP اتفاق بیفتد، توسط این لایه قابل حل نیست.

[1] Transport Layer

وظیفهٔ لایهٔ انتقال در شبکه، «فراهم آوردن خدمات سازماندهی شده، مطمئن و مبتنی بر اصول سیستم عامل، برای برنامه های کاربردی در لایهٔ بالاتر است، بگونهای که مشکلات و ناکارآمدی لایهٔ IP جبران و ترمیم شود.»

در مقام مقایسه، می توان وظیفه ای را که لایهٔ انتقال برعهده دارد با وظایفی که «سیستم مدیریت فایل» به عنوان بخشی از سیستم عامل بر عهده دارد، قیاس کرد. سیستم مدیریت فایل از یک طرف با ابزارهای ذخیره سازی اطلاعات که ذاتاً سخت افزاری، متنوع و ناهمگون هستند، سر و کار دارد و از طرف دیگر با برنامه های کاربردی در ارتباط است که برای ذخیره و بازیابی اطلاعات فقط مفهومی به نام فایل، در اختیار دارد و از دید برنامه نویس نوع ابزار و چگونگی و محل فیزیکی ذخیرهٔ داده هایش مهم نیست، بلکه فقط عملیات لازم را برنامه ریزی می کند. از دیدگاه ابزارهای ذخیره و بازیابی اطلاعات، چیزی به نام فایل، درایوهای منطقی (مجازی) و جدول FAT[1] معنایی ندارد، بلکه این ابزار می توانند یک بلوک داده را با اندازهٔ ثابت، تحویل گرفته و بر روی محل مشخصی از فضای فیزیکی ذخیره سازی اطلاعات بنویسند(یا بخوانند). سیستم مدیریت فایل که بین این ابزار فیزیکی و برنامه های کاربردی قرار می گیرد از یک ابزار فیزیکی خام، یکپارچه و پیچیده، خدماتی را در قالب مفهوم فایل به برنامه های کاربردی ارائه می کند که کاملاً قابل اعتماد، شفاف، ساده و عاری از هر گونه پیچیدگی سخت افزاری است. سیستم مدیریت فایل برای ارائه چنین خدمتی باید جداول FAT، جدول

[1] File Allocation Table

درایوهای منطقی[1]، سیستم فهرست فایلها[2] و ... را ایجاد و سازماندهی نماید. تنها کاری که برنامه نویس برای بهره گیری از خدمات سیستم فایل باید انجام بدهد آنست که فایلی را بگشاید و تقاضای خواندن از آن یا نوشتن در آن را بدهد. پیچیدگی هایی که در این بین وجود دارد توسط مدیر فایل حل و فصل می شود.

وظیفهٔ لایه انتقال همین مفهوم را دنبال می کند یعنی : « بهره گیری از خدمات لایهٔ IP که سریع و ساده و در عین حال غیر مطمئن و ناکارآمد است و ارائهٔ خدمتی مطمئن، ساختار یافته و شفاف به برنامه ریزی کاربردی در لایهٔ بالاتر ، به گونه ای که برنامه نیست از درگیری با جزییات زیرشبکه و مشکلات کانالهای انتقال و مسایلی از این قبیل به دو باشد.» برای تشریح وظایف لایهٔ انتقال باید کاستی های لایهٔIP را بررسی کرده و سپس روشی را که لایهٔ انتقال برای جبران آنها برگزیده است توضیح بدهیم. دقت کنید که منشأ کاستی های لایهٔIP، ذات کانالهای انتقال و مشکلات فیزیکی در زیرشبکهٔ ارتباطی است.

عمدهٔ این کاستی ها عبارتند از:

۱. تضمینی وجود ندارد وقتی بسته ای برای یک ماشین مقصد ارسال می شود آن ماشین آمادهٔ دریافت آن بسته باشد و بتواند آنرا دریافت کند.

۲. تصمینی وجود ندارد وقتی چند بسته متوالی برای یک ماشن ارسال می شود به همان ترتیبی که بر روی شبکه ارسال شده اند، در مقصد دریافت شوند.

---

[1] Partition Table
[2] Root Directory

۳. تضمینی وجود ندارد که وقتی بسته ای برای یک مقصد ارسال می شود به دلیل دیر رسیدن مجدداً ارسال نشود و در چنین حالتی ممکن است بسته ای به اشتباه دوبار در مقصد دریافت شود. لایهٔ IP قادر نیست تمایزی بین دو بستهٔ عین هم، که یکی از آنها زائد است قائل شود و هر دو را تحویل ماشین مقصد می دهد.

٤. لایهٔIP هیچ وظیفه ای در قبال توزیع بسته ها بین پروسه های مختلی که بر روی یک ماشین واحد اجرا شده اند ندارد. در یک محیط «چند کاربره» یا «چند وظیفه ای»[1] ممکن است چندین پروسهٔ متفاوت تقاضای ارسال یا دریافت داده داشته باشند. حال فرض کنید بسته ای به لایهٔIP از یک ماشین ماشین واحد، تحویل داده شود. داده های درون این بسته متعلق به کدامین پروسهٔ در حال اجرا روی آن ماشین است؟ از دیدگاه لایهٔIP مفهومی به نام «پروسه های متفاوت در حال اجرا»، رسمیت و هویت ندارد.

۵. لایهٔ IP هیچ وظیفه ای در قبال تنظیم سرعت تحویل بسته ها به یک ماشین ندارد. مثلاً ممکن است یک ماشین با سرعت بسیار زیاد بسته هایی را تولید کرده و تحویل لایهٔIP بدهد ولی ماشین مقصد قادر نباشد بسته ها را با این سرعت دریافت کند و بسته ها در مقصد به دلی عدم توانایی در دریافت، از بین بروند.

[1] Multi Task

٦. در لایهٔ انتقال دو پروتکل به نامهای TCP[1] و UDP[2] تعریف شده اند که ابتدا پروتکل TCP را که تمام کاستی های عنوان شده را جبران کرده معرفی می کنیم و نهایتاً به پروتکل UDP و مشخصات آن خواهیم پرداخت.

راهکارهای پروتکل TCP برای جبران کاستی های لایهٔIP

در این بخش مفهوم عملیاتی که پروتکل TCP برای جبران کاستیهای لایهٔ IP انجام می دهد، بررسی می شود و سپس جزئیات این عملیات را در بخشهای آتی ارائه می دهیم.

اولین کاستی در لایهٔIP، عدم تضمین در آماده بودن و توانایی دریافت داده ها توسط ماشین مقصد، عنوان شد. در پروتکل TCP راهکاری ساده و کارآمد برای این مشکل اتخاذ شده است: »برقراری یک ارتباط و اقدام به هماهنگی بین مبدأ و مقصد، قبل از ارسال هر گونه داده.«

برای تشریح این راه حل، فرض کنید پروسه ای روی ماشین A تمایل داشته باشد برای پروسهٔ دیگر بر روی ماشین B، داده هایی را ارسال کند؛ قبل از اقدام به ارسال داده به صورت زیر عمل می کند:

A یک بستهٔ خاص را به عنوان درخواست برای ارتباط. به آدرس B می فرستد و منتظر می ماند.

[1] Transmission Control Protocol
[2] User Datagram Protocol

B در خواست ارتباط را دریافت کرده و بر حسب شرایط، آمادگی یا عدم آمدگی خود را به A اعلام می نماید. (ممکن است B اصلاً وجود خارجی نداشته باشد و طبعاً هیچ پاسخی بر نمی گردد.)

در صورتی که A در یک مهلت زمان مشخص، پاسخ مثبت مبنی بر آماده بودن B دریافت نماید می تواند به ارسال داده ها اقدام نماید.

به پروتکلهایی که قبل از مبادلهٔ داده ها سعی در برقراری یک ارتباط و ایجاد هماهنگی قبلی می نمایند پروتکلهای «اتصال گرا»[1] گفته می شود. در این پروتکلها خاتمهٔ مبادلهٔ داده نیز باید در یک روند هماهنگ و با اطلاع قبلی انجام شود.

معضلات بعدی در لایهٔ IP تضمین به ترتیب رسیدن داده ها و صحت آنهاست. حل این مسائل چندان مشکل نیست. مجدداً فرض کنید پروسهٔ A تمایل داشته باشد برای پروسهٔ B بر روی یک ماشین مشخص، داده هایی را ارسال کند و قبل از اقدام به ارسال داده ها یک ارتباط موفق برقرار کرده باشد. برای تضمین صحت و ترتیب داده ها روند زیر قابل انجام است.

A بخشی از داده هایی که باید ارسال شوند را در قالب یک بسته سازماندهی کرده و در سرآیند آن یک «شماره ترتیب»[2] تنظیم می نماید؛ سپس ضمن نگهداری آن بسته درون یک بافر، آن را جهت هدایت به سمت مقصد، تحنیل لایهIP می دهد و یک «زمان سنج» تنظیم می نماید. همچنین برای نظارت بر خطاهای احتمالی یک کد ۱۶ بیتی کشف خطا

[1] Connection Oriented
[2] Sequence Number

در سرآیند بسته قرار می دهد. در صورتی که B بستهٔ ارسالی از A را سالم دریافت کـرد، یک « زمان سنج» تنظیم می نماید. همچنین برای نظارت بر خطاهای احتمالی یک کد ١٦ بیتی کشف خطا در سرآیند بسته قرار می دهد. در صورتی کـه B بسـتهٔ ارسـالی از A را سالم دریافت کرد،: «پیغام تصدیق» که اختصاراً Ack نامیده می شود بـرای A پـس مـی فرستد.[1]

اگر A در زمان مقرر پیغام Ack را دریافت کرد، بافر مربوط بـه آن بسـته را آزاد کـرده و اقدام به ادامهٔ ارسال داده ها به همین روال می نماید. اگـر بـه دلیـل خرابـی داده هـا (یـا خرابی پیغام Ack در مسیر برگشت) در مهلت مقرر پیغام تصدیق دریافـت نشـود، بسـتهٔ بافر شده از نو ارسال می شود.[2]

با قرار دادن شمارهٔ ترتیب برای داده ها می توان تمین کرد که جریان داده هـا بـه ترتیـب می رسند و اگر به هر دلیلی یک بسته دو بار دریافت شود، با مقایسهٔ شماره های ترتیب، یکی از آنها دور انداخته میشود. با تظیم یک کد ١٦ بیتی کشف خطـا در مبـدأ و بررسـی مجدد آن در مقصد، می توان از صحت داده ها نیز مطمئن شد. جزئیات این عملیـات بـا تشریح پروتکل TCP مشخص خواهد شد.

---

[1] ارسال Ack معمولا بصورت مجزا ارسال نمي شود بلكه ضميمة اطلاعاتي مي شود كه قرار است در پاسخ، ارسال شود، مگر آنكه داده اي براي ارسال وجود نداشته باشد؛ به اين روش Piggy Backing گفته مي شود.

[2] به پروتكلهاي كه فقط در هنگام دريافت صحيح داده ها پيغام ack بر مي گردانند و در صورت دريافت بستة خراب ساكت مي مانند، پروتكلهاي PAR (Positive Acknowledgement with Retransmission) گفته مي شود.

در پروتکل TCP برای به رسمیت شناختن پروسه های مختلفی که بر روی یک ماشین در حال اجرا هستند راه حل زیر ارائه شده است:

هر پروسه برای تقاضای برقراری یک ارتباط با پروسه ای دیگر روی شبکه، یک شمارهٔ شناسایی برای خود بر می گزیند. به این شمارهٔ شناسایی «آدرس پورت»[1] گفته می شود. در سرآیند بسته ای که توسط پروتکل TCP سازماندهی می شود آدرس پورت پروسهٔ فرستندهٔ و آدرس پورت پروسهٔ گیرندهٔ آن درج می شود. یکتا بودن شماره های پورت که به پروسه ها رسمیت و هویت می بخشد، توسط پروتکل TCP به عنوان جزئی از سیستم عامل نظارت خواهد شد. سیستم عامل جدولی را نگهداری می کند که شماره شناسایی تقاضادهندهٔ ارتباط در آن وجود دارد.

آدرس IP، یک ماشین یکتا را در کل شبکه مشخص می نماید؛ شمارهٔ پورت نیز از بین پروسه های اجرا شده بر وری آن ماشین، یکی از آنها را به عنوان مبدأ (یا مقصد) تعیین می کند. بنابراین زوج آدرس IP و آدرس پورت می تواند یک پروسهٔ یکتا و واحد را بر روی هر ماشین در دنیا مشخص نماید. در ادبیات شبکه به این زوج آدرس، «آدرس سوکت» گفته می شود:

(IP Address : Port Number)= socket Address

مثال : 193.142.22.121:80

(البته اصطلاح «آدرس سوکت» نباید با مفهوم «برنامه نویسی سوکت» اشتباه شود)

---

[1] Port Number

برای حل مسئله هماهنگی سرعت ارسال و دریافت در پروتکل TCP الگوریتمی پویا برای تنظیم مجموعهٔ زمان سنجهایی که در این رابطه انجام وظیفه می نمایند بکار گرفته شده است که در بخشی مجزا تشریح خواهد شد.

قبل از وارد شدن به جزئیات پروتکل TCP بهتر است ساختار بسته ای را که این پروتکل برای تحویل به لایهٔ IP تنظیم و سازماندهی می کند، مورد بررسی قرار بدهیم چرا که بسیاری از مسائل با بررسی ساختار این بسته آشکار خواهد شد.(بسته ای که در لایهٔ انتقال تولید و تظیم می شود. «قطعه TCP»[1] یا TPDU[2] نام دارد، که به اختصار به آن بستهٔ TCP خواهیم گفت.)

ساختار بسته های پروتکل TCP

در این بخش یک دید کلی از پروتکل TCP ارائه می نمائیم و ساختار سرآیند بسته ها را در این پروتکل، توضیح خواهیم داد. در زیرساختار یک بستهٔ TCP به تصویر کشیده شده است.

فیلد Source Port: در این فیلد یک شمارهٔ ۱۶ بیتی بعنوان آدرس پورت پروسهٔ مبدأ (که در این بسته را جهت ارسال تولید کرده)، قرار خواهد گرفت.

[1] TCP Segment
[2] Transport Protocol Data Unit

فیلد Destination Port: در این فیلد، آدرس پورت پروسهٔ مقصد که آن را تحویل خواهد گرفت، تعیین خواهد شد.

همانگونه که در بخش قبلی اشاره شد این دو آدرس مشخص می کنند که این بستهٔ از چه برنامهٔ کاربردی در لایه بالاتر تولید و باید به چه برنامه ای در ماشین مقصد تحویل داده شود. برخی از پروسه های کاربردی و استاندارد دارای شماره پورت ۲۵ است. به جدول شماره پورتهای استاندارد در دیسک جانبی کتاب نگاهی بیندازید.

فیلد Sequence Number: این فیلد سی و دو بیتی، شمارهٔ ترتیب آخرین بایتی را که در «فیلد داده» از بسته جاری قرار دارد، نشان می دهد.

در پروتکل TCP شمارهٔ ترتیب، بر حسب شمارهٔ آخرین بایتی است که در بستهٔ جاری قرار گرفته ارسال شده است. بعنوان مثال اگر در این فیلد عددی معادل ۱۹۳٤۱ قرار بگیرد به این معناست که داده ها تا بایت شمارهٔ ۱۹۳٤۱ درون فیلد قرار دارد. دقت کنید که این عدد بمعنای آن نست که به تعداد ۱۹۳٤۱ بایت، درون قسمت داده قرار دارد، بلکه همیشه به شمارهٔ ترتیب آخرین بایت داده ، اشاره می نماید. یعنی ممکن است که کلاً درون فیلد داده فقط یک بایت قرار داشته باشد در حالی که در فیلد شمارهٔ ترتیب عدد ۱۹۳٤۱ قرار داشته باشد . دقت شود که شمارهٔ ترتیب اولین بایت، از صفر شروع نمی شود بلکه از یک عدد تصادف یکه در هنگام برقراری ارتباط به اطلاع طرفین می رسد، شروع خواهد شد.

در فصل هشتم خواهید ید که نفوذگرانی که سعی در ربودن یک نشست (مثل نشست TelNet) دارند به شرطی موفق خواهند شد که بتوانند مقدار اولیهٔ فیلد Sequence Number را حدس بزنند. به مبحث «ربودن نشست یا Session Hijacking مراجعه کنید.

فیلد Acknowledgment: این فیلد ۳۲ بایتی نیز شمارهٔ ترتیب بایتی که فرستندهٔ بسته منتظر دریافت آن است را تعیین می کند. بعنوان مثال اگر در این فیلد عددی معادل ۳٤۲۳۱۰ قرار گرفته باشد بدین معناست که از رشته داده ها (که مشخص نیست چند بایت است) تا شماره ۳٤۲۳۱۰ صحیح و کامل دیافت شده است و منتظر بایتهای از ۳٤۲۳۱۱ به بعد می باشد.

فیلد TCP Header Length: عددی که در این فیلد قرار می گیرد، «طول سرآیند بستهٔTCP را بر مبنای کلمات ۳۲ بیتی تعیین می کند. بعنوان مثال اگر در این فیلد عدد ۷ قرار بگیرد طول سرآیند مقدار ۲۸=۷×٤ بایت خواهد بود. (این فیلد کلاً چهار بیتی است) دقت کنید که قسمت ثابت و اجباری در یک بستهٔ TCP حداقل ۲۰ بایت است ولی در فیلد اختیاری Options می تواند اطلاعاتی قرار بگرد و بنابراین گیرندهٔ یک بستهٔ TCP باید بتواند مرز بین سرآیند بسته و قسمت داده را تشخیص بدهد. پس عددی که در این فیلد قرار می گیرد می تواند بعنوان یک «اشاره گر»، محل شروع داده ها را در یک بستهٔTCP تعیین کند (توجه دارید که مبنای این عدد کلمات ۳۲ بیتی (چهار بایتی) هستند.)

بیتهای Flag(Code Bits): شش بیت بعدی در بستهٔ TPC هر کدام نقش یک بیت پرچم را که معنا و کاربرد مختلفی دارند را بازی می کنند.

در فصول آینده خواهید دید که بر اساس این شش بیت (Code Bits) حملات بسیار متنوعی بر علیه شبکه شک می گیرد لذا باید بدقت با عملکرد این بیتها آشنا باشید.

این بیتها و معنای آنها را به ترتیب بررسی می کنیم:

بیت URG: در صورتی که این بیت مقدار ۱ داشته باشد، معین می کند که در فیلد Urgent Pointer که در ادامه معرفی خواهد شد مقداری قابل استناد و معتبر قرار دارد و بایستی مورد پرداش قرار گیرد. در صورتی که این بیت صفر باشد فیلد Urgent pointer شامل مقدار معتبر و قابل استنادی نیست و از آن چشم پوشی می شود.

بیت Ack: اگر در این بیت مقدار ۱ قرار گرفته باشد، نشان می دهد که عددی که در فیلد Acknowledgement Number قرار گرفته است، دارای مقداری معتبر و قابل استناد است . بیت ACK و بیت SYN نقش دیگری نیز دارند که در ادامه بدان اشاره خواهد شد.

بیت PSH [1]: اگر در این بیت مقدار ۱ قرار گرفته باشد فرستندهٔ اطلاعات از گیرندهٔ تقاضا می کند که داده های موجود در این بسته را بافر نکند و در اسرع وقت آنرا جهت

[1] Push

پردازشهای بعدی تحویل برنامهٔ کاربردی صاحب آن بدهد. این عمل گاهی برای برنامه هائی مشابه Telnet ضروری است.

بیت RST: اگر در این بیت مقدار ۱ قرار بگیرد ارتباط بصورت یک طرفه و ناتمام قطع خواهد شد،[1] بدین معنا که به هر دلیلی (اعم از نقص سخت افزاری یا نرم افزاری) اشکالی بوجود آمده که یکی از طرفین ارتباط مجبور به خاتمه ارتباط فعلی شده است. همچنین بیت RST می تواند بعنوان علامت عدم پذیرش برقراری ارتباط بکار برود. اگر یکی از طرفین ارتباط یک بسته دریافت کند که در آن بیت RST مقدار ۱ داشته باشد، ارتباط بصورت ناهماهنگ و نامتعادل، قطع خواهد شد.

بیت SYN : این بیت نقش اساسی در برقراری یک ارتباط بازی می کند. برقراری یک ارتباط TCP از روند زیر تبعیت می کند:

شورع کننده ارتباط یک بستهٔ TCP بدون هیچگونه داده و با تنظیم بیتهای (ACK=0 , SYN=1) برای طرف مقابل ارسال می کند. در حقیقت ارسال چنین بسته ای به معنای «تقاضای برقراری ارتباط»[2] تلقی می شود.

در پاسخ به درخواست ارتباط، در صورتی که طرف مقابل به برقراری ارتباط تمایل داشته باشد بسته ای بر می گرداند که در آن بیت SYN=1 و بیت ACK=1 است. این بسته نقش «پذیرش یک ارتباط»[3] را بازی می کند.

---

[1] Abnormaly Ended
[2] Connection Reqest
[3] Connecion Accept

برقراری ارتباط را بیشتر توضیح خواهیم داد.

بیت FIN: اگر یکی از طرفین ارتباط، داده دیگری برای ارسال نداشته باشد در هنگام ارسال آخرین بستهٔ خود این بیت را ۱ می کند و در حقیقت ارسال اطلاعات خودش را یک طرفه قطع می کند .در این حالت اگر چه ارسال اطلاعات قطع شده ولیکن طرف مقابل هنوز ممکن است به ارسال اطلاعات مشغول باشد. زمانی ارتباط کاملاً خاتمه می یابد که طرف مقابل نیز در یک بسته با ۱ کردن بیت FIN، ارسال اطلاعات را خاتمه بدهد.

فیلد Windows Size: مقدار قرار گرفته در این فیلد مشخص می کند که فضای بافر گیرنده چند بایت دیگر ظرفیت خالی دارد. ینی به طرف مقابل اعلام می کند که مجاز است از بایت با شمارهٔ ترتیبی که در فیلد Acknowledgement مشخص شده است، حداکثر به اندازهٔ مقداری که در این فیلد درج شده، ارسال داشته باشد و در غیر اینصورت فضای کافی یرای دریافت داده ها وجود نداشته و ناگزیر دو ریخته خواهد شد. اگر مقدار این فیلد صفر باشد بدین معناس که بافر گیرنده تماماً پر شده است و امکان دریافت داده های بعدی وجود ندارد و پروسهٔ فرستنده متوقف خواهد شد؛ در این مورد نیز بیشتر توضیح خواهیم داد.

فیلد Checksum: در این فیلد ۱۶ بیتی، کد کشف خطا قرار می گیرد.

فیلد TCP Segment Length: که در آن طول کل بستهٔ TCP مشخص می شود.

فیلد Urgent Pointer: در این فیلد یک عدد بعنوان اشاره گر قرار می گیرد که موقعیت داده های اضطراری را درون بستهٔ TCP معین می کند. این داده ها، زمانی اتفاق می افتند و ارسال می شوند که عملی شبیه وقوع وقفه ها در هنگام اجرای یک برنامهٔ کاربردی رخ بدهد. بدون آنکه ارتباط قطع شود داده های لازم در همین بستهٔ جاری ارسال خواهد شد. دقت کنید که داده های اضطراری توسط برنامهٔ کاربردی در لایهٔ بالاتر پردازش خواهد شد و برای پروتکل TCP کاربردی ندارد.

فیلد Options: در این فیلد اختیاری است و مقداری نظیر حداکثر طول بستهٔ TCP در آن قرار می گیرد. برای آنکه طول بسته ضریبی از ٤ باقی بماند از این فیلد با کدهایی ارزش استفاده می شود گزینهٔ خاص دیگری در این فیلد تعریف نشده است.

روش برقراری ارتباط در پروتکل TCP

برای برقراری ارتباط در پروتکل TCP از روش «دست تکانی سه مرحله ای» استفاده می شود. البته برقراری ارتباط منوط به این قضیه است که طرفیت ارتباط آمادهٔ برقراری یک ارتباط باشند یعنی یک طرف که فعلاً آن را سرویس دهنده می نامیم برای برقراری ارتباط از طریق توابع سیستمی ()listen و ()accept اعلام آمادگی کرده باشد و طرف مقابل نیز یعنی مشتری با فراخوانی تابع سیستمی ()connect و تعیین آدرس IP و آدرس پورت پروسهٔ مقصد ، تمایل خود را برای ارتباط، ابزار نماید. (عملکرد این توابع را در دیسک جانبی کتاب- «مبحث برنامه نویسی سوکت»- مطالعه کنید.) در چنین حالتی بین

طرفین اتفاقات سه مرحله ای خواهد افتاد. در شکل (۸-۲) این مراحل به تصویر کشیده شده است:

در مرحلهٔ اول، از طرف شروع کنندهٔ ارتباط، یک بستهٔTCP (خالی از داده) ارسال خواهد شد که در آن بیت SYN=1 و بیت ACK=0 است و درون فیلد شماره ترتیب عدد X قرار داده شده که در آن x یک عدد تصادفی است. در حقیقت با این شماره به طرف مقابل اطلاع داده می شود که ترتیب داده های ارسالی از شماره x+1 شروع می شود. در پروتکل TCP شماره ترتیب ۳۲ بیتی است لذا برای پیشگیری از مشکلات احتمالی ناشی از مساوی بودن شمارهٔ ترتیب بسته های ارسالی، دادها از شماره ۰ شروع نمی شوند. بلکه از یک عدد تصادفی (که بصورت خودکار تولید می شود)، شروع می گردد و در همان مرحلهٔ اول، این شمارهٔ ترتیب به طرف مقابل اعلام خواهد شد. بعنوان مثال اگر SEQ=145500 باشد بدین معناس که دادهائی که قرار است ارسال شوند شماره ترتیب آنها از ۱٤٥٥۰۱ آغاز خواهد شد. طرف مقابل حتماً باید از این موضوع باخبر باشند.

در مرحلهٔ دوم، طرف مقابل با دریافت بسته ای با مشخصات فوق الذکر اگر تمایل به برقراری ارتباط نداشته باشد با ارسال یک بستهٔ خالی که در آن بیت RST به ۱ تنظیم شده، این تقاضا را در می کند ولی اگر تمایل به برقراری ارتباط بود یک بستهٔ خالی از داده با مشخصات زیر تولید می کند:

بیت SYN را یک می کند.

بیت ACK را یک می کند .

مقدار فیلد Acknowledgement Number را مقدار تصادفی Y قرار می دهد.

در این مرحله که به معنای پذیرش ارتباط است طرف مقابل با قرار دادن مقدار فیلد Ack=x+1 نشان می دهد که شماره ترتیب x را پذیرفته و منتظر داده ا شماره ترتیب x+1 به بعد است. در ضمن خودش عدد تصادفی y را در فیلد seq.No قرار می دهد و به طرف مقابل اعلام می کند که شماره ترتیب داده های ارسالی ازy خواهد بود.

در مرحلهٔ سوم، شروع کنندهٔ ارتباط با قرار دادن مقادیر زبر شروع ارتباط را تصدیق می کند:

بیت SYN را یک می کند.

بیت ACK را یک می کند.

فیلد seq.No=x+1 را قرار می دهد.

فیلد Ack را y+1 قرار می دهد.

با قرار دادن Seq.No.x+1 و Ack=y+1 شروع کنندهٔ ارتباط اعلام می کند که بر روی پارامترهای شماره ترتیب توافق شده است و او پذیرفته که داده های طرف مقابل را از شماره y+1 بپذیرد. پس از این مرحله ارسال و دریافت داده ها توسط طرفین (تا هنگامی که ارتباط با اطلاع طرفین خاتمه داده نشده است) آزاد است.

شکل (۸-۲) مراحل دست تکانی سه مرحله ای برای برقراری ارتباط در پروتکل TCP

برای خاتمه ارتباط روند زیر صورت می گیرد.

طرفی که داده هایش برای ارسال تمام شده است یک بستهٔTCP ارسال می نماید که در سرآنید این بیت FIN را یک قرار داده است. طرف مقابل این درخواست را دریافت می کند و با ختم یک طرفهٔ آن موافقت می کند. ولی چون ارتباط بصورت یک طرفه ختم می شود طرف مقابل می تواند تا جائی که داده دارد، آنها را ارسال کند و نهایتاً در آخرین بسته، بیت FIN را یک بگذارد تا پس از تصدیق آن، ارتباط به صورت دو طرفه ختم شود.

نکته ای که وجود دارد آنست که اگر یکی از طرفین ارتباط در اثر بروز مشکلی سخت افزاری یا نرم افزاری ارتباط را بدون هماهنگی قطع کند حق ندارد تا ۱۲۰ ثانیه به ارتباط مجدد با همان پروسه اقدام کند و این نتیجه ناشی از آن است که مطمئن باشد بسته های قبلی که که ارسال کرده یا آنکه برایش ارسال شده از زیرشبکه حذف شده اند.

در فصول گذشته ششم و هشتم خواهید دید که تقلب در «دست تکانی سه مرحله ای» شرط موفقیت در برخی از مکانیزمهای پویش یا حمله به سیستم خواهد بود!

کنترل جریان در پروتکل TCP

در اینجا بد نیست که اندکی در مورد نقش فیلد Window size بحث کنیم. همانگونه که قبلاً اشاره شد در پروتکل TCP برای کنترل جریان داده ها از بافر استفاده می شود و داده ها قبل از ارسال به برنامه کاربردی لایه بالاتر بافر شده و بصورت دسته ای تحویل

خواهد شد و گاهاً ممکن است برنامهٔ کاربردی اقدام به دریافت داده های بافر شدهٔ خود در مهلت مقرر نکرده و بافر پر شود. در این حالت گیرنده دیگر قادر به دریافت و ذخیرهٔ داده ها در بافرش نخواهد بود، بهمین دلیل در هر بستهٔ TCP که به طرف مقابل موضوع است خود را با فضای بافر، در این فیلد اعلام خواهد شد . نرم افزار TCP که به طرف دیگر ارسال می شود حجم فضای آزاد بافر، در این فیلد اعلام خواهد شد. نرم افزار TCP در طرف مقابل موظف است خود را با فضای بافر موجود هماهنگ نماید، یعنی بسته ای با طول بزرگتر از فضای بافر اعلام شده ارسال ننماید، در غیر این صورت آن بسته پذیرفته نخواهد شد. بعنوان مثال اگر در یک بستهٔ دریافتی مقدار فیلد Window Size مقدار ٤٠٩٦ باشد بدین معناست که از کل فضای بافر موجود، فعلاً چهار کیلو بایت از آن خالی است.

در این پروتکل به ازای هر ارتباط TCP که موفقیت آمیز برقرار شود، یک «ساختمان دادهٔ» خاص برای آن ایجاد خواهد شد که اطلاعاتی از آخرین وضعیت ارسال یا دریافت جریان داده ها در آن نگهداری می شود. این ساختمان داده، «بلوک نظارت بر انتقال»[1] یا اختصاراً TCB نامیده می شود. برخی از متغیرهای تعریف شده درون ساختمان دادهٔ TCB در جدول زیر معرفی شده است.

[1] Transmission Control Block

نام متغیر	توضیح
متغیرهای نظارت بر ارسال داده ها	
SUN.UNA	شماره ترتیب آخرین بسته ای که ارسال شده ولی هنوز پیغام Ack آن بر نگشته است.
SUN.NXT	شمارهٔ ترتیب آخرین بایت که داده ها از آن شماره به بعد در بسته بعدی که باید ارسال شود.
SUN.WND	میزان فضای آزاد در بافر ارسال
SUN.UP	شماره ترتیب آخرین داده های اضطراری که تحویل برنامهٔ کاربردی شده است.
SUN.WLI	
SUN.WL2	
SUN.PUSH	شماره ترتیب آخرین داده هایی که باید آنتی به برنامهٔ کاربردی گسیل (Push) شود.
SUN.ISS	مقدار اولیهٔ شمارندهٔ ترتیب داده های دریافتی که در حین ارتباط بر روی آن توافق می شود.
متغیرهای نظارت بر دریافت داده ها	
RCV.NXT	شمارهٔترتیب آخرین بایت دربستهٔبعدی که از آن شماره به بعد انتظار دریافت آنرا دارد.
RCV.WND	میزان فضای آزاد در بافر دریافت
RCV.UP	شمارهٔ ترتیب آخرین داده های اضطراری که برای برنامهٔ طرف مقابل ارسال شده است.
RCV.IRS	مقدار اولیهٔ شمارنده ترتیب داده های ارسالی که در حین ارتباط بر روی آن توافق می شود.

در فصل نهم یاد خواهیم گرفت که اساس برخی از حملات dos همین فضایی است که یک ماشین به ازای هر ارتباط TCP در حافظه ایجاد می کند. (به مبحث اشباع منابع سیستمی و همچنین SYN Fliid مراجعه کنید)!

زمان سنجها در پروتکل TCP[1]

عملکرد صحیح پروتکل TCP وابستگی شدیدی به استفادهٔ درست و منطقی از زمان سنجها دارد. در این بخش مهمترین زمان سنجهای بکار رفته در این پروتکل را بررسی می نماییم:

Retransmission Timer: به گونه ای اشاره شد پس از برقراری یک ارتباط ، وقتی بسته ای برای پروسهٔ مقصد ارسال می شود، ضمن نگهداری موقت آن در یک بافر، برای آن یک زمان سنج تنظیم و افعال می شود و اگر در مهلت مقرر پیغام دریافت آن (Ack) نرسید، آن بسته از نو برای مقصد ارسال خواهد شد . این زمان سنج که اختصاراً RT نامیده می شود به یک مقدار پیش فرض، مقدار دهی می شود و شروع به شمارش می معکوس زمان می نماید؛ ؛ هر گاه مقدار آن زمان سنج به صفر برسد ولی پیغام دریافت بسته برنگردد، «رخداد انقضای زمان تکرار» حادث شده و پروسهٔ TCP را وادار به ارسال مجدد آن بسته می کند و مراحل قبلی از نو تکرار می شود.

عملکرد این زمان سنج بسیار ساده است ولی مسئله بغرنج در شبکه آنست که: اولاً پیش فرض این زمان سنج چه مقداری باشد؟ ثانیاً عمل ارسال مجدد یک بسته چند بار تکرار شود؟

[1] TCP Timers

در شبکه های محلی سریع، زمان رفت یک بسته و برگشت پیغام دریافت آن حدود چند هزارم ثانیه طول خواهد کشید در حالی که در شبکه WAN این زمان رفت و برگشت می تواند تا چندین ثانیه طلول بکشد.

اگر قرار باشد زمان پیش فرض زمان سنج RT به مقداری کم تنظیم شود، آنگاه وقتی مقصد روی یک شبکهٔ راه دور واقع است، قبل از آنکه بسته بتواند به مقصد برسد، مهلت این زمان سنج منقضی شده و بسته مجدداً ارسال می شود و این کار برای هر بسته بطور متوالی ترک می شود و ترافیک زائد و بیهوده ای را به شبکهٔ تحمیل می کند. از طرف دیگر اگر قرار باشد زمان پیش فرض این زمان سنج با مقداری بزرگ تنظیم شود در شبکه های محلی و سریع، هنگام بروز یک خطا تاخیر زیادی بوجود خواهد آمد. بهترین راه تنظیم زمان سنج استفاده از روشهای وفقی و پویا است چرا که راندمان پروتکل TCP به شدت به آن وابسته است.

Keep-Alive Timer: ممکن است طرفین یک ارتباط به هر دلیلی ارسال اطلاعات را موقتاً متوقف کنند و هیچ داده های مبادله نشود. هر چند ارتباط TCP فعال و باز باشد. از سوی دیگر ممکن است یکی از طرفین به دلیل مثل خرابی سخت افزاری یا نرم افزاری ، بدون اطلاع ، ارتباط را رها کرده باشد. برای تمایز بین این دو حالت، فرستندهٔ اطلاعات با استفاده از این زمان سنج در بازده های زمانی منظم یک بستهٔTCP که خالی از هر گونه داده های می باشد. برای مقصد ارسال می شود و در صورتیکه پیغام دریافت آن بازگشت، نشان دهندهٔ آنست که ارتباط TCP فعال و باز است؛ در غیر این صورت

ارتباط TCP بصورت یک طرف قطع شده و تمام بافرها و فضای ایجاد شده آزاد می شوند. زمان پیش فرض این زمان سنج مقداررری بین ۵ تا ٤٥ ثانیه می باشد.

Persistence Timer: در پروتکل TCP وقتی یکی از طرفیت ارتباط، مقدار فضای بافر آزاد خود را در فیلد Window Size صفر اعلام کند، ناگزیر پروسهٔ طرف مقابل متوقف (بلوکه) خواهد شد در چنین حالتی پس از آنکه مقداری از فضای بافر پر شده تخلیه شد، این موضوع باید به طرف مقابل گزارش شود تا سیستم عالم، پروسهٔ بلوکه شده را احیا کرده و ادامهٔ ارسال از طرف مقابل ممکن باشد، در غیر اینصورت «بن بست»[1] و تاخیر بی نهایت برای پروسه بوجود خواهد آمد. با استفاده از این زمان سنج پس از آزاد شدن فضای بافر، در فواصل زمانی منظم یک بستهٔTCP برای پروسهٔ بلوکه شده ارسال می شود تا ضمن آگاهی از آخرین وضعیت فضای بافر پروسه بتواند احیا شود.

Quier Timer: ممکن است یک ارتباط TCP بسته شود ولی هنوز بسته هایی سرگردان بر روی شبکه وجود داشته باشند که پس از بسته شدن ارتباط TCP به مقصد برسند، لذا در این پروتکل پس از بسته شدن یک ارتباط با شمارهٔ پورت خاص، بقیهٔ پروسه ها تا مدتی حق استفاده از شمارهٔ پورتی که اخیراً بسته شده را ندارند. این زمان را Quiet Timer مشخص می نماید. مقدار پیش فرض این زمان سنج دقیقاً دو برابر مقدار پیش فرض زمان حیات بستهٔIP بر حسب ثانیه است. (چیزی بین ۳۰ تا ۱۲۰ ثانیه)

---

[1] Deadlock

Idle Timer: این زمان سنج برای آن است که اگر تلاش برای تکرار ارسال یک بسته بیش از حد متعارف انجام شود، ارتباط TCP را بصورت یک طرفه رها کرده و قطع نماید. مقدار معمول این زمان سنج ۳۶۰ ثانیه (٦ دقیقه) است.

پروتکل UDP

پروتکل TCP پروتکلی «اتصالگرا» است و لزوم برقراری یک ارتباط قبل از هر گونه مبادلهٔ داده، می تواند بین چند میلی ثانیه (برای شبکه های محلی سریع) تا چندین ثانیه (برای شبکه های WAN) طول بکشد؛ در ضمن تامل برای بازگشت پیغامهای Ack، یک پروسهٔ کاربردی را با تاخیر مواجه خواهد کرد. برای برخی از کاربردها این زمان قابل تحمل نیست و سرعت در رسیدن بک بسته به مقصد، ضروری تر از پرداختن به مسائلی از قبیل بررسی شمارهٔ ترتیب و ارسال پیغامهای کنترلی محسوب می شود. (کاربردهایی مثل سیستم DNS یا TFTP که در بخشهای آتی بررسی می شوند.)

در لایهٔ انتقال از مدل TCP/IP برای چنین کاربردهایی یک پروتکل ساده و سریع به نام UDP معرفی شده است که به صورت ذاتی «بدون اتصال »[1] است، یعنی بدون هیچ اطلاعی از سرنوشتی که در انتظار یک بسته است، به سمت مقصد ارسال می شود. هر گونه اطلاعی از رسیدن یا نرسیدن داده ها باید در لایه بالاتر بررسی و مدیریت شود.

[1] Connectionless

پروتکل UDP ، تمام کاستی های لایهٔIP را دارد (به غیر از نظارت بر خطای کانال که می تواند وجود داشته باشد) و تنها ارمغان این پروتکل برای پروسه ها سرعت ارسال و کم شدن تأخیرات ناشی ار نظارت بر جریان بسته هاست.

فیلد Source Port: در این فیلد، یک شمارهٔ ۱۶ بیتی بعنوان آدرس پورت پروسهٔ مبدأ که این بسته را جهت ارسال، تولید کرده، قرار خواهد گرفت.

فیلد Destination Port: در این فیلد، آدرس پورت پروسهٔ مقصد که آن را تحویل خواهد گرفت، تعیین خواهد شد.

همانگونه که در بخش قبلی اشاره شده این دو آدرس مشخص می کنند که این بسته از کدام برنامهٔ کاربردی در لایه بالاتر تولید و باید به چه برنامه ای در ماشین مقصد تحویل داده شود.

فیلد UDP: در این فیلد طول بستهٔUDP بر حسب بایت، شامل سرآیند و داده ها ، درج می شود.

فیلد UDP Checksum: در این فیلد ۱۶ بیتی کد کشف خطا درج می شود. روش محاسبهٔاین کد دقیقاً همانند روشی است کهدر پروتکل TCP معرفی شد. تنها تفاوت در آنست که بکارگیری این فیلد اختیاری است و در صورت عدم نیاز به آن، تمام بیتهای آن به صفر تنظیم می وشد. (برای کاربردهایی مثل ارسال دیجیتال صدا یا تصویر)

مناسبترین کاربرد پروتکل UDP برای پروسه هایی است که عملیاتشان مبتنی بر یک تقاضا و یک پاسخ است (سیستم DNS)

با توجه به آنکه UDP پروتکلی بدون اتصال است، جستجوی پورتهای باز UDP، اندکی نفوذ گر را با مشکل مواجه می کند . در این مورد به فصل ششم مراجعه کنید.

مفهوم پورتهای باز

وقتی گفته می شود پورت شمارهٔ N بر روی یک ماشین باز است بدین معناس که بر روی آن ماشین یک پروسهٔ فعال وجود دارد که بسته ای TCP ورودی با شمارهٔ پورت N را پذیرفته و پردازش می کند . در حقیقت آن پروسه از سیستم عالم تقاضا کرده که تمام بسته های TCp (یا UDP) را که شمارهٔ پورت مقصدشان N است، به سمت آن پروسه هدایت کند.

شما می توانید با استفاده از فرمان nestat-na، فهرست تمام پورتهای باز ماشینتان را بدست بیاورید در شکل شکل (۹-۲) خروجی این فرمان دیده می شود.

در فصل ششم به این نکته خواهیم پرداخت که نفوذگران بشدت تلاش می کنند تا فهرست پورتهای باز یک ماشین را کشف نمایند . یک پورت باز به معنای یک پروسهٔ فعال است و یک پروسهٔ فعال می تواند یک رخنهٔ نفوذ به ماشین باشد!

در فصل دهم یاد خواهیم گرفت که نفوذ گران پس از رخنه به سیستم یک پروسهٔ اسب تراوا (که نقش جاسوس را در سیستم بازی می کند) روی ماشین فعال می کنند . حال برای آنکه مسئول آن ماشین نتواند از این موضوع بویی ببرد نفوذ گر مجبور است برنامهٔ اجرایی nestat را به نحوی آلوده کند که فهرست پورتهای باز و پروسه های فعال را بدرستی نشان ندهد.

دیوار آتش

دیوار آتش سیستمی است که در بین کاربران یک شبکه محلی و شبکه بیرونی (مثلاً اینترنت) قرار می گیرد. و ضمن نظارت بر دسترسی ها، در تمام سطوح ورود و خروج اطلاعات را تحت نظر دارد. مدلی ساده برای یک سیستم دیوار آتش در شکل (۱۰-۲) ارائه شده است. در این ساختار هر سازمان با نهادی که بخواهد ورود و خروج اطلاعات شکبه را کنترل کند موظف است تمام ارتباطات مستقیم شبکه داخلی خود را با دنیای خارج قطع کرده و هر گونه ارتباط خارجی را طریق یک دروازه که دیوار آتش یا فیلتر نام دارد، انجام شود.

قبل از آنکه اجزای یک دیواره آتش را تحلیل کنیم باید عملکرد کلی و مشکلات استفاده از یک دیوار آتش را بررسی کنیم.

بسته های TCP و IP قبل از ورود به شبکه (یا خروج از آن) ابتدا وارد دیوار آتش می شوند و منتظر می ماند تا طبق معیارهای حفاظتی و امنیتی پردازش شوند. پس از پردازش و تحلیل بسته سه حالت ممکن است اتفاق بیفتد:

- اجازه عبور بسته صادر شود. (Acdcept Mode)
- بسته حذف گردد. (Blcking Mode)
- بسته حذف شده و پاسخ مناسب به مبداء آن بسته داده شود. (Response Mode)

(به غیر از پیغام حذف بسته می توان عملیاتی نظری ثبت، اخطار، ردگیری، جلوگیری از ادامه استفاده از شبکه و توبیخ هم در نظر گرفت)

در حقیقت دیوار آتش محلی است برای ایست و بازرسی بسته های اطلاعاتی به گونه ای که بسته ها بر اساس تابعی از قواعد امنیتی و حفاظتی، پردازش شده و برای آن مجوز عبور یا عدم عبور صادر شود.

اگر P مجموعه ای از بسته های ورودی به سیستم دیوار آتش در نظر گرفته شود و S مجموعه ای متناهی از قواعد امنیتی باشد داریم:

$$X=F(P,S)$$

F تابع عملکرد دیوار آتش و x نتیجه تحلیل بسته (شامل سه حالت Accept, Blocking, Response) خواهد بود. به مجموعهٔ، قواعد دیوار آتش «سیاستهای امنیتی » نیز گفته می شود؛ به شکل (۱۱-۲) دقت کنید: دیوار آتش یک یا یک بسته ها و تقاضاهای ارتباط TCP را مطابق با سیاستهای امنیتی بازرسی کرده و برای آنها مجوز عبور (یا دستور حذف) صادر می کند.

همانطوریکه همه جا عملیات ایست و بازرسی وقت گیر و اعصاب خرد کن است دیوار آتش هم بعنوان یک گلوگاه[1] میتواند منجر به بالا رفتن ترافیک، تاخیر، ازدحام و نهایتاً بین بست در شبکه شود. (بن بست زمانی است که بسته ها آنقدر در حافظه دیوار آتش معطل می شوند تا طول عمرشان تمام شده و فرستده اقدام به ارسال مجدد آنها کرده و این کار بطور متناوب تکرار شود) به همین دلیل دیوار آتش نیاز به طراحی صحیح و

[1] Bottleneck

دقیق دارد تا از حالت گلوگاهی خارج شـود. (تـاخیر در دیـوار آتـش مجموعـاً اجتنـاب ناپذیر است فقط بایستی بگونه ای باشد که بحران ایجاد نکند.)

مبانی طراحی دیوار آتش

از آنجایی که معماری در شبکه بصورت لایه به لایه است، در مدل TCP/IP برای انتقـال یک واحد اطلاعات از لایه چهارم بر روی شبکه، باید تمام لایه ها را بگذرانـد؛ هـر لایـه برای انجام وظیفهٔ خود تعدادی فیلد مشخص به ابتدای بستهٔ اطلاعاتی اضافه کرده و آنـرا تحویل لایه زیرین می دهد. قسمت اعظم کار یک دیوار آتش تحلیل فیلدهای اضافه شده در هر لایه و سرآیند هر بسته می باشد. در بسته ای که وارد دیوار آتش تحلیل فیلـدهای اضافه شده در هر لایه و سرآیند هر بسته می باشد. در بسته ای که وارد دیوار آتـش مـی شود به تعداد لایه ها (٤ لایه) سرآیند متفاوت وجود خواهد داشت معمولاً سـرآیند لایـه اول (لایه فیزیکی یا Network Interface در شبکهٔ اینترنت) اهمیت چندانی نـدارد چـرا که محتوای این فیلدها فقط روی کانال فیزیکی در شبکه محلی معنا دارنـد و در گـذر از هر شبکه یا مسیریاب این فیلدها عوض خواهند شد. بیشترین اهمیت در سرآیندی است که در لایه های دوم، سوم و چهارم به یک واحد از اطلاعات اضافه خواهد شد:

در لایه شبکه از دیوار آتش فیلدهای سرآیند بسته IP را پردازش و تحلیل می کند.

در لایه انتقال از دیوار آتش فیلدهای سـرآیند بسـته هـای TCP یـا UDP را پـردازش و تحلیل می کند.

در لایه انتقال از دیوار آتش فیلدهای سرآیند و همچنین محتوای خود داده ها را بررسی می کند. (مثلاً سرآیند و محتوای یک نامه الکترونیکی یا یک صفحه وب می تواند مورد بررسی قرار گیرد.)

با توجه به لایه لایه بودن معماری شبکه لاجرم یک دیوار آتش نیز چند لایه خواهد بود. اگر یک بسته در یکی از لایه های دیوار آتش شرایط عبور را احراز نکند همانجا حذف شده و به لایه های بالاتر ارجاع داده نمی شود بلکه ممکن است آن بسته جهت پیگیریهای امنیتی نظیر ثبت عمل و ردگیری به سیستمی جانبی تحویل داده شود سیاست امنیتی یک شبکه مجموعه ای متناهی از قواعد امنیتی است که بنابر ماهیتشان در یکی از سه لایه دیوار آتش تعریف می شوند، بعنوان مثال:

قواعد تعیین بسته های متنوع (بسته های سیاه) در اولین لایه از دیوار آتش

قواعد بستن برخی از پورتها متعلق به سرویسهایی مثل Telnet یا FTP در لایه دوم

قواعد تحلیل سرآیند متن یک نامه الکترونیکی یا صفحه وب در لایه سوم

لایه اول دیوار آتش

لایه اول در دیوار آتش بر اساس تحلیل بسته IP و فیلدهای سرآیند این بسته کار می کند و در این بسته فیلدهای زیر قابل نظارت و بررسی هستند:

آدرس مبداء: برخی از ماشینهای داخل یا خارج شبکه با آدرس IP خاص «حق ارسال» بسته نداشته باشند و بسته های آنها به محض ورود به دیوار آتش حذف شود. (آدرسهای IP غیرمجاز توسط مسئول دیوار آتش تعریف می شود)

شماره شناسایی یک دیتاگرام قطعه قطعه شده[1]: بسته هائی که قطعه قطعه شده اند یا متعلق به یک دیتاگرام خاص هستند حذف شوند.

شماره پروتکل: بسته هایی که متعلق به پروتکل خاصی در لایه بالاتر هستند می توانند حذف شود. یعنی بررسی اینها بسته متعلق به چه پروتکلی در بالاتراست و آیا برای تحویل به آن پروتکل مجاز است یا خیر.

زمان حیات بسته بسته هائی که بیش از تعداد مشخصی مسیریاب را طی کرده اند مشکوک هستند و باید حذف شوند.

زمان حیات بسته: بسته هائی که بیش از تعداد مشخصی مسیریاب را طی کرده اند مشکوک هستند و باید حذف شوند.

بقیه فیلدها بنا بر صلاح دید و قواعد امنیتی مسئول دیوار آتش قابل بررسی هستند.

مهمترین خصوصیت لایهٔ اول از دیوار آتش آنست که در این لایه بسته ها بطور مجزا و مستقل از هم بررسی می شوند و هیچ نیازی به نگه داشتن بسته های قبلی یا یا بعدی یک بسته نیست. بهمین دلیل ساده ترین و سریعترین تصمیم گیری در این لایه انجام می شود. امروزه برخی از مسیریابها با امکان لایه اول دیوار آتش به بازار عرضه می شوند یعنی به غیراز مسیریابی، وظیفهٔ لایه اول یک دیوار آتش را هم انجام می دهند که به آنها «مسیریابهای فیلتر کنندهٔ بسته»[2] گفته می شود. بنابراین مسیریاب قبل از اقدام به

[1] Identifir & Fragment offset
[2] Pocket Filtering Router

مسیریابی، بر اساس جدولی بسته های IP را غربال می کند و تنظیم این جدول بر اساس نظر مسئول شبکه و برخی از قواعد امنیتی انجام می گیرد.

با توجه به سریع بودن این لایه هر چه درصد قواعد امنیتی در این لایه دقیقتر و سختگیرانه تر باشد حجم پردازش در لایه های بالاتر کمتر و در عین حال احتمال نفوذ پایینتر خواهد بود ولی در مجموع بخاطر تنوع میلیاردی آدرسهای IP نفوذ از این لایه با آدرسهای جعلی یا قرضی امکان پذیر خواهد بود. و این ضعف در لایه های بالاتر بایستی جبران شود.

لایه دوم دیوار آتش

در این لایه از فیلدهای سرآیند لایه انتقال برای تحلیل بسته استفاده می شود. عمومی ترین فیلدهای بسته های لایهٔ انتقال جهت بازرسی در دیوار آتش، عبارتند از:

شماره پورت پروسه مبدإ و شماره پورت پروسه مقصد با توجه به آنکه پورتهای استاندارد شناخته شده هستند ممکن است مسئول یک دیوار آتش بخواهد سرویس ftp (انتقال فایل فقط در محیط شبکه محلی امکان پیر باشد و برای تمام ماشینهای خارجی این سرویس وجود نداشته باشد بنابراین دیوار آتش می تواند بسته های TCP با شماره پورت ۲۰ و ۲۱ (مربوط به ftp) که قصد ورود یا خروجی از شبکه را دارند، حذف کند. یکی دیگر از سرویسهای خطرناک که ممکن است مورد سوءاستفاده قرار گیرد Telnet است که می توان براحتی ۱ورت ۲۳ را مسدود کرد یعنی بسته هائی که شماره پورت مقصدشان ۲۳ است حذف شوند.

فیلد شماره ترتیب[1] و فیلد Acknowledgment : این دو فیلد نیز بنا بر قواعد تعریف شده توسط مسئول شبکه قابل استفاده هستند.

کدهای کنترلی (TCP Code BITS): در بخشهای قبلی با نقش کلیدی این بیتها آشنا شدید. دیوار آتش با بررسی این کدها، به ماهیت آن بسته پی برده و سیاستهای لازم را بر روی آن اعمال می کند. بعنوان مثال یک دیوار آتش ممکن است بگونه ای تنظیم شود که تمام بسته هایی که از بیرون به شبکه وارد می شوند و دارای بیت SYN=1 هستند را حذف کند. بدین ترتیب هیچ ارتباط TCP از بیرون به درون شبکه برقرار نخواهد شد.

از مهمترین خصوصیات این لایه آنست که تمام تقاضاهای برراری ارتباط TCP بایستی از این لایه بگذرد و چون در ارتباط TCP ، تا مراحل «دست تکانی سه گانه اش» به اتمام نرسد انتقال داده امکان پذیر نیست لذا قبل از هر گونه مبادله داده دیوار آتش میتواند مانع برقراری ارتباط شود. دیوار آتش در این لایه نیاز به جدولی از شمارهٔ پورتهای غیر مجاز دارد.

[1] Sequence Number

مجوع قواعد امنیتی تعریف شده در لایهٔ اول و دوم در یک جدول همانند مثال زیر تنظیم و به دیوار آتش اعمال می شوند.

Action	Source Address	Destination Address	Protoco	Source port	Destination Port	Code Bit
Allow	Inside Network Address	outside Network Address	TCP	Any	80	Any
Allow	Inside Network Address	outside Network Address	TCP	80	>1023	ACK
Deny	All	All	All	All	All	All

لایه سوم دیوار آتش

در این لایه حفاظت بر اساس نوع سرویس و برنامه کاربردی انجام می شود. یعنی بـا در نظر گرفتن پروتکل در لایه چهارم به تحلیل داده های پردازند. تعداد سرآیند در این لایـه بسته به نوع سرویس بسیار متنوع و فراوان است. بنابراین در لایه سوم دیوار آتـش بـرای هر سرویس مجرا (مثل سرویس پست الکترونیکی، سرویس ftp سرویس وب و...) باید یک سلسله پردازش و قواعد امنیتی مجزا تعریف شود و به همین دلیل حجم و پیچیدگی پردازش در لایه سوم زیاد است. توصیه موکد آنست که تمام سرویسهای غیرضـروری و شماره پورتهایی که مورد استفاده نیستند در لایه دوم مسدود شوند تا کـار در لایـه سـوم کمتر باشد.

بعنوان مثال فرض کنید موسسه ای اقتصادی، سرویس پسـت الکترونیکـی خـود را دائـر کرده ولی نگران فاش شدن برخی اطلاعات محرمانه است. در این حالت دیوار آتـش در لایه سوم می تواند کمک کند تابرخی از آدرسهای پست الکترونیکـی مسـدود شـود، در

عین حال می تواند در متون نامه های رمز نشده دنبال برخی از کلمات کلیدی حساس بگردد و متون رمزگذای شده را در صورتی که موفق به رمز گشائی آن نشود حذف نماید.

بعنوان مثال دیرگ یک مرکز فرهنگی علاقمند است قبل از تحویل صفحه وب به یک کاربر، درون آنرا از لحاظ وجود برخی از کلمات کلیدی بررسی کند و اگر کلماتی که با معیارهای فرهنگی مطابقت ندارد درون متن صفحه یافت شده آن صفحه را حذف نماید.

فیلترهای Stateful و هوشمند

دقت کنید که فیلترهای معمولی کارآیی لازم را برای مقابله با حملات ندارند زیرا آنها بر اساس یکسری قواعد ساده بخشی از ترافیک بسته های ورودی به شبکه را حذف می نمایند.

امروزه بر علیه شبکه ها حملاتی بسیار تکنیکی و هوشمند طرح ریزی می شود بگونه های که یک فیلتر ساده (که قواعد آن بر همگان آشکر است) قابل اعتماد و موثر نخواهد بود. در فصل ششم خواهید دید که نرم افزار Firewalk بسادگی قواعد دیوار آتش را کشف کرده و در اختیار نفوذگر قرار می دهد؛ سپس او بر اساس این قواعد برای رسوخ به شبکه تلاش خواهد کرد.

بدیهی است که یک فیلتر یا دیوار آتش قطعاً بخشی از ترافیک بسته ها را بدرون شبکه هدایت خواهد کرد.(زیرا در غیر این صورت شبکهٔ داخلی، هیچ ارتباطی با دنیای خارج نخواهد داشت.) نفلوذ گر برای آنکه ترافیکدادههای مخرب او حذف نشود تلاش می

کند تا با تنظیم مقادیر خاص در فیلدهای بستهٔTCP (و IP) آنها را با ظاهری کاملاً مجاز از میان دیوار آتش یا فیلتر بدرون شبکه بفرستد. (در این مورد در فصول ششم، هفتم، دهم و یازدهم بحث خواهد شد.)

بعنوان مثال فرض کنید فیلتری تمام بسته ها بغیر از شمارهٔ پورت ۸۰ (مربوط به ترافیک وب) را حذف می کند. حال یک نفوذگر در فاصلهٔ هزاران کیلومتری می خواهد فعال بودن یک ماشین را در شبکه بیازماید. بدلیل وجود فیلتر او قادر نیست با ابزارهایی مثل Ping، Nmap، Cheops و نظایر آنها، از ماشینهای درون شبکه اطلاعاتی کسب کند؛ بنابراین برای غلبه بر این محدودیت، بصورت مصنوعی یک بستهٔSYN_ACK (با شمارهٔ پورت مبداء ۸۰) به سمت ماشین هدف می فرستد. یک دیوار آتش معمولی، با بررسی فیلد Source Port به این بسته اجازه ورود به شبکه را می دهد؛ زیرا ظاهر این بسته نشان می دهد که توسط یک سرویس دهندهٔ وب تولید شده است و حامل داده های وب می باشد. بسته بدورن شبکهٔ داخلی راه یافته و چون ماشین داخلی انتظار دریافت آنرا نداشته پس از دریافت، یکی از پاسخهای RESET یا ICMO Port Unreliable را بر می گرداند. هدف نفوذگر بررسی فعال بودن چنین ماشنی بوده است و بدین ترتیب به هدف خود می رسد؛ فیلتر بسته (یا دیوار آتش) نتوانسته از این موضوع باخبر شود:

برای مقابله با چنین عملیاتی دیوار آتش باید فقط به آن گروه از بسته های SYN-ACK اجازه ورود به شبکه بدهد که در پاسخ به یک تقاضای SYN قبلی ارسال شده اند. همچنین باید بشرطی بسته های ICMP Echo Reply بدون شبکه هدایت شود که حتماً

در پاسخ به یک پیام ICMP Echo Request باشد. بعنی دیوار آتش (یا فیلتر) باید بتواند پیشینهٔ بسته های قبلی را حفظ کند تا در مواجهه با چنین بسته هـایی، بدرسـتی تصـمیم بگیرد.

دیوارهای آتش با فیلترهایی که قادرند مشخصـات ترافیـک خروجـی از شـبکه را بـرای مدتی حفظ کنند و بر اساس پردازش آنها مجوز عبور صادر نمایند. » دیوار آتش یا فیلتـر هوشمند و Stateful» نامیده می شوند.

البته نگهداری مشخصات ترافیک خروجی شـبکه (یـا ورودی) در یـک فیلتـر Stateful همیشگی نیست بلکه فقط کافی است که ترافیک چنـد ثانیـهٔ آخـر را بـه حافظـهٔ خـود بسپارد!

وجود فیلترهای Stateful باعی می شود بسته هایی که با ظاهر مجاز می خواهند بـدورن شبکه راه پیدا کنند از بسته های واقعی تمیز داده شوند. در زیـر مثـالی از جـدول قواعـد یک فیلتر Stateful را ملاحظه می کنید:

Source Address	Destination Address	Source Port	Dextination Port	Timeout (seconds)
10.1.1.20	10.34.12.11	2341	80	60
10.1.1.34	10.22.21.45	32141	80	40

بزرگترین مشکل این فیلترها غلبه بر تأخیر پردازش و حجم حافظهٔ مورد نیاز مـی باشـد ولی در مجموع قابلیت اعتماد بسیار بالاتری دارنـد و ضـریب امنیـت شـبکه را افـزایش خواهند داد. اکثر فیلترهای مدرن از این مکانیزم بهره گرفته اند.

یک« دیواره آتش یا فیلتر هوشمند و Stateful» پیشینهٔ ترافیک خروجی را برای چند ثانیه به خاطر می سپارد و بر اساس آن تصمیم می گیرد که آیا ورود یک بستهٔ مجاز است یا خیر.

دیوار آتش مبتنی بر پراکسی (Proxy Based Firewall)

فیلترها و دیوارهای آتش معمولی و Stateful فقط نقش ایست و بازرسی بسته ها را ایفا می کنند. هرگاه مجوز برقراری یک نشست صادر شد این نشست بین دو ماشین داخلی و خارجی بصورت مستقیم (انتها به انتها) برقرار خواهد شد؛ بدین معنا که بسته های ارسالی از طرفین پس از بررسی، عیناً تحویل آنها خواهد شد.

فیلترهای مبتنی بر پراکسی و رفتاری کاملاً متفاوت دارند:

وقتی ماشین مبدأ، تقاضای یک نشست (مثل نشست FTP یا برقراری ارتباط TCP با سرویس دهندهٔ وب) را برای ماشین مبدأ، این نشست را برقرار می کند. یعنی طرف نشست دیوار آتش خواهد بود نه ماشین اصلی! سپس یک نشست مستقل بین دیوار آتش و ماشین مقصد برقرار می شود. پراکسی داده های مبداء را میگیرد، سپس از طریق نشست دوم برای مقصد ارسال می نماید. بنابراین:

در «دیوار آتش مبتنی بر پراکسی « هیچ نشست مستقیم و رو در روئی بین مبداء و مقصد شکل نمی گیرد بلکه ارتباط آنها بوسیلهٔ یک ماشین واسط برقرار می شود. بدین نحو دیوار آتش قادر خواهد بود بر روی داده های مبادله شده در خلال نشست. اعمال نفوذ کند.

نشست بین مبداء و پراکسی

نشست بین پراکسی و مقصد

حال اگر یک نفوذ گر بخواهد با ارسال بسته های کنترلی خاص (مثلا SYN-ACK) که ظاهراً مجاز به نظر می آیند واکنش ماشین هدف را در شبکهٔ داخلی ارزیابی کند در حقیقت واکنش دیوار آتش را مشاهده می کند و لذا نخواهد توانست از درون شبکهٔ داخلی اطلاعات مهم و باارزشی بدست بیاورد.

با توضیحات فوق شاید حدس زده باشید که «دیوار آتش مبتنی بر پراکسی » در لایهٔ سوم عمل می کند و قادر است حتی بر داده های ارسالی در لایهٔ کاربرد (مثل محتوی نامه های الکترونیکی با صفحات وب)

دیوارهای آتش مبتنی بر پراکسی به حافظهٔ نسبتاً زیاد و CPU بسیار سریع نیازمدند و لذا نسبتاً گران تمام می شوند.

چون دیوار آتشی مبتنی بر پراکسی، باید تمام نشستهای بین ماشینهای درون و بیرون شبکه را مدرت و اجرا کند لذا گلوگاه شبکه محسوب می شود و هر گونه تأخیر با اشکال در پیکربندی آن، کل شبکه را با بحران جدی مواجه خواهد کرد.

بهترین پیشنهاد: استفادهٔ همزمان از هر دو نوع دیوار آتش

ممکن است از شما سوال شود که استفاده از کدام نوع دیوارهای آتش (Stateful یا Proxy Based) در شبکه ای که امنیت داده ها در آن حیاتی است، منطقی تر و امن تر خواهد بود؟

اگر قرار باشد از دیوار آتش مبتنی بر پراکسی در شبکه استفاده شود، اندکی از کارآیی سرویس دهنده هائی که ترافیک بالا (مثل سرویس دهندهٔ وب) دارند کاسته خواهد شد، زیرا پراکسی یک گلوگاه در شبکه محسوب می شود. اگر سرویس دهنده ای را برای کل کاربران اینترنت پیکربندی کرده اید بهتراست در پشت یک دیوار آتش مبتنی بر پراکسی قرار نگیرد.

در طرف مقابل فیلترها و دیوارهای آتش معمول سریعند ولیکن قابلیت اعتماد کمتری دارند و نمی توان بعنون حصار یک شبکه به آنها اطمینان اکرد در نتیجه بهترنی پیشنهاد، استفادهٔ همزمان ا رهر دو نوع دیوار آتش است! بهشکل (۱۳-۲) دقت کنید: شبکه های متعلق به سازمنها یا موسسات تجاری، در دو بخش سازماندهی و پیکربندی می شوند:

بخش عمومی شبکه شامل سرویس دهنده های وب، پست الکترونیکی و FTP که به عموم کاربران اینترنت سرویس می دهد. این بخش اصطلاحاً DMZ (بخش غیرمحرمانه، غیر نظامی!) نام دارد.

بخش خصوصی یا محرمانهٔ که صرفاً با هدف سرویس دهی به اعضای آن سازمان یا موسسه پیاده سازی شده است.

مطابق با شکل (۱۳-۲) بخش عمومی شبکه توسط یک فیلتر (معمولی یا هوشمند) حفاظت می شود تا از کارآیی سرویس دهده ها کاسته نشود. شبکهٔ داخلی در پشت یک دیوار آتش مبتنی بر پراکسی پنهان می شود تا ضمن غیر قابل نفوذ بودن با اینترنت در ارتباط باشد. در چنین ساختاری یک نفوذگر خارجی برای برقراری ارتباط با یک ماشین

داخلی دو مانع عمده بر سر راه دارد: فیلتر و دیوار آتش مبتنی بر پراکسی؛ حال اگر بتواند حتی با مکانیزمهای متداول از سد فیلتر بگذرد پشت دیوار آتش متوقف خواهد شد.

دیوارهای آتش شخصی!

یک دیوار آتش کل ماشینهای شبکه داخلی را حفاظت می کند. سوال مهم آنست که در محیطهای عمولی همانند ISP که هیچ دیوار آتش یا فیلتری نصب نشده و ماشینهای اعضای شبکهای حفاظ رها شده اند تکلیف کاربران بی گناه چیست؟!

بسیاری از کاربردان ISP که از مودم های معمولی یا سریع (مثل سری xDSL)برای اتصال به شبکهٔ اینترنت استفاده می کنند بدلیل عدم وجود یک سیستم امنیتی قدرتمند به دام نفوذ گران بدخواه می افتند، داده هایشان سرقت می شود و یا مورد آزاد و اذیت قرار می گیرند. اینگونه حوادث نادر نیست بلکه هر روز اتفاق می افتد؛ حال چگونه می توان از این ماشینها حفاظت کرده؟ دیوار آتش شخصی!

دیوار آتش شخصی (personal Firewall) یک ابزار نرم افزاری است که بر روی ماشین نهایی (Host) نصب می شود و ورود /خروج بسته ها به از آن ماشین را نظارت می کند؛ مانع دسترسی غیرمجار به منابع آن ماشین شده و از داده های یک کاربر بی اطلاع حفاظت می کند!

در ویندوز XP هنگام نصب، یک دیوار آتش رایگان (با یکسری قواعد پیش فرض و نسبتاً مطمئن) بر وری ماشین کاربر فعال شده ترافیک بسته ها را نظارت می کند و حتی

الامکان از دسترسی غیرمجاز به آن جلوگیری می نماید. اگر نسخه های قدیمی تر (مثل 9x یا ME) را نصب کرده اید باید از نرم افزارهای مستقل استفاده کنید. مشهورترین دیوارهای آتش نرم افزاری عبارتند از:

- Norton Firewall 2002
- ZoneAlarmpro (ZAP 2001)
- ProtectX Professional

این سه نرم افزار بهمراه دو کتاب ارزشمند در زمینهٔ دیوار آتش در CD جانبی کتاب موجود می باشد.

برای آشنایی با دیوار آتش ZoneAlamPro به ضمیمهٔ (ب) مراجعه کنید.

راهکارهای تامین امنیت در سطح شبکه

در TCP/IP عملایت حفاظتی و امنیتی گنجانده نشده و تامین امنیت داده ها بر عهده برنامه های کاربردی گذاشته شده است. در سالیان اخیر تلاشهای بسیار زیادی در بالا بردن امنیت شبکه های مبتنی بر TCP/IP صورت گرفته است؛ مهمترین نتیجهٔ این تلاشها را که در قالب استانداردهای جهانی عرضه شده اند، می توان دو استاندارد زیر برشمرد:

SSL[1]: رمز نگاری داده ها قبل از تحویل به لایهٔ انتقال:

یکی از راهکارهای تامین امنیت داده ها در سطح برنامه های کاربردی استفاده از SSL است. برنامهٔ کاربردی (مثل مرورگر یا سرویس دهندهٔ وب) داده های خود را جهت ارسال تحویل لایه انتقال نمی دهند بلکه آنها رابه لایهٔ دیگری به نام SSL می فرستند تا قبل از تحویل به لایهٔ انتقال رمز نگاری شوند. شکل (۱۴-۲) شما یکی SSL و موقعیت آنرا در مدل چهار لایه ای TCP/IP نشان می دهد.

هر کاربر یا گروه می تواند یک گواهینامهٔ دیجیتالی داشته باشد تا قبل از آنکه داده ها بین طرین مبادله شوند، مبتنی بر SSL یکدیگر را احراز هویت کنند. (البته احراز و یا در SSL انتخابی است و برنامهٔ کاربردی ممکن است فقط به رمز نگاری داده ها بسنده کند.) اگر هنگامی که شما به یک سایت وب مراجعه کرده اید در گوشهٔ پایین مروگر شما تصویر یک قفل یا کلید ظهر شود به معنای آنست که داده ها ارسالی از سرویس دهنده وب، رمز نگاری شده ارسال می شود.

در فصل هشتم مکانیزم «احراز هویت در SSL» را توضیح داده ایم: لذا برای پرهیز ارتکراری شدن مطالب شما را به بخش «استراق سمع از سوکتهای SSL » ارجاع میدهیم.

در فصل هفتم خواهید دید که یک نفوذگرا از طریق پراکسی Achilles تلاش می کند حتی داده های رمزنگاری شده توسط SSL مشاهده کند!

---

[1] Secure Socket Layer

IPSec ؛ رمز نگاری و احراز IETF [1] در اواسط ۱۹۹۰ نتایج تحقیقات خود را در مورد تامین امنیت داده ها در سطح لایهٔ IP، با عنوان IPSec که در REC 2401 تا RFC 2412 توصیف شده است با Ipv.4 سازگار است و لازم نیست در آن تغییر داده شود. یعنی بسته های رمزنگاری شدهٔ IPSec نهایتاً در بستهٔIP جاسازی می شوند. (کپسوله می شوند.) با این مشخصه، دیگر مسیریابهای میانی نیازی به تغییر نداشته و فرآیند مسیریابی تغییر نمی کند.

این RFC ها که مستندات مرجع محسوب می شوند در CD جانبی کتاب ضمیمه شده اند.

شرح مشخصات IPSec خود به یک کتاب مفصل نیاز دارد ولیکن برای ارائه یک پیش زمینهٔ کلی به ویژگیهای عمده آن می پردازیم:

IPSec با استفاده از مکانیزمهای احراز هویت و امضای دیجیتالی، قبل از پذیرش داده ها ابتدا از مبداء آن بسته مطمئن خواهد شد؛ بدین ترتیب هیچ نفوذگری نخواهد توانست بسته های IP جعلی بسازد و آنرا از طرف یک ماشین دیگر ارسال کند.

---

[1] در سال ۱۹۸۳ کمیتة (Interne Control and Configuration Board)ICCB بعنوان گروه طراحي اينترنت يا IAB (Internet Architecture Board) به جهان معرفي شد. اين كميته يك سازمان مستقل براي طراحي استانداردها و ترويج تحقيقات در زمينة تكنولوژي اينترانت است. كميتة IAB اكنون نيز وجود دارد و در دو قسمت فعاليت مي كند:

- گروه IETF يا Internet Engineering Task Force : موارد فني و مشكلات استانداردها و تكنولوژي بكار گرفته شده در شبكة اينترنت را بررسي و حل مي كند و جزئيات پروتكلهاي فعلي را در اختيار عموم قرار مي دهد.
- گروه IRTF يا Intenet Research Task Force: كار تحقيقات به منظور بهبود و ارتقاء اينترنت را بر عهده دارد.

IPSec امنیت اطلاعات را با استفاده از الگوریتمهای مختلف رمزنگاری تامین می کند. بدین ترتیب نفوذگران بهیچوجه قادر به استراق سمع داده ها نیستند، زیرا تا وقتی کلید رمز را بدست نیاورند هیچ نفعی از اطلاعات استراق سمع شده نخواهند برد.

IPSec صحت و سلامت اطلاعات (Integrity) را تضمین خواهد کرد و هیچ نفوذگری نخواهد توانست داده ها را در طول مسیر، دستکاری کند.

IPSec در سطح لایهٔ IP (لایهٔ دوم) پیاده سازی شده و برای بکارگیری آن بر روی یک ماشین، پروتکلهای لایه های بالاتر نیاز به تغییر ندارند.

IPSec از دو پروتکل مجزا به نامهای AH و (Encapsulating Security Payload)ESP تشکیل شده است که هر کدام وظیفهٔ خاصی دارند:

(Authentication Header)AH

بگونه ای که از نام این پروتکل مشخص است، AH بسته ها را احراز هویت می کند . یعنی قبل از بهره برداری از بسته مطمئن می شود که یه بسته از مبداء واقعی آن تولید شده است یا آنکه جعلی است.

AH برای تایید هویت تولید کنندهٔ بسته از اصول امضاهای دیجیتالی بهره می گیرد که مقدمهٔ کوتاهی از آن در ضمیمهٔ (ج) ارائه شده است. در ضمن وظیفهٔ دارد تا صحبت داده ها و عدم تغییر یک بسته در مسیر را تایید کند. ساختار بسته AH (از استاندارد IPSec) بصورت زیر است:

در حقیقت در یک بستهٔ IPSec بجای آن که پس از سرآیند بستهٔ IP، سرآیند بستهٔ متعلق به لایهٔ انتقال قرار گرفته باشد، سرآیند IPSec شروع می شود. برای تحلیل فیلدها و جزئیات آنها باید به منابع موثق IETF که ضمیمهٔ کتاب است، مراجعه کنید.

فرض کنید که احراز هویت مبداء تولید کنندهٔ بسته و تایید صحت آن توسط AH بدقت انجام شود؛ آیا این عملیات امنیت داده ها را تضمنی می کند؟ هنوز نه! زیرا گر چه نفوذ گر قادر نیست داده های ارسالی بر روی مسیر را دستکاری کند یا بسته ای را بصورت جعلی برای ماشینی بفرستد ولی مسئلهٔ استراق سمع داده ها حل نشده است . اینجاست که نقش پروتکل ESP آشکار می شود:

(Encapsulating Security Payload)ESP

ESP مسئلهٔ محرمانه ماندن اطلاعات و خطر استراق سمع را رفع خواهد کرد. در زیرساختار بستهٔESP را مشاهده می کنید:

در این بسته بین دو فیلد ESP Header و ESP Trailer بستهٔ متعلق به لایهٔ انتقال (مثلاً TCP) قرار می گیرد که کاملاً رمزنگاری شده است و تا کسی کلید رمز آنرا نداشته باشد نخواهد توانست آنها را استخراج و بهره برداری کند. ESP می تواند از طریق امضاهای دیجیتالی بسته ها را احراز هویت کند. (به ضمائم کتاب مراجعه کنید.)

IPSec با تمام توانائیهایش در تامین امنیت از اطلاعات هنوز نتوانستهٔ گسترش جهانی پیدا کند اگر چه شرکتهای بزرگی چون مایکروسافت بر روی آن سرمایه گذاری کرده اند و در محصولات جدید خود (مثل Windows 2000/XP) آنرا عرضه نموده اند وی

بکارگیری عملیات آن بسیار ناچیز بود است. شاید بتوان دلیل عدم رونق IPSec را به دو عامل زیر ربط داد:

IPSec بسیار پیچیده و طولانی است و این پیچیدگیها باعث شده تا نسخه های پیاده سازی شدهٔ آن از لحاظ سازگاری در شبکه شکلاتی داشته باشند، لذا نباید انتظار داشت موسسات سازمانها تمام حیثت خود را در گروه نصب IPSec بگذارند و مشکلاتی که کاربران آنها خواهند داشت گریبان آنها را بگیرد!

مشکل دیگری که مانع رشد سریع IPSec بگذارند و مشکلاتی که کاربران آنها خواهند داشت گریبان آنها را بگیرد!

مشکل دیگری که مانع رشد سریع IPSec شده «مراکز گواهی امضاهای دیجیتالی و توزیع کلیدهای رمزنگاری است. تا موقعی که یک عزم جهانی برای تاسیس «سرویس دهنده های مطمئن گواهی امضاء و توزیع کلید Keys Distribution » وجود نداشته باشد، استفاده IPSec نیز گسترش انتظار می رود در آینده ای نه چندان دور ، باغوغایی که نفوذ گران براه انداخته اند، IPSec جای خود را در شبکه بازکند، ولیکن اکنون IPSec نسبتاً مهجور و بلااستفاده است و نفوذگران هم به کار خود مشغولند! البته با توجه به هوش بشی IPSec یا هر مکانیزم امنیتی، مطمئناً بساط نفوذگران را جمع نخواهد کرد بلکه جمع آنها را کوچکتر و جدالهای آنها را پیشرفته و دیدنی خواهد کرد!

هنر استفاده از موتورهای جستجو در اینترنت

به غیر از بررسی سایت وب شبکهٔ هدف که اطالعات محودی را در اختیار نفوذگر قرار می دهد. اطلاعات دیگری در مورد یک شبکه بصورت پراکنده در وب وجود دارد. برای گردآوری این اطلاعات پراکنده، بهترین ابزار موتورهای جستجوی رایگان مثل Google ، Altavista و Excite هستند. گروههای خبری و حتی مجلات و روزنامه های الکترونیکی و سایتهای متعلق به شرکتهای رقیب نیز گاهی اوقات اطلاعات مفیدی را در اختیار نفوذگر قرار می دهند.

از طریق موتورهای جسجو می توان تمام سایتهائی که به شبکهٔ هدف حمله لینک داده اند (یعنی صفحاتی وبی که بهر طریق به آدرس شبکهٔ هدف ارجاع داده اند) را پیدا کرد. بگونه ای که در شکل (۱-۵) مشاهده می شود، در موتور جستجوی Alavista می توان با تایپ گزینهٔ :Link و درج آدرس حوزهٔ مورد نظر (در جلوی آن)، تمام سایتهای وبی را که به این آدرس ارجاع داه اند، بدست آورد. ممکن است ارجاع دهندگان به این آدرس شرکای تجاری یا رقبای شبکهٔ هدف باشند. البته برای سایتهای معروف و شناخته شده، پیدا کرده سایتهای وبی که به آدرس آنها ارجاع داده اند چندان مفید نخواهد بود زیرا ممکن است هزاران هزار سایت به چنین آدرسی ارجاع داده باشند.

استفاده از senet

گروهی خبری Usenet در شبکهٔ اینترنت یکی از جولانگاههای نفوذگران برای جستجوی اطلاعات خرد و کلان در مورد شبکهٔ هدف محسوب می شود. کارکنان یک

شبکه ممکن است برای پاسخ به سؤالات کاربران یا به اشتراک گذاشتن اطلاعات خود، در گروههای خبری Usenet حضور داشته باشند. این محیط فضای خوبی جهت بیرون کشیدن اطلاعات حساس در مورد شبکهٔ هدف می باشد. مثلاً ممکن است یکی از کارمندان شبکه هدف در پاسخ به سؤال یک کاربر، جوابی مفصل و تشریحی در مورد چگونگی پیاده سازی، پیکربندی و رفع اشکال از یک سرویس دهندهٔ خاص ارائه کند.نوذ گر از این پرسش و پاسخ متوجه می شود که احتمالاً در شبکهٔ هدف چنین سرویس دهنده ای نصب شده است. ممکن است شخص نفوذگر از یکی از کارمندان شبکهسؤالی بپرسصدتا بتواند حدسیات خود در مورد شبکه هدف را گسترش بدهد. یا آنکه نفوذگر در پاسخ به یکی از پرسشگران Usenet که سؤالی در مورد شبکهٔ هدف پرسیده است. جوابی کاملاً غلط ارائه می کند تا یکی از کارکنان شبکه آنرا بطور کامل، صحیح و تقصیلی جواب بدهد.

بهرحال پرسه زدن در Usenet همانند ماهیگیری در یک رودخانهٔ گل آلود است. می تواند نتیجهٔ چشمگیی داشته باشد و از طرفی ممکن است هیچ نفعی به نفوذگر نرساند.

برای جستجوی گروههای خبری می توان از موتور جستجوی Googleاستفاده کرد:

http://groups.google.com

این موتور جستجو حجم عظیمی ار گروههای خبری را در بایگانی خود درج کرده و با روش بسیار مناسب و ساده ای روی آن به جستجو می پردازد. (در سال ۲۰۰۱،Google،

سایت مشهور و جهانی Dejanews را تصاحب و در خود ادغام کرد تا به قدرتمندترین موتور جستجوی گروههای خبری تبدیل شود.)

بانک اطلاعاتی Whois

پس از جستجو در وب. نفوذگر به سراغ بانکهای اطلاعاتی Whois در اینترنت می رود. این بانکهای اطلاعاتی می توانند در مورد آدرسهای اینترنت (IP)، نامهای حوزه (Domain Name) و روش برقراری تماس با یک فرد مسئول در شبکهٔ هدف، اطلاعات قابل توجهی را ارائه نمایند.

«آدرس حوزه» یک ماشین یا یک گروه از ماشینها را روی اینترنت مشخص می کند. بعنوان مثال ww.microsoft.coml آدرس ماشینی است که نقش سرویس دهندهٔ وب شرکت مایکروسافت را ایفا می کند، در حالی که Microsoft.com گروهی از ماشینها را مشخص می کند که همگی متعلق به شرکت مایکروسافت هستند. این ماشینها همگی دارای نامی هستند که در آخر نامشان Microsoft.com ظاهر می شود:

machine-name.microsoft.com

در برخی از کتابها مثل ww.microsoft.com یک «زیر دامنهٔ» یا «زیرحوزه» از آدرس حوزه microsoft.com گفته می شود. وقی موسسهٔ شما یک شبکهٔ کامپیوتری جهت سرویس دهی در اینترنت ایجاد می کند که شامل سرویس دهنده های وب، پست الکترونیکی ، FTP و یا هر سرویس دیگر است مجبور خواهد بود تا حداقل یک آدس حوزه (مثل Yourcompany.com) ثبت نماید این آدرس جهانی به مجموعهٔ تمام

ماشینهای شبکه شما اشاره می کند نه یک ماشین خاص. برای آنکه شما تک تک ماشینهای شبکهٔ خود را نامگذاری کنید باید ابتدا یک سرویس دهندهٔDNS بر روی یک ماشین از شبکه نصب کنید و بر روی آن سرویس دهنده اسامی ماشینها و آدرس IP آنها را تعریف نمائید. آدرس این سرویس دهندهٔ DNS ، در بانک اطلاعات جهانی DNS که آدرس را از طریق آنها ثبت داده اید درج می شود. هر کسی در هر کجای دنیا بخواهد آدرسی مثل xxx. Yourcompany.com را به آدرس IP تحلیل و تبدیل (Resolve) کند از طریق سرویس دهنده های جهانی DNS به آدرس ماشین DNS شبکهٔ شما ارجاع داده می شود. آن ماشین هم تحت کنترل و نظارت شماست و هر نامی که به Yourcompany.com ختم شده باشد باید روی مشین DNS محلی، تعریف شود.

به محض آنکه یک آدرس حوزه را از طریق ثبت دهنده های جهانی برای سازمان یا شرکت خود به ثبت می رسانید تمام اطلاعات مربوط به آن نام، بصورت خودکار در تمام سازمان یا شرکت خود به ثبت می رسانید تمام اطلاعات مربوط به آن نام، بصورت خودکار در تمام بانکهای اطلاعاتی Whois در شبکه اینترنت بار خواهد شد. با توجه بدانکه در هنگام ثبت آدرس، اطلاعاتی راجع به نام شرکت، مسئل شبکه، آدرس و شماره تلفن از شما دریافت می شود و این اطلاعات در بانکهای اطلاعاتی whois درج خواهد شد لذا هر کسی که اراده کند می تواند با مراجعه به سرویس دهنده های Whoisاین اطلاعات را بدست بیاورد.

قبل از آنکه چگونگی استاده از بانکهای اطلاعاتی Whois را بیاموزیم بایستی ببینیم «ثبت دهنده های آدرس حوزه Domain Name Registrar » چه نقشی را در شبکهٔ اینترنت ایفا می کنند.

وقتی شما می خواهید یک آدرس حوزه با پسوندهای .com , .net یا .org را در اینترنت ثبت بدهید باید به یکی از سایتهای ثبت دهندهٔ نام مراجعه کرده و پس از پرداخت وجه مربوطه، نام مورد نظر خود را تعیین کنید. این سایتها بررسی خواهند کرد که نام انتخابی شما متعلق به دیگران نباشد، سپس با گرفتن مجموعه ای از اطلاعات نام انتخابی شما را در «سروس دهنده های جهانی نام» وارد می نمایند و از آن به بعد نام شما در جهان شناخته شده خواهد بود. سرویس دهنده های Whois نیز مشخصات عمومی شما را در بانک اطلاعاتی جهانی و همگانی خود درج می نمایند.

هنگامی که نفوذگر بخواهد شبکه شما را شناسائی کند Whois تمام این اطلاعات را بصورت قانونی و رایگان تقدیم او خواهد کرد. البته جستجو بدنبال اطلاعات مربوط به یک نام حوزه بستگی به آن دارد که آن نام دارای پسوندهای .com , .net یا .org است یا آنها پسوندهای کشوی مثل .ir یا .nl یا پسوندهای غیرعمومی مثل .mil دارد. جستجو در مورد این دو دسته نام اندکی متفاوت است.

استفاده از سایت ARIN جهت تحقیق در مورد آدرس IP

به غیراز اطلاعاتی که از طریق سایتهای ثبت دهندهٔ نام در اختیار نفوذ گر قرار می گیرند، او می تواند برای کسب اطلاعات بیشتر به سایت ARIN مراجعه کرده و در مورد

آدرسهای IP اختصاص یافه به شبکهٔ هدف تحقیق جامع تری به عمل بیـاورد. بگونـه ای که در شکل (٦-٥) نشان داده شده است ، سایت ARIN[1] سروسیس مشابه Whois برای جستجوی اطلاعات در اختیار کاربران قرار می دهد در این سایت شما می توانید بفهمیـد که مثلاً یک آدرس IP متعلق به چه شرکت یا سـازمانی اسـت ARIN jlhl Hnvsihd IP تخصیص داده شده به یک شرکت یا موسسهٔ خاص را به کاربران ارائه می کند. سـرویس جسـتجوی ARIN در آدرس http://www.arin.net/whois/arinwhois.html/ در اختیار همگان قرار گرفته است. البته باید اشاره کنیم که ایـن سـایت فقـط آدرسـهای IP تخصیص داده شده به شرکتها و موسستی را که در محدوده نواحی زیر هستند، در اختیار کاربران قرار می دهد:

- منطقهٔ آمریکا (شمالی و جنوبی)
- منطقهٔ Caribbean
- منطقه صحرای آفریقا

برای یافتن آدرسهای IP تخصیص داده شده در اروپـا بایـد بـه سـایت RIPE NCC[2] در آدرس زیر مراجعه کنید:

http://www.ripe,net/

مشخصات آدرسهای IP تخصیص داده شده در آسیا، در سایت APNIC[1] با آدرس ذیـل در دسترس می باشد:

---

[1] American Registry for Internet Numbers
[2] R Seaux IP Europ ens Network Coordination Centre

http://www.apnic.net/

سیستم DNS

DNS یا «سیستم نامگذاری حوزه»، یک روش سلسله مراتبی است که بانک اطلاعاتی مربوط به نامهای نمادین حوزه و معادل IP آنها را روی کل شبکهٔ اینترنت توزیع کرده است و هر ایستگاه می تواند در یک روال منظم و سلسله مراتبی آدرس IP آنها را روی کل شبکهٔ اینترنت توزیع کرده است و هر ایستگاه می تواند در یک روال منظم و سلسله مراتبی آدرس IP معادل با ایستگاه مورد نظرش را در نقطه ای از شبکهٔ اینترنت پیدا کند؛ این سیستم در سال 1984 معرفی شد.

در DNS، کل آدرسهای اینترنت درون بانکهای اطلاعاتی توزیع شده ای هستند که هیچ تمرکزی روی نقطه ای خاص از شبکه ندارند. روش ترجمهٔ نام بدین صورت است که وقتی یک برنامهٔ کاربردی مجبور است برای برقراری یک ارتباط، معادل آدرس IP از یک ماشین با نامی مثل cs.ucsb.edu را بدست بیاورد، قبل از هر کاری یک تابع کتابخانه ای «تابع تحلیگر نام»[۲] گفته می شود. تابع تحلیلگر نام، یک آدرس نمادین را که بایستی ترجمه شود، بعنوان پارامتر ورودی پذیرفته و سپس یک بستهٔ درخواست[۳] به روش UDP تولید کرده و به آدرس یک سرویس دهندهٔ DNS را در اختیار داشته باشند. این «سرویس دهندهٔ DNS (که به صورت پیش فرض مشخص می باشد) ارسال می کند.

---

[1] Asia Pacific Network Information Cnter
[2] Name Resolver
[3] Query Packet

همهٔ ماشینهای میزبان، حداقل باید آدرس IP از یک سرویس دهندهٔ DNS را در اختیار داشته باشند. این «سرویس دهندهٔ محلی[1]» پس از جستجو، آدرس IPمعادل با یک نام نمادین را بر می گرداند. «تابع تحلیلگر نام» نیز آن آدرس IP را به برنامهٔ کاربردی تحویل می دهد. با پیدا شدن آدرس IP، برنامهٔ کاربردی می تواند عملیات مورد نظرش را ادامه بدهد.

همانگونه که اشاره شد بانک اطلاعاتی که اسامی حوزه اینترنت را تعریف کرده، متمرکز نیست بلکه روی کل اینترنت توزیع شده است. حال باید دید اسامی اینترنت چگونه سازماندهی میشود تا نهایتاً بتوان روش جستجو روی یک بانک اطلاعای توزیع شده را توضیح داد. اسامی نمادین زیرا در نظر بگیرید:

www.bristol.edu یا www.president.ir

بدیهی است که نامهای حوزه همانند مثالهای بالا بدون بدون مسمی و دلیل انتخاب نمی شوند بلکه اطلاعاتی ارزشمند برای جستجو در بانک اطلاعاتی توزیع شدهٔ نامهای نمادین در خود دارند. بگونه ای که مشهود است یک نام حوزه از چند بخش مجزا که با علامت «.» از هم تفیک شده، تشکیل می شود. هر کدام از این بخشها که «سطح» نام دارد به یک قسمت از بانک اطلاعاتی توزیع شده اشاره می نماید که به محدوتر شدن فضای جستجو کمک می کند.

[1] Local DNS Server

برای تحلیل یک نام حوزهٔ، سطوح از سمت راست به چپ تفکیک می شوند و در یک روند سلسله مراتبی ، سرویس دهندهٔ متناظر با آن سطح پیدا می شود. فعلاً از بالاترین سطح که در سمت راست نام حوزه قرار می گیرد شرع می کنیم. نامهای حوزه به هفت منطقهٔ عمومی و حدود صد و اندی منطقه کشوری تقسیم بندی شده است. حوزه سطح بالا بدین معناست که شما با یک نگاه ساده به انتهای آدرس نمادین، می توانید ماهیت آن نام و سرویس دهنده متناظر با آن را حدس بزنید. یعنی اگر انتهای نامهای حوزه متفاوت باشد منطقهٔ جستجو برای یافتن آدرس IP معادل نیر متفاوت خواهد بود. هفت حوزه عمومی که همهٔ آنها سه حرفی هستند عبارتند از:

org, net, mil, int, gov, edu, com

نامهای حوزهٔ بسیار زیادی در اینترنت تعریف شده اند که هیچیک از حوزه های سه حرفی هفتگانه را در انتهای آنها نمی بینید. معمولاً در انتهای این آدرسها یک رشتهٔ دو حرفی مثل ir. یا nl. قرار گرفته است. این رشتهٔ دو حرفی مخفف نام کشوری است که آن آدرس و ماشین صاحب آن نام، در آن کشور واقع است.

هر حوزه می تواند به زیر حوزه های کوچکتری تقسیم می شود: ac-ip و co-jp. که اولی یک موسسهٔ علمی و دانشگاهی و دومی یک موسسهٔ بازرگانی یا تجاری را در ژاپن تعیین می نماید؛ یعنی محل جستجو برای ترجمهٔ یک نام متفاوت خواهد بود. بعنوان مثال آدرس زیر بسادگی قابل تحلیل است.

کشور : ژاپن

هویت: دانشگاهی

نام دانشگاه: کامپیوتر

برای سادگی در درک حوزه و زیر حوزه به نمودار زیر دقت کنید:

در شکل قبل چگونگی شکسته شدن یک آدرس به زیر حوزه های کوچکتر به تصویر کشیده شده است. با این ساختار برای ترجمهٔ یک نام حوزهٔ مثل robotic.ai.cs.sharif.edu ، عملیات از فایلی به نام «ریشه» شروع می شود؛ سپس آدرس ماشینی که فایل راهنمای edu. در آنجا واقع شده بدست می آید با مراجعه به چنین فایلی مجدداً آدرس ماشینی که فایل راهنمای sarif.edu.. در آنجا قرار دارد به دست می آید؛ این روند تا رسیدن به آدرس IP معادل ادامه می یابد. این عملیات در چند مرحلهٔ محدود تکرار می شود و با توجه به آنکه از پروتکل UDP استفاده می شود، تاخیر بحرانی نخواهد داشت.

دقت کنید شما نمی توانید هرنام دلخواه را برای شرکت یا سازمان خودتان انتخاب نمائید بلکه برای اینکار باید نام مورد نظر را ثبت نمائید؛ در غیر این صورت چنین آدرسی در اینترنت هویت نخواهد داشت. برای ثبت آدرس باید به سایتهای ثبت کننده نام حوزهٔ مراجعه کرده و تقاضای ثبت آدرس نمائید . این مؤسسات ضمن ثبت و درج نام در بانک اطلاعاتی یکی از حوزه های سطح بالا، تضمین خواهند کرد که نام انتخابیتان در کلی شبکهٔ اینترنت منحصر بفرد باشد.

کسب اطلاعات از سرویس دهندهٔDNS در راستای حمله

در بخش های قبلی آموختید که سیستم DNS حاوی اطلاعات بسیار میدی است که متاسفانه گاهی در خدمت نفوذگر قرار می گیرد. بطور معمول نفوذگر برای شناسائی مقدماتی شبکهٔ هدف بوسیله DNS، مراحل زیر را دنبال می نماید:

او ابتدا نیاز دارد تا حداقل یک سرویس دهندهٔDNS را در شبکهٔ هدف پیدا کردن سرویس دهندهٔ DNS از یک شبکهٔ (بگونه ای که تشریح شد) بسادگی و از طریق سرویس Whois در اینترنت امکان پذیر است. بعنوان مثال مطابق با شکل (۵-۵) سرویس دهنده های DNS از شبکهٔ Security.com با آدرسهای IP زیرمعرفی شده اند:

216.57.130.1 (سرویس دهندهٔDNS اصلی)

216.57.120.2 (سرویس دهندهٔDNS ثانویه) و سرویس دهندهٔ DNS سوم و چهارم سرویس دهندهٔ اولیه و سرویس دهنده ثانویه تفاوتی ندارند؛ سرویس دهنده ثانویه، بعنوان پشتیبان سرویس دهندهٔ اولیه، قابلیت اعتماد شبکه رابالا می برد و در مواقعی که سرویس دهندهٔ اصلی مختل شود سرویس دهندهٔ دوم در اختیار کاربران اینترنت قرار می گیرد.

برای استخراج اطلاعات لازم از سرویس دهندهٔDNS نفوذگر باید از ابزارهای خاص استفاده کند.یکی از ابزارهای عمومی و ساده در سیستمهای عامل یونیکس و windos برنامهٔ nslookup است که در خطر فرمان اجرا می شود. نفوذ گر بسادگی فرمان

nslookup را در خطرمان تایپ کرده و کلید Enter را فشار می دهد. پس از اجرای این برنامهٔ، نفوذگر می توند با سرویس دهندهٔDNS فعل و انتقال داشته باشد.

در مرحلهٔ بعدی نفوذگر تلاش می کند تا از طریق nslookup اقدام به «دریافت کل اطلاعات یک Zone »[1] نماید. بدین معنا که تمام رکوردهای موجود در ارتباط با یک نام حوزه منتقل شود. Nslookup از DNS متلق به یک شرکت یا موسسه تقاضا می کند تا تمام رکوردهائی که در خصوص یک نام حوزه در بانک اطلاعاتی ذخیره شده است، برایش ارسال نماید. بدین منظور پس از اجرای nslookup باید از طریق فرمان server، نام سرویس دهندهٔ مورد نظر تعیین شود:

server [نام سرویس دهندهٔهدف]

برای آنکه nslookup انتقال کل رکوردهای موجود در خصوص یک نام حوزه را تقاضا کند، باید در خطر فرمان از فرمان زیر استفاده شود:

set type = any

سپس برای آنکه انتقال رکوردهای موجود در DNS هدف شورع شود باید فرمان زیر در خط فرمان صادر شود:

is-d [نام حوزهٔ مورد نظر]

[1] Zone Transfer

پس از اجرای این فرمان رکودهای ارسالی توسط DNS هدف روی خروجی نشان داده می شود در مثال زیر مراحل استفاده از از nslookup جهت انتقال کامل رکوردهای موجود در خصوص نام skoudissuff.com نشان داده شده است:

$ nslllokup

Default server: evil.attacker.com

Address: 10.200.100.45

Server 10.1.1.34

Default server: ns.skouisstuff.com

Address: 10.1.1.34

Set type=any

Is-d skouisstuff.com

Susyeml	ID	IN	A	IO.1.1.36
	ID	IN	HINFO	Solaris 2.6 Mailserver
	ID	IN	MX	10 maill
Web	ID	IN	A	10.1.1.48
	ID	IN	HINFO	“NTSWWW”
ntftp	ID	IN	A	10.1.1.49

ws	ID	IN	A	10.1.1.22
	ID	IN	TXT	“Adminisrator workstation”

(در مثال بالا تعداد خروجیها خلاصه شده اند تا خواناتر باشند.)

حال اگر به دقت رکوردهای نمایش داده شده را بررسی کنید اطلاعاتی مهمی در آن خواهید دید این اطلاعات برای نفوذگر بسیار با ارزش است:

- system1: رکودهای اول تا سوم از خروجی مشخص کنندهٔ آنست که ماشینی با آدرس 10.1.1.36، سرویس دهندهٔ پست الکترونیکی و سیستم عامل آن Solaris 6.2 است.
- Web: رکورد چهارم و پنجم از خروجی مشخص کنندهٔ آنست که ماشینی با آدرس 10.1.1.48 نقش سرویس دهندهٔ وب را بازی می کند و سیستم عامل آن Windows NT 4.0 است.
- Ntftp: رکورد ششم از خروجی نشان می دهد که سرویس دهندهٔ FTP بر روی ماشینی با آدرس 10.1.1.49 نصب شده و احتمالاً سیستم عامل آن Windows NT است . (ntftp NTFTP→)
- Ws: دو رکورد آخر از خروجی نشان می دهد که ماشین با آدرس 10.1.1.22 متعلق به مسئول شبکه است.

بدین ترتیب نفوذگر اطلاعات مورد نیاز در مورد شبکه بدست آورده است.

به غیر از برنامهٔ nslookup سه برنامهٔ زیر نیز برای بهره برداری از اطلاعـات یـک DNS موجود می باشد:

- دستور host موجود در انواع سیستمهای عامل سازگار با یونیکس
- دستور dig موجود در انواع سیستمهای عامل سازگار با یونیکس
- Adig: برنامه جالبی برای محیطهای windows(شامل نسـخهٔ XP,2000,NT,9x)

این برنامه در آدرس زیر عرضه می شود:

http://nscan.hypermart.net/index.cgi?index=dns

ابزار Sam spade

نرم افزار مجتمع و پکپارچهٔ Sam spade ، ابزاری سـاده بـرای شناسـائی مقـدماتی یـک شبکه محسوب می شود. این ابزار رایگـان کـه توسـط Steve Atkins نوشـته شـده، در آدرس زیر در دسترس عموم قرار گرفته است:

http://www.samspade.org/ssw

Sam spade بگونه ای که در شکل (۱۰-۵) مشـاهده مـی شـود، دارای امکانـات بسـیار متنوعی برای شناسائی شبکه می باشد. این ابزار بـا یـک پنجـرهٔ گرافیکـی زیبـا بـر روی سیستمهای عامل windows (WX,2000,NT,9x) قابل اجرا است.

قابلیتهای برجسته Sam spade در زیر فهرست شده است:

Whois: ابزار Sam spade برای شناسائی یک آدرس حوزهٔ، بطور مسـتقیم بـا سـرویس دهنده whois ارتباط برقرار کرده و نتیجه را در اختیار کاربر قرار می دهد. این ابزار یـک

سرویس دهندهٔ whois را بصورت پیش فرض می شناسد ولی کاربر قادر است تعیین کند که از کدام سرویس دهندهٔ whois در اینترنت استفاده شود. تبلیغات پیش فرض Sam spade کاملاً مناسب است و برای جستجوی نامهائی که دارای پسوند .net,.com و org هستند هیچ تغییر خاصی در سرویس دهندهٔ پیش فرض آن نیاز نیست.

IP Block Whois: Sam spade قادر است تعیین کند که مجموعه ای از آدرسهای Ip (در محدودهٔ مشخص) متعلق به کدام سازمان یا موسسه است. Sam spade بطور پیش فرض سرویس دهندهٔ Whois و بانک اطلاعاتی ARIN استفاده می کند.

NS Lookup: این مشخصه به کاربر اجازه می دهد که مستقیماً با سرویس دهندهٔ dns فعل و انفعال داشته باشد و از DNS در خصوص یک نام حوزهٔ پرس و جو نماید.

DNS Zone Transfer: این مشخصه کاربر را قادر می کند تا بتواند تمام رکوردهای موجود درون یک DNS را در خصوص یک نام حوزه، منتقل نماید.

Ping: این ابزار یک بستهٔ ICMP Echo Request به سمت یک ماشین در شبکه ارسال می نماید و منتظر پاسخ ICMP Echo Reply می ماند . در صورت برگشت پاسخ اعلام می کند که آن ماشین در شبکه موجود و فعال است . (در مورد عمل ping در فصل بعدی توضیح داده خواهد شد)

Dig: این مشخصه برای دریافت اطلاعات تفصیلی از DNS پیرامون یک سیستم خاص در شبکه کاربرد دارد.

Traceroute: این مشخصه فهرستی از مسیریابهای موجود بین یک ماشین مبداء و ماشین مقصد را پیدا کرده و در خروجی نشان می دهد. در این خصوص نیز در فصل بعد مفصلاً توضیح خواهیم داد.

Finger: از طریق این مشخصه می توان فهرست کاربران حاضر و فعال در یک سیستم را بدست آورد. (بشرطی که سرویس دهنده Finger روی آن ماشین اجرا شده باشد)

SMTP VRFY: از طریق این مشخصه می توان بررسی کرد که « آیا یک آدرس e-mail روی یک سرویس دهندهٔ پست الکترونیکی معتبر و تعریف شده است یا خیر؟» این قابلیت بر اساس پروتکل SMTP کار می کند.

Web Browesr: ابراز Sam spade یک مرورگر وب درونی بهمراه دارد که می تواند صفحات وب را بصورت کدهای خام HTML (بهمراه سرآیند ارسالی از پروتکل HTTP) به کاربر نشان بدهد . از این مشخصه می توان برای حمله به برنامه های کاربردی تحت وب استفاده کرد. (در این خصوص در فصل هفتم توضیح خواهیم داد)

به غیر از Sam spade ، نرم افزار NetInfo نیز برای شناسائی شبکه معرفی شده که فقط برای سی روز رایگان است. این ابزار از بسیاری جهات مشابه Sam spade است ولی مشخصه هائی نسبت به Sam spade بیشتر و مشخصه هائی نیز کمتر دارد. پنجرهٔ اصلی این نرم افزار در شکل (۱۱-۵) نشان داده شده است. این ابزار در ارد مقایسه با ابزار Sam spade معرفی می کنیم:

Local Info: مشخصات ماشین محلی که این ابزار در حال حاضر برروی آن اجـرا شـده را برمی گرداند. این مشخصات شامل موارد زیر است:

- نام کاربر
- آدرس IP ماشین
- نسخهٔ Win Sock و مشخصات آن
- حالت سیستم
- تعداد سوکتهای ازاد
- اندازهٔ بستهٔ UDP
- Connections : مشخصات تمام ارتباطات TCP ، پورتهـای بـاز UDP و حالـت ارتباطات موجود روی ماشین را نشان می دهد.
- Ping: این مشخصه دقیقاً عملکردی مشابه با Ping در ابزار Sam spade دارد.
- Trace: عملکرد این مشخصه مشابه با Traceroute در ابزار Sam spade است.
- Lookup: مشابه با NS Lookup در Sam spade
- Finger: مشابه با مشخصهٔ Whois در Sam spade
- Whoix: مشابه با مشخصهٔ Whois در Sam spade
- Day Time: این مشخصه سعی می کند تا از سرویس دهندهٔ Day-Time کـه نـام آن را کاربر مشخص می نماید ساعت و تاریخ فعلی را بدست آورده و نمایش بدهد. (بشرط آنکه روی ماشین مشخص شده سرویس دهندهٔDay Time اجرا شده باشد).

- Time: برای بدست آوردن زمان از یک سرویس دهنده که نام آنرا کاربر معین خواهد کرد.
- HTML: مشابه با گزینهٔ Web Browser در Sam spade
- Scannr: با این مشخصه می توان مجموعه ای از ماشینها را از لحاظ روشن و فعال بودن بررسی کرده مثلاً اگر کاربر بخواهد بررسی کند که کدامیک از ۲۵٤ ماشین با آدرسهای 192.153.18.x فعال هستند از این مشخصه استفاده می کند.
- Services: این مشخصه قدرتمند، با گرفتن نام یک ماشین، سرویسهای معرف و شناخته شده ای که آن ماشین ارائه می کند را فهرست خواهد کرد.
- به غیر از Sam spade،نرم افزارهای زیر نیز برای شناسائی شبکهٔ هدف کاربرد دارند:
- Cyber Kit: یک ابزار رایگان تحت ویندوز که در آدرس زیر قابل تهیه است:

http://www.cyber Kit.net/index.html/

- Net Scan Tools: یک ابزار ۲۵ دلاری که برای سی روز اول به منظور ارزیابی آن، رایگان است و در آدرس زیر قابل تهیه می باشد:

http://.www.netscantools.com/nstmain.html

- INet Tools: یک نرم افزار شناسائی شبکه بر روی ویندوز و کمپنتاش که در آدرس زیر در دسترس عموم قرار گرفته است.

http://www.wildpackets.com/products/inettools

گامی دوم: پویش و جستجو در شبکه بندل رخنهٔ نفوذ

در مرحلهٔ شناسائی مقدماتی، نفوذگر مقداری اطلاعات پایه و حیاتی در ارتباط با شبکهٔ هدف گردآوری کرده است. این اطلاعات شامل موارد زیر است:

- تعداد شماره تلفن از خطوط دسترسی به شبکه
- تعدادی آدرس IP از شبکهٔ هدف و ماشینهای هدف
- تعدادی آدرس حوزهٔ (Domain Name)
- اطلاعات کلی و عمومی پیرامون نوع خدمات شبکهٔ هدف

نفوذ گر در فاز پویش (Scanning) بررسی های دقیق و عمیق خود را متوجه شبکهٔ هدف،ماشن یا ماشینهای هدف و ورودیهای باز و نقاط ضعف می نماید.

مسئولین شبکه باید بدقت با مفاهیم این فصل آشنا باشند زیرا از دیدگاه امنیت سیستم پیشگیری از حمله مقدم بر مقابهٔ بعد از وقوع آن است . اگر مسئول شبکه آگاهی هاس لازم را داشته باشد قادر است پارامترهای شبکهٔ خود به گونه ای تمظیم کند تا نفوذگران به اشتباه بیفتند یا تلاش آنها برای شروع حمله عقیم بماند. لذا مسئولین شبکه را به فراگیر روشهای پویش، راه حلهای پیشگیری و مقابله و همچنین آشنائی با نرم افزارهای پویش دعوت می کنیم.

در جستجوی مودمهای شبکه

مودم از رایجترین ابزارهای شخت ازاری مورد استفاده در شبکه هاست که به کاربران امکان می دهد تا از طریق خط تلفن اقدام به مبادلهٔ داده بین دو ماشین بنمایند. نفوذگر در

اولین مرحلهٔ پویش و جستجوی رخنه های نفوذگر را از یک در پنهان به دورن شبکه هدایت و راهنمایی می نماید!

در این مرحله نفوذگر با در اختیار داشتن فهرستی از شماره های تلفن که از مرحلهٔ «شناسائی مقدماتی» بدست آورده، برای یافتن مودمهای آسیب پذیر شروع به پویش آنها می کند! پویش مودمها را هدف یافتن یک راه نفوذ طبق مکانیزم زیر انجام می شود:

نفوذ در مجموعه ای از شماره های تلفن را که احتمال می دهد متعلق به شبکهٔ هدف حمله است مشخص می نماید . او مجبور است این شماره ها را درون یک فایل ذخیره کند(که اندکی وقت گیر است) یا بسادگی یک محدوده از شماره ها را تعیین نماید. (مثلاً شماره ها 98760000 تا 9876999 معادل هزار شمارهٔ تلفن).

سپس برای پویش این خطوط تلفن، از یک ابزار خودکار استفاده می کند تا در روانی پی در پی با این شماره های تلفن تماس برقرار شود و به محض تشخیص یک سیگنال حامل (Carriwe) متعلق به مودم، شمارهٔ تلفن مربوطه در جائی ذخیره گردد. این روال ممکن است چند ساعت وقت بگیرد ولی برای یک نفوذگر حرفهای ارزش آن را دارد! انجام این مرحه قطعاً نیازمند به ابزار خودکار است که در ادامه مهمترین این ابزارها را تشریح خواهیم کرد.

پس از یافتن شماره های تلفنی که به مودم متصل هستند، نفوذگر از طریق یک ابزار نرم افزاری روی این مودمها تمرکز می کند تا:

- نوع مودم و پروتکل ارتباطی آنرا تشخیص بدهد.

- سرویس دهندهٔ آن مودم را بشناسد.
- کلمهٔ عبور برای وصل ارتباط آن مودم را بدست بیاورد.
- نفوذگر در صورت پیدا کردن مدم آزاد و متصلی که بتوان از طریق آن بـه یکـی از ماشینهای شبکه متصل شد، مراحل بعدی حمله را طرح ریزی می کند.

دو اصطلاح در دنیای نفوذگران

War Dialer: جستجو در بین مجموعه بسیار عظیمی از شماره هـای تلفـن بـرای یـافتن مودمهای متصل و فعال در شبکه یا ماشین هدف

Demon Dialer: حمله برعلیه یک شمارهٔ تلفن (که اتصال آن به مودم محرز شده است) برای یافتن کلمهٔ عبور و راهی جهت نفوذ به ماشینی که به آن مودم متصل است.

برای تمام افردی که با مودم آشنا هستند این امر محرز است که: « یـک مـودم کـه بـا آن تماس برقرای می شود به یک سرویس دهنده نیاز دارد تا آن سرویس دهنده مـن دسـتور وصل ارتباط. هویت کاربر راه دور را تشخیص داده و پس از احراز هویت به او سرویس بدهد.» اگر چه ممکن است شمار روی کامپیوتر خانگی خود مودم داشته باشید ولی اگـر کسی از طریق مودم با شماره تلفن شما تماس بگیرد ارتباطش برقرار نخواهـد شـد زیـرا شما سرویس دهنده ای که ارتباط را برقرار و به او سرویس بدهـد روی کـامپیوتر خـود نصب و اجرا نکرده اید.

معمولاً برای سرویس دهی با مودم (از راه دور) از نرم افزارهای مشهوری مثل:

- Symantec’s pc Anywhere: (موجود بر روی CD جانبی کتاب)

- Laplink
- ConrlIT

استفاده می شود که هر کدام از آنها در صورت پیکربندی نادرست، قش یک خائن ستون پنجمی را روی سیستم ایفاء می کنند!

این ابزارهای نرم افزری به کاربران راه دور اجازه می دهد تا با یک ماشین ارتباط برقرار کرده و همانند یک کاربر معمولی و در محل از آن سرویس بگیرند، به سیستم فایل آن ماشین سرکشی کنند، از منابع اشتراکی آن ماشین(مثل چاپگر یا اسکنر) استفاده کنند یا برنامه ای را اجرا نمایند.

برخی از این سرویس دهنده ها گاهی بشدت خطرناک هستند زیرا آنها بطور پیش فرض نیاز به هیچ کلمهٔ عبوری ندارند و به محض برقراری ارتباط، سرویس دهی به طرف مقابل آغاز می شود.

PcAnywhere همینگونه است و کافی است تا بدون دقت روی ماشینی نصب شده باشد. در این حالت هر کسی از طریق مودم قادر است آن ماشین را تصاحب کرده و هر کاری روی آن انجام بدهد.

دقت داشته باشید که برخی از شبکه ها (بالاخص ISP های عمومی یا خصوصی) صدها خط تلفن و مودم دارند. قابل نفوذ بودن یکی از آنها امنیت کل شبکه را به خطر خواهد انداخت. متاسفانه کم نیستند مسئولان شبکه و کارکنانی که از بین صدها خط تلفن خریداری شده برای شبکه، یک خط مستقیم را به خود اختصاص داده اند و آنرا از طریق

مودم به ماشینشان (همچنین یک دستگاه گوشی تلفن زیبا!!) متصل کرده اند تا اولاً بتوانند کارهای شبکه را از خانه خود پیگیری کنندو در مواقع لزوم کری را از راه دور انجام بدهند؛ ثانیاً در اختیار گرفتن یک خط مستقیم برای پیگیری امور اداری شبکه انگیزهٔ خوبی است تا به این بهانه از آن استفادهٔ شخص کنند!

برای شبکه های مهم و حساس، این خطوط مستقیم (با مودمهای فعال) یک خطر امنیتی بسیار بزرگ محسوب می شود، حتی اگر برای استفاده از آنها کلمهٔ عبور در نظر گرفته شده باشد.

نکتهٔ ایمنی:

گزارشی وجود دارد که در آزمایش نفوذپذیری یک شبکهٔ صدها هزار دلاری که از ابزارهای پیشرفته ای مثل دیوار آتش، سیستم IDS و سرویس دهنده های امن استفاده شده بود، هیچ رخنهٔ نفوذی پیدا نشد مگر « یک شمارهٔ تلفن آزاد و متصل به مودم یک ماشین در شبکهٔ داخلی!»

یک خط آزاد و متصل به مودم (در شبکهٔ داخلی) تاثیر تمام ابزارهای پیشرفتهٔ امنیتی را از بین خواهد برد.

در ادامهٔ بخش باید ابزارهای جستجوی مودم را معرفی نماییم. قبل از آن یادآوری می کنیم که طبق اصولی که در فصل پنجم اشاره شد نفوذگر مجموعه ای از شماره های تلفن را با استفاده از روشهای زیربرای شروع پوش و جستجو بدست آورده و یادداشت کرده است:

- دفترچه راهنمای تلفن
- مراجعه به سایتهای وب شبکهٔ هدف حمله
- سرویس whois در اینترنت
- روشهای روان شناختی و مهندسی اجتماعی

معمولاً نفوذگر در ابتدا، صدها شمارهٔ تلفن را برای بررسی فهرست می کند و در این مورد فوق العاده دست و دلباز است چون بطور متوسط در هر ساعت می توان صد شماره تلفن را بررسی کرد. هرگاه این شماره تلفن متعلق به یک شخص و متصل به گوشی باشد هیچ اتفاقی نخواهد اتفاد. (ارتباط بصور خودکار قطع می شود.)

حملات بعد از پیدا شدن مودمهای فعال

فرض کنید نفوذ گر خطوط تلفن متصل به مودمهای فعال را از طریق ابزاری مثل THC-Scan پیدا کرده است. اقدام بعدی او چیست؟ بگونه ای که اشاره شد این ابزار، فهرست خطوط تلفن و احتمالاً نوع سرویس دهندهٔ مودم متصل به آن را شناسائی کرده و در یک فایل ثبت می نماید. نوع سرویس دهنده مودم از پیغام ارسالی یا علامت ورود (Prompt Login) که توسط آن ارسال می گردد جدس زده می شود. بعضی از سرویس دهنده ها پس از برقراری ارتباط، یک رشتهٔ کاراکتری برای خوش مدگوئی (Wellcome String) به مودم تماس گیرنده ارسال می نمایند که همین رشته ممکن است هویت آن سرویس دهنده را مشخص نماید. به ترتیب تشخیص نوع مودم و نوع سرویس دهنده کمک بزرگی به نفوذگر جهت اقدام برعلیه آن ماشین می نماید. به عنوان

مثال pc Anywhere بطور پیش فرض کلمهٔ عبور ندارد. اگر نفوذگر مودمی با این سرویس دهنده پیدا کند،کار تمام است و اهدال بعدی خود را پیگیری خواهد کرد چرا که هیچ مانعی بر سر راهش نیست. (در مورد اهداف احتمالی او پس کشف چنین مودمهائی در فواصل بعدی صحبت خواهیم کرد) اگر نفوذ گر با مودمی مواجه شود که برای ورود او کلمهٔ عبور تقاضا می کند، اقدام بعدی او حدس زدن کلمهٔ عبور و ورود به زور (Brute Froce) به آن سیستم است. او برای اینکار می تواند از ابزارهای خودکار استفاده کند یا آنکه بصورت دستی عمل نماید TG\HC Login Hacker یک ابزار جهت ورود به سیستمهائی است که از کاربر کلمهٔ عبور تقاض می نمایند. این ابزار در آدرس زیر عرضه می شود:

http://the.inferno.tusulum.edu/

بهر حال عمل کشف کلمهٔ عبور بیار وقتگیر خواه بود (چه بصورت خودکار و چه بصورت دستی) ولی تنها سرمایهٔ یک نفوذگر برای نفوذ به یک سیستم «زمان» است و احتمالاً ساعتها یا هفته ها برای اینکار هزینه خواهد کرد.

در مورد روشهای حدس و کشف کلمهٔ عبور، بطور مشروح در فصل بعدی صحبت خواهیم کرد ولی در اینجا بد نیست تا کلمات عبور رایج را که معمولاً افراد ناآشنا و سهل انگار بعنوان کلید رمز ورود به سیستم از راه دور بوسیلهٔ مودم بر می گزینند، معرفی کنیم:

- <blank>

- root
- sync
- bin
- nobody
- oprator
- manager
- admin
- administrator
- system
- days of the week
- COMPANY-NAME
- COMPANT-PRODUCT

هر کدام از آیتمهای فوق بایستی بعنوان کلمهٔ عبور یا مشخصهٔ کاربردی (User ID) امتحان شوند. پس از آزمودن مجموعهٔ بسیار گسترده ای از کلمات عبور (با صرف وقت کافی)، احتمال آنکه نفوذگر بتواند از راه دور به سیستم وارد شود وجود دارد. بالاخص هر گاه کلمهٔ عبور ضعیف ، کوتاه یا معنی دار انتخاب شده باشد. نفوذ از طریق مودمهای ناامن به معنای باز شده دروازهٔ سیستم بر روی نفوذگر برای عملیات بعدی اوست.

مقابله با نفوذ از طریق مودمهای نا امن

اگر شبکهای بواسطهٔ مودمهای ناامن یا فاقد کلمهٔ عبور مورد حمله قرار گیرد، هیچ توجیهی برای مسئول شبکه بالا سهل انگاری و سستی در مسئولیت ، پذیرفته نخواهد بود چرا که پیشگیری از چنین حملاتی چندان مشکل ایست. نکات زیر را به عنوان روشهای مقابله با نفوذ از طریق مودم توصیه می نمائیم:

سیاست مدون و مصوبی در مورد نحوهٔ استفادهٔ کارکنان از مودم اتخاذ کرده و بصورت صریح به آنها ابلاغ کنید. وجود خطوط مستقیم در یک سازمان شاید کارکنان را وسوسه کند که در برخی از زمانها از اشتراک اینترنت شخصی خود استفاده کنند و بالاخص مودم خود را برای پیگیری کارهای اداری و منزل فعال نگاه دارند.توصیهٔ موکد آنست که بهیچوجه اجازهٔ استفادهٔ پراکنده و شخصی از مودم را به هیچکس ندهید. اگر احتمالاً مجموعهٔ شما به تعدادی مودم احتیاج دارد آنرا بصورت متمرکز و یکجا تحت نظارت مستقیم خودتان (بعنوان مسئول شبکه) قرار بدهید تا امکان بازرسی آن بسادگی وجود داشته باشد.

اگر جهت ارتباط مستقیم با یک موسسهٔ دیگر مجبور به استفاده از مودم هستید نظارت و مسئولیت مستقیم مودم و سرویس دهندهٔ آن را برعهده گرفته و کلمات عبور بسیار مشکل و طولانی برای ورود از طریق آن در نظر بگیرید.

دقت کنید که اگر قرار باشد مورد یک تهاجم سازمان یافته و طراحی شده قرار بگیرد، احتمال آنکه خطوط تلفن متصل به مودم شما کشف شود بسیار زیاد است لذا در ذهن

خود فرض کنید این خطوط شناخته شده هستند و در عوض، ورود به آنرا به وسیلهٔ کلمهٔ عبور مناسب و سرویس دهنده های قوی غیرممکن کنید.

هر چند در ذهن خود خطوط متصل به مودم را کشف شده فرض می کنید ولی در هر حال تا موقعی که از مودمها بعنوان خطوط اختصاصی شبکه با یک موسسه یا یک شخص استفاده می شود شمارهٔ تلفن چنین خطوطی را حتی از کارکنان بخود مخفی نگه دارید.(بگذارید حداقل نفوذ گر برای کشف خط مودم وقت تلف کند و هزینه بپردازد.)

تنظیم خطوط تلفن مبتنی بر سیاستهای مدون:

فرض کنید که سیاست امنیتی موسسهٔ شما ایجاب می کند که استفاده از مودم بطور عام ممنوع باشد ولی چند کاربر بدلائل کاری مجبور هستند که از طریق مودم با بیرون شبکه ارتباط برقرار کنند. بهیچوجه خط ارتباط مستقیم به کسی اختصاص ندهید بلکه یک خط آزاد از طریق مرکز سوئیچ داخلی موسسهٔ خودتان (PBX) برای این منظور در نظر بگیرید. بگونه ای که فقط بتوان از دورن با بیرون ارتباط برقرار کرد و ارتباطات از بیرون به دورن وصل نشود؛ اینکار در مراکز سوئیچ تلفن دیجیتلای بسادگی میسر است. بدین نحو نفوذگر بهیچوجه قادر نخواهد بود با چنین مودمی ارتباط برقرار کرده و فعل و انفعال داشته باشد.

اگر یک کاربر نیاز به ارتباط دو طرفه داشته باشد (که آنرا شما تشخیص می دهید) باز هم خط لازم را از طریق PBX برقرار کنید و در ضمن بر آن نظارت مستقیم داشته

باشید، کلمهٔ عبور را بررسی کنید و آن را به انجام امور اداری و تعریف شده محدود کنید.

مودمهای خود را قبل از نفوذگر پیدا کنید: تمام سیاستهای امنیتی، محدود کردن خطوط تلفن، ابلاغ دستورالعمل و بخشنامه، ممکن است توسط کارکنان فرصت طلب یا سهل انگار یا بی توجهی مواجه شود و شماره قدرت اجرائی کافی برای برخورد با آنها را نداشته باشید . بهمین دلیل توصیه می کنیم ابزاری را که نفوذگر برای نفوذ به شبکهٔ شمار استفاده می کند، خودتان بر علیه شبکه بکار بگیرید تا مودمها و سرویس دهنده های آسیب پذیر را پیدا کنید و برای آن چاره ای بیندیشید. توصیهٔ ما آنست که هر سه ماه (یا ٦ ماه – بر اساس بزرگی سازمان و حساسیت آن) یکبار از THC-Scan بر علیه شبکهٔ خود استفاده کنید. چون شما محدودهٔ شماره های تلفن مجموعهٔ خود و تعداد آنها را می دانید، مشکل چندانی در انجام این کار نخواهید داشت. (فهرست شماره های تلفن را از مسئول مخابرات یا PBX خود سؤال کنید)

یکی از تمهیدات سخت گیرانه آنست که هراز گاهی از مرکز مخابرات فهرست شماره هائی که در طول ماه با آن شماره ها تماس گرفته شه است را گرفته و بدقت بازرسی کنید و برای شماره های نامشخص و مشکوک از کاربر مربوطه توضیح بخواهید.

سختگیری فیزیکی: اگر امنیت را در حد بالائی انتظار دارید، اولاً اجازهٔ ورود هیچ کامپیوتر کیفی با دستی و حتی تلفن همراه را به کارمندان خود ندهید. آنها باید مجبور باشند این ابزار را در هنگام ورود تحویل بدهند. ثانیاً هر از گاهی بصورت سرزده

کامپیوتر ها را میز به میز بگردید و آنها را از لحاظ اتصال به مودم بازرسی کنید. (قبل از آن باید دستور جمع آوری تمام مودمها صادر شده باشد) ثالثاً طبق معمول شبکه های دولتی و حساس ، تمام کامپیوترها را لاک و مهر کره و شماره گذاری کنید. هیچ کسی حق باز کردن درب کامپیوتر را به هر منظوری ندارد. رابعاً خطوط مستقیم در اختیار کارکنان قرار ندهید.

تعیین پورتهای باز بر روی یک ماشین

در این مرحله نفوذگر ماشینهای فعال شبکهٔ شما را و همچنین توپولوژی تقریبی آنرا می شناسد حال او می خواهد بداند هر ماشین چه وظیفه ای به عهده دارد و چه خدماتی را ارائه می کند. در ضمن هر کدام از این سرویسها به چه نحو در اختیار کاربران قرار می گیرند.

به گونه ای که در فصل مفاهیم TCP/IP اشاره شد، پورتهای باز و فعال TCP یا UDP روی هر ماشین سرویس هائی را که آن ماشین ارائه می دهد و پروسه هائی را که روی آن اجرا شده اند، مشخص می کنند.

در پروتکل TCP/IP، هر ماشین می تواند حداکثر ۶۵۳۵ پورت TCP و بهمین تعداد پورت UDP باز داشته باشد که البته بر روی یک ماشین فقط تعداد بسیار محدودی از آنها باز و فعال است؛ یعنی مجموع پروسه های در حال اجرا، فقط به تعداد محدودی از این پورتها گوش می دهند. (به تناسب سرویسی که ارائه می کنند)

هر پورت باز روی ماشین، یک درب ورودی به آن ماشین محسوب می شود. بعنوان مثال اگر شما یک برنامهٔ web server را اجرا کرده باشید، به پورت شمارهٔ ۸۰ گوش می دهد و بسته ای TCP با شماره پورت مقصد ۸۰ را می پذیرد. اگر شما یک سرویس دهندهٔ DNS داشته باشید پورت 53 UDP باز است؛ همچنین با نصب و اجرای سرویس دهندهٔ پست الکترونیکی ، پورت 25 TCP باز و فعال خواهد شد.

سرویس دهندهٔ های مشهور و استاندارد جهانی دارای پورتهای مشخص و معینی هستند. فهرست سرویس دهنده های استاندارد و شمارهٔ پورتهای آنها توسط IETF در RFC 1700 مشخص شده ودر اختیار عموم قرار گرفته است. (موجود بر روی CD جانبی کتاب)

اگر پورتهای UDP و TCP را بعنوان دربهای ورودی یک ماشین فرض کنیم عمل پویش پورت (port Scan) به منزلهٔ در زدن است؛ تا ببینیم آیا پروسه ای پشت این در هست یا خیر؟ اگر پروسه ای به آن پورت گوش بدهد پاسخی از آن بر خواهد گشت ولی اگر هیچ پروسه ای به آن شماره پورت گوش نکند هیچ جوابی بر نمی گردد.

تمام ابزارهای پورت (port Scan) می توانند فهرست مشخصی از شماره پورتها را بررسی و پویش نمایند و از باز یا بسته بودن آن آگاه شوند. برای آنکه وت زیادی گرفته نشود و همچنین نفوذگر لو نرود معمولاً مجموعهٔ مشخص و کوچکی از شمارهٔ پورتها بررسی و پویش می شود. این مجموعهٔ مشخص می تواند فهرست سرویسهای مشهوری مثل FTP HTTPm ,Telnet, SMTP و نظایر آن باشد.

عمل پویش صورت توسط نرم افزارهائی که بنام Port Scanner (پویشگر پورت) مشهورند انجام می شود. این ابزارها ذاتاً برای عملیات مدیریت و رفع اشکالات شبکه طراحی می شوند؛ ولی نفوذگر از آنها در راستای اهداف شوم خود بهره می گیرد!

برخی از این ابزارهای رایگان، در آدرسهای زیر در دسترس هستند:

- Nmap, by Fyodor, at http://www.insecure.org/Nmap
- Strobe, by Julian Assange, at http://packetstorm.securify.com/UNIX/scanners/
- Ultra Scan for windows NT/2000/98/XP,at http://packetsorm.securify.com/NT/scanners/
- Super Scan for windows NT/2000/98/XP , at http://members.home.come/rkeir/software.html/

قطعات به یک ابزار Port Scanner نیاز دارید:

ابزارهای پویش پورت برای اشکالزدایی از شبکه و بررسی های فنی و امنیتی مورد نیاز مسئول شکبه هستند. یک مجموعهٔ غنی از این ابزارها شامل چهار مورد فوق، بهمراه کدهای منبع برخی از آنها (به زمان C) بر روی دیسک ضمیمهٔ کتاب موجود می باشد. آدرس موقعیت آنها بر روی CD، در انتهای فصل مشخص شده است.

از بین تمام ابزارهای پویش پورت، نرم افزار Nmap بهترین و کاملترین ابزار محسوب می شود. این ابزار از مکانیزمهای متنوعی برای پویش پورت استفاده می کند که در نوع

خود بی نظیر است ودر هیچ ابزار دیگری بصورت مجتمع وجود ندارد. شکل (٦-٦) پنجرهٔ ابزار Nmap رانشان می دهد. در ادامهٔ این بخش مکانیزمهایی را که Nmap برای پویش پورت پیاده سازی کرده، به تفصیل بررسی خواهیم کرد.

مکانیزمهای مختلف پویش پورت

در این بخش مکانیزمهای گوناگون پویش پورتهای باز را بررسی می کنیم. (تمام این مکانیزمها در نرم افزار Nmap پیاده شده اند.)

بگونه ای که در فصل مفاهیم TCP/IP تشریح شد، برقراری ارتباط بین دو پروسه روی دو ماشین اتصالگرا (Connection Oriented) است و قبل از مبادلهٔ هر گونه داده باید یک ارتباط (اتصال) TCP به روش دست تکانی سه مرحله ای (3-Way Handshake) برقرار شود. به شکل (٦-٧) دقت کنید: ماشین Alice اقدام به برقراری ارتباط TCP با ماشین Bob می نماید. بدین منظور بستهای با تنظیم SYN=1 و ISN_A=x ارسال می شود؛ x عددی تصادفی است و بر اساس آن ترتیب صحیح بسته ها حفظ می شود . اگر سمت ما قبل پروسهای به شماره پورت مورد نظر گوش بدهد در پاسخ به این تقاضا بسته ای با مشخصات ACK=1 و SYN=1 و ISN_a=y ارسال می کند . ISN_a نیز عددی است و ترتیب صحیح بسته های سمت مقابل را تضمین می کند. پس از ارسال نهائی یک بسته با مشخصات ACK=1 ، SYN=1 $Ack.N.ISN_a$ پروسه های طرفین، قادر به مبادلهٔ داده خواهند بود. بسته هائی که از Alice به Bob ارسال می شوند با شمارهٔ ترتیب ISN_A شروع می شوند و بالعکس بسته های Bob به Alice با شمارهٔ

ترتیب ISN_a آغاز می شوند. در خلاف مبادلهٔ داده ها، هر گاه بسته ای در مسیر گم یا خراب شود، گیرنده با اعلام شمارهٔ آن در فیلد .Ack.No تقاضای ارسال مجدد آنرا می نماید.

پویش به روش TCP ACK Scan

همانند سه مکانیزم قبلی، در مکانیزم TCP ACK Scan برای پویش یک پورت از ارسال غیرمتعارف بسته ACK به سمت یک هدف استفاده می شود. یعنی بدون مقدمه یک بستهٔ SYN-ACK، به سوی یک پورت در ماشین هدف ارسال می شود. (بطور معمول و متعارف این بسته در پاسخ به بستهٔ SYN فرستاده میشود.) حل وقتی این بسته به یک پورت باز ارسال می شود، چون ماشین منتظر دریافت چنین بسته ای نبوده است، آنرا حذف می کند لذا عدم بازگشت پاسخ نشانگر آنست که احتمالاً آن پورت باز است ولیکن اگر پورت مربطه بسته باشد در پاسخ بستهٔ RESET برمی گردد.

پویش بروش TCP ACK Scan یک امتیاز بسیار مهم نسبت به بقیهٔ روشها دارد و آنهم امکان عبور چنین بسته ای از دیوار آتش یا مسیریابتهای فیلتر کننده می باشد:

بطور معمول از یک شبکهٔ داخلی که هیچگونه سرویسی را به خارج از شبکه نمی دهد بوسیله دیوار آتش حراست می شود؛ دیوار آتش اجازهٔ ورود هیچگونه بستهٔSYN را به درون شبکه نمی دهد یعنی چون قرار نیست هیچ سرویسی به بیرون از شبکه داده شود لذا ورود بسته های SYN که اولین مرحله از برقراری یک ارتباط TCP و سرویس گیری محسوب می شود، موردی ندارد و باید حذف شود. بدین نحو در مکانیزمهای

پویش Polite Scan و TCP SYN Scan بسته های SYN قادر نخواهند بود از دیوار آتش عبور کنندو بالطبع این مکانیزمهای برای آگاهی از باز یا بسته بودن پورتها عملاً ناتوان خواهند بود. هر چند دیوارهای آتش، بسته های SYN وارد شده از خارج را حذف می کنند ولی بسته های SYN-ACK از دیوار آتش عبور داده خواهد شد زیرا بسته های SYN-ACK در پاسخ به بسته ای SYN که از درون به بیرون ارسال شده، به شبکهٔ وارد می شوند تا مرحلهٔ دوم از دست تکانی سه مرحله ای کامل شود. شکل (۸-٦) این مفهوم را نشان می دهد: در این شکل دیوار آتش مانع ورود بسته SYN به درون شبکه می شود در حالی که ورود بسته SYN-ACK مجاز است.

دقت کنید که وقتی یک ماشین داخلی می خواهد به یک سایت وب دسترسی داشته باشد قاعدتاً یک شماره پورت بزرگتر از ۱۰۲٤ انتخاب کرده و اقدام به ارسال یک بستهٔSYN به سمت پورت ۸۰ از سرویس دهنده می کند و منتظر بستهٔSYN-ACK می شود. بنابراین ورد بسته SYN-ACK از پورت ۸۰ به سمت پورتی با شمارهٔ بالاتر از ۱۰۲٤ کاملاً طبیعی و مجاز است. با استفاده از همین مفهوم نفوذ گر قادر است پورتهای باز یک ماشین را پویش نماید. در شکل (۹-٦) چگونگی پویش پورتهای باز از میان دیوار آتش به تصویر کشیده شده است.

- اگر در پاسخ به بستهٔ SYN-ACK پاسخ RESET برگردد نشان می دهد که آن پورت باز است . زیرا پورتهای که به آن شمارهٔ پورت گوش می داده ، انتظار دریافت

بستهٔ SYN-ACK نداشته است؛ لذا تعجب خود را از دریافت چنین بسته ای با ارسال بستهٔ RESET اعلام می کند!!

- اگر در پاسخ به بسته های SYN-ACK پاسخی برنگردد نمی توان از باز بودن و بسته بودن پورت اطمینان حاصل کرد بهمین دلیل ابزارهای پویش پورت (همانند Nmap) اگر در یک مدت زمان معین، پاسخی دریافت نکند آن پورت را فیلتر شده (Fileed) در نظر می گیرند یعنی فرض می شود که آن بسته توسط مسیریاب یا دیوار آتش حذف شده است مگر آنکه بوسیلهٔ تکنیکهای دیگر خلاف آن ثابت شود.

پویش به روش FTP Bounce Scan

یک نفوذگر تمایل دارد که در حین پویش یک شبکه و جستجوی پورتهای باز، هویتش ناشناس بماند و آدرس IP ا کشف نشود. یکی از مکانیزمهائی که نفوذگر برای پویش مخفیانه بکار می گیرد قابلیتی است که سرویس دهنده های قدیمی FTP در اختیار کاربران می گذاشتند. بدین ترتیب که کاربران می توانستند ضمن بررسی ارتباط TCP با سرویس دهنده و ایجاد یک نشست، تقاضا بدهند تا یک فایل از آن سرویس دهنده به یک ماشین ثالث منتقل شود. یعنی در حقیقت یک کاربر دارای خط انتقال کم ظرفیت می توانست از سرویس دهنده های سریع (دارای خطوط با پهنای بالا) بخواد تا یک فایل را به ماشینی دیگر در شبکهٔ منتقل نماید تا این انتقال با سرعت بیشتری انجام شود. این

امکان اگر چه راحتی کاربران را فراهم می کند ولیکن به نفوذ گر امکان می دهد تا پویش پورتهای باز یک ماشین اراز طریق سریس دهندۀ FTP انجام بدهد!

با استفاده از این قابلیت، نفوذگر یک ارتباط TCP با سرویس دهندۀFTP برقرار می کند و از آن می خواهد تا با یک شماره پورت مشخص در ماشین هدف ارتباط برقرار نماید. (به منظور انتقال فایل) اگر پورت مورد نظر بر روی ماشینی هدف باز نباشد، سرویس دهندۀFTP به نفوذگر گزارش می دهد که پورت مربوطه بسته است و قادر به برقراری ارتباط نیست؛ اما اگر پورت مربوطه باز باشد، سرویس دهنده به نفوذ گر گزارش می دهد که پورت مربوطه بسته است و قادر به برقرری ارتباط نیست؛ اما اگر پورت مربوطه باز باشد، سرویس دهنده به نفوذ گر گرارش می دهد که پورت مورد نظر او باز است ولی قادر به مبادلۀ داده طبق پروتکل FTP نیست. این همان چیز است که نفوذگر می خوهد بداند! حال نفوذگر پورتهای بعدی را امتحان می کند. وقتی که ماشین هدف، مشخصات ماشین پویشگر را ذخیره و درج می نماید در حقیقت مشخصات سرویس دهندۀ FTP را درج کرده، در حالیکه یکواسطۀو کاملاً بی گناه است بنابراین هویت نفوذگر مخفی خواهد ماند. تنها از طریق بررسی قابل مراجعات به سرویس دهندهFTPمی توان نفوذگر را شناسائی کرد. شکل (۱۰-٦) مکانیزیم این نوع پویش را نشان می دهد.

به این قابلیت سرویس دهندۀFTP File- Forwarding گفته می شود که بدلیل همین نوع مشکلات، امروزه در اکثر سرویس دهنده های FTP از آن حمایت نمی شود. اگر در

شبکهٔ خود سرویس دهندهٔFTP نصب کرده اید مطمئن می شوید که چنین قابلیتی را عرضه نمی کند چرا که می تواند قربانی توطئه نفوذگران قرار بگیرد. اگر می خواهید از وجود یا عدم وجود این قابلیت در سرویس دهندهٔ FTP خود مطمئن شوید. از نرم افزاری که توسط گروه CERT در دانشگاه کارنگی ملون نوشته شده است، استفاده کنید:

http://www.cert.org/advisories/CA-1997-27.html

تنظیم زیرکانهٔ شماره پورت مبداء (Source Port) برای پویش موفق

برای آنکه شانس عبور بسته هائی که توسط نرم افزار پویشگر تولید می شود از مسیریاب و دیوار آتش یک شبکه افزایش یابد، نفوذگر سعی می کند شمارهٔ پورت مبداء بسته های TCP و UDP ارسالی خود را به نحو زیرکانه و دقیقی تنظیم نماید.

به خاطر دارید که نرم افزار پویشگر پورت ، یک سری از بسته های متوالی (TCP یا UDP) به آدرس پورتهای مختلف از ماشین هدف ارسال می نماید تا تعیین کند کدام پورت باز و کدام پورت بسته و غیرفعال است . لذا فیلد Destination Port از بستهٔ ارسالی تعیین کنندهٔ شمارهٔ پورتی است که قرار است تحت بررسی و آزمایش قرار بگیرد.

فیلد Source Port تعیین کرده اند. هدف نفوذگر آنست که بسته های ارسالی به سمت ماشین هدف از فیلتر عبور نمایند؛ یعنی بهمان صورتی که بسته های معمولی و مجاز از

دیوار آتش یا فیلتر عبور می کنند بسته های TCP ارسالی به سمت ماشین هـدف نیـز از فیلتر عبور نمایند.

نفوذ گر به دقت شمارهٔ پورتهائی را کـه اگـر در فیلـد Source Port از یـک بسـتهٔ TCP تنظیم شود قادر به عبور از دیوار آتش خواهد بود، می شناسد. بـه عنـوان مثـال معمـولی ترین شمارهٔ پورت، TCP است بسته ای که با این شمارهٔ پورت به سمت ماشـین هـدف ارسال شود شانس زیادی برای عبور از فیلتر ها و دیوارهای آتش دارد چرا که به نظر می رسد این بسته از طرف یک سرویس دهندهٔ وب ارسال شده و ناشی از تقاضای قبلـی آن ماشین بوده است؛ در اینجا فیلتر به ناچار بسته را عبور خواهد داد: یعنی نفوذگر برای بـه اشتباه انداختین فیلتر، بسته TCP را با مشخصات زیر ارسال می نماید.

(جعلی) Source Port=80

شمارهٔ پورت مورد آزمایش = Destination Port

مشخصهٔ یک بستهٔ SYN-ACK ACK Bit = 1

یکی دیگر از شماره پورتهای مشهور که شانس عبور زیـادی از فیلتـر دارد، پـورت TCP است که به طور معمول از طرف سرویس دهندهٔ نامهٔ الکترونیکی یـا SMTP تولیـد مـی شود.

ترافیک پورت TCP شمارهٔ ۲۰ نیز شانس عبور از فیلتر را دارد؛ زیرا که این پورت متعلق به سرویس دهندهٔ FTP است. به گونه ای که در شکل (۱۲-٦) مشخص است در شرایط عادی و متعارف، برقراری یک نشست FTP منوط به برقـراری دو ارتبـاط مجـزای TCP

است که هر یک کانال نامیده می شود. ارتباط TCP با پورت شمارهٔ ۲۱ از سرویس دهنده جهت مبادلهٔ فرامین FTP (کانال فرمان) و ارتباط TCP با پورت شمارهٔ ۲۰ از سرویس دهنده FTP جهت مبادلهٔ فایل. (کانال داده)

کانال فرمان از سمت برنامهٔ مشتری به سرویس دهنده FTP باز می شود و جهت عملیاتی نظیر ورود (Login)، تقاضای فایل، فهرست گیری و .... کاربرد دارد . پس از باز شدن این کانال و به منظور مبادلهٔ داده های فایل، سرویس دهندهٔFTP یک ارتباط مستقل TCP بین شمارهٔ پورت ۲۰ خود و شمارهٔ پورت پیشنهاد شده از مشتری برقرار می نماید.با این توضیح هر سرویس دهندهٔ FTP خارجی باید اجازه داشته باشد تایک ارتباط TCP از بیرون به درون یک شبکه برقرار نماید و بالطبع دیوار آتش یا فیلتر باید مجوز عبور برای بسته های TCP با مشخصهٔSource Port=20 ارسال می کند و لاجرم دیوارهای آتش معمولی و فیلتر ها آنها را عبور می دهند. (شکل ۱۳-٦)

برای ارسال بسته های UDP نیز بهترین شماره ای که می توان به عنوان Source Port استفاده کرد، پورت ۵۳ (متعلق به سرویس دهندهٔDNS) است زیرا ورود بسته های DNS به درون هر شبکه تقریباً الزامی و اجباری است.

نرم افزار Namp این امکان را فراهم آورده تا بتوان هر شمارهٔ پورت دلخواه برای بسته های TCP یا UDP بعنوان Source Port انتخاب کرد.

رد گم کردن

هیچ نفوذگری علاقه ندارد که هویت او فاش شود و سریعاً آدرس IP او بدست آید. در این صورت اگر نتوان به روشهای حقوقی و کیفری او را مجازات کرد، حداقل پاسخی که می گیرد حملهٔ متقابل است . بهمین دلیل نفوذگر تمایل دارد که در تمام مراحل حمله شامل پویش و جستجوی پورتهای باز، آدرس IP حقیقی خود استفاده کند تا قادر باشد پاسخ بسته های ارسالی به ماشین هدف را دریافت کند.

نفوذ گر برای رد گم کردن، تعدادی آدس IP جعلی انتخاب کرده و هنگامی که یک بسته را با آدرس حقیقی خودش ارسال می نماید چند بسته بی مصرف با آدرس IP جعلی نیز بهمراه آن می فرستد.

بعنوان مثال از هر پنج بسته فرستاده شده به یک شمارهٔ پورت، چهار تای آن آدرس IP جعلی دارد؛ لذا بازرسان امنیتی بخواهند با بررسی فایلهای ثبت شده (Loge Files) به هویت نفوذگر پی ببرند با مجموعه ای از آدرسهای IP مواجه می شوند که مشخص نیست کدامیک از آنها نفوذ گر است و کدام بی گناه!!

در حقیقت وقتی پنج بستهٔ TCP جهت پویش و جستجوی پورت باز به سمت یک ماشین هدایت می شود، پنج بستهٔ پاسخ برای پنج ماشین متفاوت باز می گردد که ماشین نفوذگر در بین آنها است ولی از دیدگاه ماشین قربانی ، آدرس دقیق نفوذگر مشخص نیست.

اگر به فرض پهنای باند مناسبی در اختیار نفوذگر باشد و به ازای هر بستهٔTCP که به سمت یک شمارهٔ پورت از ماشین هدف ارسال می شود، سستی بستهٔ TCP با آدرس IP جعلی نیز ارسال شود ردگیری و تعقیب نفوذگر بسیار مشکل خواهد بود زیرا آنچه از فایل ثبت (log file) استنباط می شود آنست که از بین ۳۱ آدرس IP، یکی از آنها نفوذگر بوده است. چطور می توان ۳۱ نفر را مجازات کرد؟ پس نفوذگر عملاً هویت خود را مخفی نموده است.

Firewalk بر علیه FirewalL

بدیهی است که اگر در شبکه هیچ گونه فیلتر یا دیوار آتش نداشته باشد، آن شبکه براحتی میدان تاخ تو تاز ابزارهای پویشگر پورت قرار خواهد گرفت و هر فرد بی تجربه نیز قادر خبوهد بود ماشینهای شبکه شما را برانداز نماید!

پویش و جسجوی پورتهای باز، تعیین پروسه های فعال روی ماشین هدف و تعیین سیستم عامل ماشینهای هدف، برای شروع یک حمله کافی نیست. نفوذگر قبل از هر اقدامی، تلاش می کند جزئیات بیشتری را از شبکهٔ هدف بدست بیاورد.

فراموش نکنید که اگر در شبکه یک مسیریاب با قابلیت فیلترینگ بسته یا یک دیوار آتش نصب کرده اید در حقیقت کار را برای نفوذگر سخت کرده اید نه غیرممکن! زیرا دیوار آتش یا فیلتر تنها یک ابزار که یکسری قواعد را به اجرا می گذارد در حالی که نفوذگر از تمام هوش و تجربهٔ خود برای نفوذ به شبکهٔ بهره می گیرد.

یکی از ابزارهای بسیار بدیع و زیرکانه که می تواند در خدمت نفوذگر قرار گیرد نرم افزار Firewalk است که بر ضد FirewalL (دیوار آتش) بکار می رود. این نرم افزار که توسط دو نفر بنامهای دیوید گلداسمیت و میخائیل شفمن نوشته شده در آدرس زیر در دسترس عموم قرار گرفته است است:

http://www.packetfactory.net/projects/Firewalk-final.html

Firewalk سعی می کند تا متوجه شود چه شماره پورتهائی از طریق دیوار آتش باز مانده است! در حقیقت Nmap قادر نخواهد بود تعیین کند کدام پورت بر روی ماشین نهائی باز است و کدام پورت از طریق دیوار آتش باز مانده است. فراموش نکنید که اگر پویگری مثل Nmap پورتی را ر روی یک ماشین بسته اعلام کرد، نمی توانید یقین داشته باشید که آن پورت واقعاً بسته است؛ شاید دیوار آتش مانع از کشف باز بودن آن شده باشد. هر گاه پورتی واقعاً بسته باشد می گوییم «پورت بر روی ماشین نهایی بسته است» ولی اگر دیوار آتش مانع از برقراری ارتباط با یک پورت باز در درون شبکه بشود. «پورت ازطریق دیوار آتش بسته است». Firewalk بدنبال کشف همین موضوع است!

قطعاً به Firewalk نیاز دارید!

در یک جملهٔ ساده ، Firewalk سعی می کند قواعد فیلترینگ دیوار آتش شبکه را استخراج نماید. به عنوان مسئول شبکه قبل از آنکه نفوذگر از آن بر علیه شما استفاده کند، آنرا بیازمائید، Firewalk نیز همانند بقیهٔ ابزارهای قدرتمند بر روی لینوکس اجرا می شود! این ابزار ، بهمراه مستندات کامل آن بر روی CD ضمیمهٔ کتاب، می باشد.

حال باید بررسی کنیم نرم افراز Firewalk چگونه عمل می کند و بر چه اصولی استوار است:

Firewalk شبیه به ابزار traceroute که در ابتدای همین فصل توضیح داده شد، از فیلد TTL(Time To Live) در بستهٔIP بهره می گیرد و در پی آنست که بفهمد دیوار آتش در یک شبکه کدام بسته های TCP یا UDP را فیلتر می کند و کدام را فیلتر نمی کند.

برای شروع، Firewalk دو آدرس IP از کاربر طلب می کند:

- آدرس IP متعلق به دیوار آتش یا مسیریاب فیلتر کننده
- آدس ماشین هدف که در پشت دیوار آتش پناه گرفته است.

(این دو آدرس در مراحل قبلی پویش با استفاده از ابزاری همانند Nmap بدست آمده اند.)

بر اساس این دو آدرس ، Firewalk در دو مرحله سعی می کند قواعد دیوار آتش را استخراج نماید و شمارهٔ پورتهایی را که ترافیک آنها می تواند به درون شبکه وارد شود، کشف کند:

الف) کنکاش در شبکه

در مرحلهٔ اول که در شکل (۱٤-٦) به تصویر کشیده شده است، Firewalk یک سلسله از بسته های IP را با تنظیم فیلد TTL به ۱ و ۲ و ۳ و .... به سمت هدف می فرستد. این عمل باعث خواهد شد مسیریابهای بین ماشین نفوذگر و دیوار آتش مشخص شود زیرا به محض حذف بسته با TTL=1 در اولین مسیریاب پاسخ ICMP Time

Exceeded برخواهد گشت و بهمین ترتیب دومین مسیریاب مسیریابهای بعدی تا رسیدن به دیوار آتش کشف می شوند. این عمل تا اینجا دقیقاً مشابه با عملکرد برنامهٔ Traceroue است؛ ولی Firewalk نیازی به دانستن آدرس مسیریابیها بین راه ندارد بلکه فقط نیاز دارد « تعداد» آنها را تا رسیدن به دیوار آتش بداند. قوتی این تعداد (N) مشخص شد، Firewalk وارد مرحلهٔ بعد می شود.

ب) مرحلهٔ پویش:

در این مرحله بگونه ایی که در شکل (۱۵-٦) نشان داده شده، Firewalk سلسله ای از بسته های IP با یلد TTL دقیقاً معادل با N+1 بهسمت ماشین هدف ارسال می نماید. (مسیریاب N+1 یعنی اولین مسیریاب یا ماشین دقیقاً بعد از دیوار آتش) بسته IP محتوی یک بسته TCP با شمارهٔ پورت مورد جستجو است؛ آدرس ماشین هدف در پشت دیوار آتش است. یقیناً این بسته به دیوار آتش خواهد رسید؛ اگر بسته از دیوار آتش عبور کند و به ماشین بعدی برسد بلافاصله پیغام ICMP Time Exceeded بر خواهد گشت لذا Firewalk اعلام می کند که شماره پورت مورد جستجو از میان دیوار آتش باز و در دسترس می باشد زیرا دیوار آتش مجوز عبور بسته به درون شبکه را صادر کرده است. این پیغام بدین معنا نست که پورت مربوطه روی ماشین هدف باز است، بلکه بدین معناست که بسته از دیوار آتش عبور کرده و دقیقاً در ماشین بعدی عمرش تمام شده و حذف گردیده است. اگر بسته نتواند از دیوار آتش عبور کند، هیچ پاسخی باز نخواهد گشت.(یا پیغام ICMP Port Unrrtachable باز می گردد.) در این حالت مطمئن

خواهیم شد که دیوار آتش بسته های TCP (یا UDP) با شماره پورت مورد نظر را حذف خواهد کرد! دقت کنید که عدم برگشت پاسخ یا ارسال پیغام ICMP Port Unreachable بهیچوجه مربوط به ماشین هدف نیست چرا که با تنظیم TTL=N+1 چنین بسته ای قاعدتاً به ماشین هدف نخواهد رسید، لذا مطمئن می شویم که این اتفاق درون دیوار آتش رخ داده است!!

Firewalk مکانیزم فوق را برای شماره پورتهای مختلف تکرار کرده و ضمن تحلیل نتیجه، پورتهائی را که دیوار آتش برای آنها مجوز عبور صادر خواهد کرد، اعلام می نماید.

باید به یک نکتهٔ مهم در مورد Firewalk اشاره کنیم: مکانیزمی که این نرم افزار اتخاذ کرده است فقط برای دیوارهای آتش یا مسیریابهای فیلتر کننده موثر خواهد بود و در مورد پراکسی Proxy نتیجهٔ موثری نخواهد داشت. دقت داشته باشید که پراکسی (Proxy) بدین نحو عمل می کند که با دریافت یک تقاضای برقراری ارتباط TCP از یک ماشین خارجی:

- یک ارتباط TCP بین پراکسی و ماشین مبداء برقرار می شود.
- یک ارتباط TCP بین پراکسی و ماشین مقصد برقرار می شود.

داده ها ابتدا از مبداء تحویل گرفته می شود و سپس برای مقصد ارسال می گردد. (داده های ارسالی از مبداء ابتدا بطور کامل تحویل لایهٔ کاربرد (Application) در پراکسی شده و پس از بررسی های لازم از طریق ارتباط TCP بین پراکسی و هدف، تحویل داده

خواهد شد.) بدین ترتیب هیچ ارتباط مستقیم و انتها به انتها[1] بین ماشین مبداء و ماشین هدف ایجاد نخواهد شد.

از آنجایی که پراکسی بسته ها را عبور نمی دهد بلکه آنها را بطور کامل دریافت کرده و پس از بررسی های لازم مجدداً از طریق یک ارتباط مستقل TCP به سمت هدف ارسال می کند، لذا مکانیزم, Firewalk برای کشف پورتهای مجاز و باز کاملاً بی تاثیر است زیرا بسته ای ارسالی با TTL=N+1 بطور کامل در پراکسی جذب شده، از آن عبور نخواهند کرد و قطعاً پیغامی شبه ICMP Time Exceeded باز نخواهدگشت.

ابزار Firewalk مجموعهٔ شماره پورتهائی که از طریق دیوار آتش باز هستند را به نفوذگر اعلام خواهد کرد لذا او وقت خودرا برای نوذ از طریق پورتهایی که فیلتر می شوند، تلف نخواهد کرد. (بار دیگر تاکید می کنیم که وقتی شماره پورتی از طریق دیوار آتش یا فیلتر بسته (بلوکه) می شود ممکن است روی ماشین هدف باز باشد ینی در درون شبکهٔ داخلی می توان بدین پورت متصل شد و مبادلهٔ داده کرد ولی از بیرون از شبکه اتصال با این پورت امکان پذیر نیست و بسته های ورودی به دیوار آتش با شمارهٔ پورت مورد نظر حذف می شوند. برعکس آن امکان پذیر است یعنی شماره پورتی از طریق دیوار آتش باز است و بلوکه نشده در حالی که پورت مربوطه روی ماشین هدف بسته است یعنی اگر چه دیوار آتش اجازهٔ عبور بسته هائی با شمارهٔ پورت مورد نظر را می دهد ولیکن پروسهٔ فعالی روی ماشین هدف وجود ندارد. (تا به پورت موردنظر گوش بدهد.)

---

[1] End to End

نفوذگر در نخستین مراحل تهیهٔ طرح حمله، بوسیلهٔ Firewalk شمارهٔ پورتهائی را کـه از طرق دیوار آتش باز مانده اند، استخراج می کند تا بروشهائی که در فصول بعـد توضـیح می دهیم پروسه ای را روی ماشین هدف فعال کند تا به شمارهٔ پورتهای باز گوش بدهـد. (روشی مثل اسبهای تروا) یعنی تلاشهای بعدی نفوذگر معطـوف بـه ایجـاد سرویسـهائی روی ماشین هدف خواهد بود که به شمارهٔ پورتهای مجاز و بولکه نشـده گـوش بدهنـد. مثلاً فرض کنید نفوذگرتوسط نرم افزار Firewalk متوجه شده که پـورت ۲۳ مربـوط بـه TelNet از طریـق دیـوار آتـش بـاز مـی باشـد و بلوکـه نشـده اسـت ولـی سـرویس دهندهٔTelNet روی ماشینهای هدف وجود ندارد. ممکن است او با نوشـتن یـک برنامـهٔ کوچک (یا یک قطه اسکریپت) بطور متوالی و شبانه روزی، پـورت ۲۳ را روی ماشـین هدف بررسی نماید. (مثلاً هر ۱۵ دقیقه یکبار) احتمالاً این پورت بدین دلیل بازاست کـه در برخی مواقع، مسئول شبکه برای عملیاتی شبکه رفع اشکال از شبکه، سرویس دهنـدهٔ Telnet را موقتاً فعال می کند تا بتواند از راه دور به یک ماشین وارد شود. در این حالـت برنامهٔ نوشته شده در سمت نفوذگر فعال شدن سرویس دهنده و با شده پورت را گزارش می کند و او را برای اجرای مقاصد بعدی خود آگاهی سازد. لذا حتی بـاز بـودن شـماره پورتهائی که بطور طبیعی سرویس دهندهٔ آن روی هیچیک از ماشینهای شبکه فعال نیست می توند خطری بالقوه برای آن شبکه محسوب شود!

باز بودن شمارهٔ پورتهائی که سرویس دهندهٔ آنها روی ماشینهائی نهائی فعال نیسـت ایـن امکان را به نفوذگر می دهد که در مرحلهٔ نفوذ پروسه ای را روی ماشین هدف فعال کنـد

تا به شمارهٔ پورت را عبور خواهد داد؛ چرا که مسئول شبکه فکر می کرده هیچ پروسه ای که به این شماره پورت گوش بدهد در شبکهٔ او وجود خارجی ندارد. برای مثال فرض کنید که مسئول شبکه دیوار آتش را بگونه ای تنظیم کرده که شمارهٔ پورت ۲۷۳۷٤ باز مانده باشد. در شرایط عادی هیچ پروسه ای روی ماشینهای شبکه به چنین پورتی گوش نمی دهد و لذا به زعم او خطر متوجه هیچ ماشینی در شبکه نیست. حال فرض کنید که یک برنامهٔ آلوده به اسب تروای Sub7 (که د رفصل دهم معرفی خواهد شد) توسط نامهٔ الکترونیکی (یا حتی دیسکت آلوده) برای یکی از کاربران ارسال شود؛ با اجرای آن روی ماشین کاربر، پروسهٔ Sub7 فعال شده و به پورت ۲۷۳۷٤ گوش خواهد داد . لذا یکی از ماشینهای شبکه بطور کامل در اختیار نفوذگر قرار گرفته و راه برای حملات آتی او باز شده است و هیچ کاری نیز از دیوار آتش برنمی آید!!

مقابله با Firewalk

راههای متنوعی برای مقابله با پویش و جستجوی شبکه توسط نرم افزار Firewalk وجود دارد:

- تمام پورتهای باز (مجموعاً ٦٥٥٣٥ پورت) را به استثنای آنهائی که یقیناً باید باز باشند، بدون هیچ تردیدی ببندید. با توجه به آنکه نفوذگر احتمالاً با استفاده از برنامه های خودکار یا اسکریپتها بطور متناوب و شبانه روزی ، شبکهٔ شما را جستجو می کند و مترصد نفوذ است حتی برای چند ساعت نیز پورتهای یر ضروری را باز نکنید.

- دیوارهای آتش و مسیریابهای فیلترکننده در مقابل Firewalk ضعف دارند و بر اساس روشی که Firewalk اتخاذ کرده است قواعد فیلترینگ آنها آشکار می شود، لذا توصیه آنست که اگر شبکهٔ بسیار حساس و حیاتی را مدیریت می کنید بجای استفاده از دیوار آتش از فیلترهای پراکسی استفاده کنید. پراکسیها هویت شبکهٔ داخلی را پنهان نگاه می دارند چرا که هیچ ارتباط مستقیمی بین ماشین خارجی و ماشینهای داخلی برقرار نخواهد شد بلکه تمام ارتباطات TCP خارجی با پراکسی خواهد بود لذا هیچ گونه بسته ای (چه مجاز و چه غیر مجاز) مستقیماً برای هدف ارسال نخواهد شد. (در بازار به چنین فیلترهائی Proxy Based Firewall گفته می شود.) البته باید اذعان داشت که استفاده از فیلترهای پراکسی ممکناست سرعت شبکه را پائین بیاورد بنابرای قبل از تهیهٔ این گونه فیلترها از کارآتی آن مطمئن باشید.(یعنی بررسی کنید آیا سرعت عملیات پراکسی با پهنای باند در اختیار شما متناسب است یا خیر؟)
- ICMP اگر چه برای اهداف خیرخواهانه پیاده سازی شده است ولی استفاده هائی که از آن برای اهداف شوم نفوذگران می شود گاهی زیادتر از منافع آن است لذا برای مقابله با Firewalk سعی کنید روی تمام ماشین های حساس شبکهٔ داخلی ، ICMP را غیرفعال کنید، یا آنکه فیلتر خود را بگونه ای تنظیم کنید که تمام بسته های ICMP (که شمارهٔ پروتکل آن در سرآیند بستهٔ IP دقیقاً مشخص است) فیلتر و حذف شوند. در چنین حالتی برای هیچ کدام از بسته هائی که از دیوار آتش عبور می کنند

یا حذف می شوند، پیغام برنخواهد گشت لذا Firewalk اشتباهاً پورتهای باز را هم بسته اعلام خواهد کرد که این به نفع شماست زیرا نفوذگر را مایوس خواهد کرد!

سیستمهای کشف مزاحمت (IDS)

سیستم کشف مزاحمت که به اختصار IDS نامیده می شود، برنامه ایست که با تحلیل ترافیک تجاری شبکه یا تحلیل تقاضاها سعی در شناسائی فعالیتهای نفوذ گر می نماید و در صورتی که تشخیص دارد ترافیک ورودی به یک شبکه یا ماشین از طرف کاربران مجاز و عادی نیست بلکه از عالیتهای یک نفوذگر ناشی می شود به نحو مناسب مسئول شبکه را در جریان می گذارد یا یک واکنش خاص نشان می دهد. در حقیقت IDS نقش آژیر دزدگیر شبکه را ایفا می نماید.

در این بخش پس از بررسی علمکرد IDS در سطوح مختلف، روشهای فرار نفوذگر از آنرا نیز بررسی خواهیم کرد. سیستم IDS در دو سطح «لایهٔ شبکه» و « لایهٔ کاربرد» عمل می کند و مکانیزم هر یک یا دیگری متفاوت است.

عملکرد سیستم IDS مبتنی بر لایهٔ شبکه

در این نوع سیستم کشف مزاحمت، IDS تمام بسته های IP وارده به شبکهٔ محلی را دریافت، جمع آوری و پردازش می کند و پس از تحلیل بسته ها ، بسته های معمولی و بسته های مزاحم (متعلق به نفوذگر) را تشخیص می دهد. IDS باید انبوهی از بستهای IP (و محتویات آنها شامل بستهای TCP و UDP) را مرتب کرده و بروز واقعی یک حمله را تشخیص بدهد.

بطور معمول سیتمهای IDS یک بانک اطلاعاتی از الگوی حملات مختلف در اختیار دانـد. (بـه ایـن بانـک اطلاعـاتی، بانـک ویژگیهـا و امضـای حملـه Attack Signatures& Features گفتـه مـی شـود) در حقیقـت اکثـر سیسـتمهای IDS تحلیلهای خود را بر تطابق الگوهای حمله با ترافیک موجود در شبکه متمرکز کرده اند و هر گاه الگوی ترافیک جاری در شبکه با ویژگی یکی از حملات منطبق باشد یک حملـه گزارش خواهد شد. لذا نفوذگر برای فرار از IDS، سعی می کنـد بـه روشـهای مختلـف مراحل حمله را بگونه ای سازماندهی کند که IDS آنرا ترافیک معمولی و طبیعی بپندارد. (در این مورد صحبت خواهیم کرد.)

وقتی حمله ای کشف شود، سیستم IDS با ارسال e-mail، سیستم پی جر (Pager) یا به صدا درآوردن بوق آژیر آنرا به اطلاع مسئول شبکه می رساند و در عین حال بـه تعقیـب حمله ادامه می دهد. شکل (۱۹-٦) یک سیستم IDS معمولی (در سطح شـبکه) را نشـان می دهد.

در این شکل سیستم IDS در حین نظارت بر ترافیک شبکه، متوجه تلاش برای ارتباط بـا پورتهای ۸۰ و ۲۳ شده است. این سیستم تلاش برای برقراری ارتباط TCP با پـورت ۸۰ را عادی تلقی میکند ولی تلاشهای متوالی برای برقراری ارتباط با پـورت۲۳ (مربـوط بـه TelNet) را اصلاً طبیعی نمی داند و آنرا به عنوان علائم یک حمله گزارش می کنـد. یـا مثلاً سیستم IDS با تحلیل جریان بستهای IP متوجه می وشد که چند هـزار بسـتهٔ SYN

با فیلد Source IP یکسان و با شماره های مختلف پورت به شبکه ارسال شده است؛ این مسئله قطعاً علامت بروز یک حمله است.

حال باید دید نفوذگر به چه نحوی تلاش می کند از IDS مبتنی بر لایهٔ شبکه فرار کند؟ نفوذگر از مکانیزمهای زیر برای فرار از IDS(IDS Evasion) بهره می گیرد: ترافیک ارسالی به شبکهٔ هدف بگونه ای تنظیم که باالگوی هیچ حمله ای تطابق نداشته باشد. در چنین حالتی ممکن است نفوذگر از برنامه نویسی استفاده کند چرا که ابزارهای موجود الگوی حمله شناخته شده ای دارند.

بسته های ارسالی به یک شبکه بگونه ای سازماندهی می شوند که عملکرد دقیق آن فقط در ماشین برای روشن شدن نکات ابهام در روشهای فوق به چند مثال عملی خواهیم پرداخت:

بگونه ای که در فصل مفاهیم TCP/IP تشریح شد یک بستهٔ بزرگ می توانند به یک سلسلهٔ از قعطات کوچکتر (Fragment) شکسته شود. هر بستهٔ شکسته شده در ماشین مقصد بازسازی خواهند شد. وقتی سیستم IDS با بستهای قطعه قطعه شدهٔ IP را در قطعات بسیار کوچک (مثلاً ۸ بایتی) شکسته و آنها را ارسال کند.در ضمن برای فلج کردن IDS بسته های IP بسیار زیاد و قطعه قطعه شدهٔ بی هدفی را نیز لابلای بسته های حمله ارسال می کند. IDS باید بافر بسیار زیادی در اختیار داشته باشد تا بتواند ضمن بازسازی قطعات شکسته شده، درون آنها به جستجوی الگوی حمله بپردازد.

تا تابستان سال ۲۰۰۰ تقریباً هیچ سیستم IDS وجود نداشت که قادر به بازسازی قطعات بسته های IP باشد لذا هر نفوذگری با قطعه قطعه کردن بسته های IP (محتوی بستهٔ TCP یا UDP) از سیستم IDS فرار می کرد. بعنوان مثال ابزار Snort (که یک نرم افزار Opent Source و رایگان است) بعنوان یک سیستم IDS بسیار معروف، تا سال ۲۰۰۰ در مقابله با بستهای قطعه قطعه شده ناتوان بود!

در ضمن نفوذ گر می تواند قطعه قطعه کردن بستهٔ IP را به روش های نامتعارف انجام بدهد بگونه ای که سیستم IDSنتواند بدرستی آنرا بازسازی کند. مکانیزیم این نوع حمله به شرح زیر است:

حمله به IDS بر اساس قطعات کوچک و قعطات هم پوشان IP

روش حمله از طریق بسته های قطعه قطعه شدهٔ کوچک بر علیه IDS در شکل (۲۰-۶) به تصویر کشیده شده است.

فرض کنید یک بستهٔ IP محتوی یک بسته TCP (در فیلد ParLoad) باشد؛ چون بخش Payload از هر بستهٔIP می تواند قطعه قطعه شود. لذا بطور عمدی قطعهٔ اول به قدری کوچک در نظر گرفته می شود که فقط دو فایل اول از بسته TCP را شامل شود و بنابراین دو بایبت دوم از بسته TCP که شمارهٔ پورت مقصد (Destination Port) را در برمی گیرد در بستهٔ دوم ارسال می شود. معمولاً سیستمهای IDS برای تشخیص حمله به سرآیند بستهٔTCP احتیاج دارند تا مثلاً تلاش برای برقراری ارتباط با پورت ۲۳ مربوط به TelNet را کشف نمایند. چون بستهٔ اول، سرآیند کامل بستهٔTCP و شمارهٔ پورت

مقصد را ندارد معمولاًIDS آنرا معمولی در نظر گرفتـه و از آن مـی گـذرد؛ بدینصـورت نفوذگر IDS را دور می زند.

نوع دیگر حمله به IDS، حمله بر اساس قطعات همپوشان (Fragment Overlap) است که با دستاری و تغییرات عمدی در فیلـد Fragment Offset (از بسـتهٔIP) انجـام مـی شود. بگونه ای که در مبحث IP اشاره شده ،این فیلد برای بازسازی قعطات بستهٔ تنظـیم می شود و در حقیقت این شمارهٔ محل قرار گرفتن قطعـهٔ جـاری را در دیتـاگرام اصـلی، مشخص می کند . به شکل (۲۱-٦) دقت کنید. قطعات همپوشان با مکانیزیم زیر تنظیم و ارسال می شوند:

اولین قطعهٔ بستهٔ IP که شامل سرآیند بسته TCP است دارای شمارهٔ پورت مجـاز اسـت. (مثل دومین قطعه بگونه ای تنظیم می شود که پس از بازسازی، بر روی بخشی از قطعـه های قبل نوشته شده و مقادیر قبلی را بازنویسی کند، لذا شماره پـورت واقعـی در قطعـهٔ دوم مشخص می شود شـمارهٔ پـورتی کـه در قعطـهٔ اول درج شـده پـوچ اسـت و بعـداً بازنویسی خواهد شد.

چون احتمالاً فقط قطعه اول از هر بستهٔIP توسط IDS بررسی می شود لذا قعطهٔ دوم کـه قطعــهٔ اول را بازنویســی مــی کنــد توســط IDS تشــخیص داده نخواهــد شــد! قطعات همپوشان پس از بازسازی در ماشین هدف، بستهٔ TCP اصلی را با شمارهٔ پورت واقعی تشکیل می دهد!

مکانیزمهای Whisker برای فریب دادن IDS

Whisker ده روش متنوع و قدرتمند برای گول زدن IDS بکار می گیرد که این روشها در زیر معرفی شده اند: (در تمام این روشها منظور از تقاضا، ارسال یک فرمان HTTP به سرویس دهندهٔ و برای فعل و انفعال با برنامهٔ CGI تلقی شده است.)

URL Encoding: قسمت آدرس در URL ارسالی با کدهای معمولی ASCII ارسال نمی شود بلکه ابتدا هر کاراکتر با معادل پونی کد آن (یعنی با قالب %xx تعریف شده در MIME) جایگزین و سپس ارسال می شود . برخی از سیستمهای IDS قادر نیستند چنین قالبی را تشخیص بدهند و لذا یک تقاضای خطرناک کشف نخواهد شد. مثال:

Get / %63%46%69%2d%62%69%6e / broken.cgi HTTP/1.0

نکتهٔ: طبق استاندارد MIME ، برای معادلسازی کاراکترها در رشتهٔ URL ابتدا کاراکتر % و سپس کد هگزادسیمال عدد قرار می گیرد.

Directory Insertion: URL ارسالی شامل کاراکترهای /./ است که در برخی از سرویس دهنده های وب به این شکل تعبیر و تفسیر می شود: لطفاً به شاخهٔ جاری تغییر مسیر بدهید!

تغییر مسیر به شاخهٔ جاری هیچ خاصیت یا ضرری ندارد بلکه فقط شکل URL را بگونه ای تغییر می دهد تا به الگوی حمله شباهت نداشته باشد و IDS آنرا مجاز بداند.

مثال:

GET /./ cgi-bin/./ broken.cgi Http/1.0

Premature URL Ending: در URL ارسالی اطلاعاتی در خصوص اسکریپت مورد نظر قرار داده می شود. در عوض این اطلاعات در بخش سرآیند HTTP جاسازی می شوند. به مثال زیر دقت کنید.

Get/HTTP/1.0\r\nHEADER:../../cgi-bin/bin/broken.cgi/HTTP/1.0

آنهایی که با پروتکل HTTP آشنا هستند صحبت URL فوق و اعتبار تقاضای GET را تایید می کنند.

Long URL : قسمت آدرس در URL ارسالی شامل نام بسیار طولانی یک شاخه است که وجود ندارد. در انتهای این نام کاراکترهای/../ قرار می گیرد؛ بدین ترتیب در سرویس دهندهٔ وب از نام شاخه چشم پوشی می شود. برخی از سیستمهای IDS فقط بخش اول از آدرس URL را بررسی می کنند و لذا یک تقاضای خطرناک کشف نخواهد شد. مثال:

GET /thisisabunchofjunktomakethURL longer/../cgir-bin/broken.cgi

HTTP/1.0

Fake Parameter: URL ارسالی شامل پارامترهایی است که هیچ خاصیت یا ضرری ندار؛ فقط شکل URL را بگونه ای تغییر می دهد تا به الگوی حمله شبیه نباشد و IDS آن را مجاز بداند. مثال:

GET /index.html?param=/../cgi-bin/broken.cgi HTTP/1.0

TAB Separation: بخشهای مختلف URL ارسالی بجای آنکه با کاراکتر «فاصله خالی» (Space) جدا شده باشند با کاراکتر <Tab> جدا می شوند. در این حالت شکل URL

بگونه ای تغییر می کند تا به الگوی حمله شباهت نداشته باشد و IDS آنرا مجـاز بدانـد. (برخی از سیستمهای IDS بدین نحوه گمراه شده و به آن تقاضا اجازهٔ اجرا مـی دهنـد و برخی دیگر آنرا حذف می کنند.) مثال:

GET<Tab> /cgi-bin/broken.cgi<Tab> HTTP/1.0

Case Sensitivity: برخی از سیستمهای IDS، انتظار URL با حروف کوچک انگلیسـی دارند ولیکن در تعدادی از سرویس دهنده های وب (مثل IIS در ویندوز) ارسال URL با حروف بزرگ و کوچک فرقی نمی کند و قابل اجراسـت. بـدین ترتیـب سیسـتم IDS فریب می خورد و تقاضای ارسالی اجرا می شود. مثلا:

GET /CGI-BIN/broken.cgi HTTp1.0

Windows Delimiter: در سیستم عامل ویندور استفاده از علامت \ بجای / (جداکننـدهٔ شاخه) مجاز شمرده می شـود در حالیکـه برخـی از سیسـتمهای IDS بـه آن حساسـیت ندارند لذا در مورد شکل URL گمراه می شوند. مثال:

GET /Cgi-bin\broken.cgi HTTP.1.0

Session Splicing: برخلاف نه روش قبلی، مکانیزم Session Splicing در سطح لایـهٔ TCP پیاده سازی شده است. تقاضا ابتدا تکه تکه شده و در بسته ای جداگانهٔ یک تا سـه کاراکتری ارسال می شود. سیار از سیستمهای IDS، توانـایی بازسـازی قطعـات URL را ندارند و لذا اجباراً برای این بسته ها مجوز اجرا صادر می کنند. مثال:

بسته های TCP هر کدام محتوی فقط بخشی از تقاضای HTTP هستند:

NULL Method: بسیاری از سیستمهای IDS برای تحلیل رشتهٔ URL از توابع رشته ای استفاده می کنند. حال اگر در بین رشتهٔURL، کاراکتر 00%(NULL Character) وجود داشته باشد توابع رشته ای آنرا بعنوان خاتمهٔ رشته تلقی می کنند و بدین نحو IDS گمراه می شود در حالی که اعتبار خود را از دست نخواهد داد. (اکثر سرویس دهنده های وب از کاراکتر 00% چشم پوشی می کنند.) مثال:

GET%00 /cgi-bin/broken/cgi Http/1.0

بگونه ای که دیده می شود Whiskerاز روشهای ساده و قدرتمندی برای مخفی ماندن بهره می گیرد. نتیجه ای که از روشهای ده گانه فوق حاصل می شود اینست که:

مکانیزمهای Whisker:

در Whisker برای گول زدن و فرار از IDS، سعی شده است تا رشتهٔ ارسالی در قالب متود GET (که منجر به فراخوانی اسکریپت CGI خواهد شد) بگونه ای شکل طبیعی خود را از دست بدهد و منجر به تحریک سیستم IDS نشود و در ضمن از دیدگاه پروتکل HTTP قابل تفسیر و معتبر باشد.

گام سوم: نفوذ از طریق رخنه در سیستم عامل یا برنامه های کاربردی

در مراحل قبلی، نفوذگر اطلاعات و نقشه های لازم را برای شروع یک حملهٔ هدفمند و مؤثر بدست آورده است. حال او یادداشتهای زیر را پیش رو دارد به شروع حمله می اندیشد.

- نقشهٔ تقریبی از شبکهٔ هدف حمله
- توپولوژی نسبی شبکه و پیکربندی سرویس دهنده ها
- قواعد دیوار آتش یا فیلترها
- سیستم عامل سرویس دهنده های مهم و مشخصات هر سرویس دهنده
- پورتهای بار هر ماشین
- نقاط آسیب پذیر ماشینهای هدف

زمان حمله فرا رسیده است: حمله ای که موفقی در آن مستقیماً به میران آگاهی و دانش فنی نفوذ گر بستگی دارد. چرا که گاهی برای حمله به یک سیستم، نفوذ گر مجبور به برنامه نویسی خواهد شد. استفاده ابزارهای موجود که به نامهای Hacking Software عرضه می شوند برای پیروزی در یک حمله، شانس موفقیت بسیار کمی دارند زیرا عملکرد آنها برای همه آشکار است و بالطبع روشها و ابزارهای مقابله با آنها نیز مشخص و مهیاست.

نکته:

در یک حملهٔ موفق، نفوذگر از روشهای بدیع، عملی و مصلحت گرایانـه (Pragmatic) بهره می گیرد نه از حملات کور و متکی به شانس! حمـلات کـور بـه سیسـتمها معمـولاً توسط افراد معمولی ، ماجراجو و بچه های کم سن و سال صورت مـی گیـرد و معمـولاً هدف خاصی را الا هیجان و عقده کشائی دنبال نمی کند.

به عنوان مسئول یک شبکهٔ مهم و حساس باید همانند نفوذگران حرفه ای با اصـول یـک حملهٔ موفق آشنا باشید. اگر حملهای اتفاق بیفتد، ضعف دانش فنی مسئول شبکه، جنـگ را به نفع مهاجم تمام خواهدکرد و چیزی که باقی می ماند یک شـبکهٔ ویـزان یـا غـارت شده است! برای عبرت آمـوزی بـه آدرس http://www.attrition.org/ مراجعـه و پیشینهٔ حملات را مطالعه کنید.

نفوذ گر به غیر از اطلاعات ، مستندات و نقشـه هـای شـبه هـدف، کـدهای لازم جهـت شروع حمله را برنامه نویسی کرده و ابزارهای مورد نیاز آمادهٔ اجرا هستند. سؤال اصـلی برای صدور فرمن حمله آنست که : « اگر نقاط آسیب پـذیر متعـدد باشـد از کجـا بایـد شروع کرد؟» آیا باید از نقاط آسیب پذیر سیستم عامل بهره گرفت یا از نقاط آسیب پذیر برنامه های کاربرد؟

پاسخ این سوال مستقیماً به هدف حمله بستگی خواهد داشـت: مـثلاً اگـر هـدف حملـه جلوگیری از سرویس دهی یک ماشین یا شبکه باشد حمله به سیستم عامـل کارآمـدتر و بنیادی تر است. اخلال در عملکرد یک سیستم عامل یا سرویس دهنده و نهایتاً فروپاشـی

آن (Crash) هدف اصلی حملات نوع DOS است که در فصل نهم تشریح خواهد شد در حالی که اگر هدف حمله سؤاستفاده از حساب کاربران و تزریق اطلاعات غلط به یک ماشین یا شبکه باشد، حمله به نقاط آسیب پذیر یک برنامهٔ کاربردی مؤثر خواهد بود. بهر حال در این فصل حالات مختلف حمله به شبکه را بررسی خواهیم کرد.

اگر چه مکانیزمهای «شناسائی و پویش شبکهٔ هدف» تا حدودی سیستماتیک و مشخص است، ولی در عوض مراحل حمله بسیار پیچیده و موموز خواهد بود. نفوذگر ممکن است در زمان مله تاکتیکهای خود را برای موفقیت در فرجام کار عوض کند و این کاملاً به هوش، دانش فنی و تجربهٔ او بستگی دارد.

انواع حملاتی که در این فصل تجزیه و تحلیل می شوند مکانیزمهای شناخته شده ای دارند و یک نفوذگر مجرب با تمام آنها آشناست. سایتهای وب شناخته شده ای در جهان موجودند که آسیب پذیری بررسی دهنده های گوناگون، سیستمهای عامل و برنامه های کاربردی و مکانیزم حمله به آنها را مفصلاً تشریح کرده اند. مشهورترین آنها عبارتند از:

- Http://packetstorm.securify.com/ Packet Security
- http://www.technotronic.com/ Technotronic Security Information
- http://www.securityfocus.com/ Security Focus Bugtraq Archive

شکل (۱-۷) صفحهٔ وب سایت Packet Srorm را نشان می دهد. هر کسی می تواند برای پیدا کردن نقاط ضعف رایج در سیستمها و مکانیزمهای نفوذ و حمله به آن مراجعه کند. در شکل (۲-۷) صحفهٔ اصلی وب سایت Security Focus را ملاحظه می کنید؛

این سایت آخرین اطلاعات در مورد کرمها و ویروسها، حملات نفوذ گران و شکافهای امنیتی کشف شده در سیستمهای مهم را ارائه می کند.

پیدا کردن نقاط آسیب پذیر

نفوذگر از کجا متوجه می شود که یک برنامهٔ یا پروسه دارای نقطه ضعف است و پشتهٔ آن استعداد سرریز شدن دارد، تا آنرا هدف بگیرد؟

آسیب پذیرترین برنامه ها ، آنهایی هستند که کد برنامهٔ آنها در اختیار نفوذگر است. او با یک نگاه کلی به توابعی که ورودی/خروجی آنها (I/O) از طریق سوکت و پورت های باز انجاممی شود و همچنین بررسی توابع بکار گرفته شده در این برنامه ها می تواند به آسیب پذیر بودن برنامه پی ببرد استفاده از توابع زیر در زبان C، برنامه را شدت آسیب پذیر می کند:

- fgets
- gets
- getws
- memcpy
- memmove
- scanf
- sprintg
- strcat

- strncpy
- ……

- اگر نفوذگر، کد برنامهٔ مورد نظرش را در اختیار نداشته باشد؟، می تواند با استفادهٔ از یکDebugger هدفش را تعقیب کند.
- اگر استفاده از Debugger نتیجه ای نداشت، پیدا کردن نقطه ضعف برنامه از طریق سعی و خطا انجام می شود . بدین معنا که نفوذ گر یک نسخه از برنامه ای را که قرار است در یک ماشین راه دور مورد حمله قرار بگیرد، در آزمایشگاه اجرا می کند؛ سپس با پورتهای باز آن یک اتصال برقرار و رشته های طولانی و متفاوت برای آن ارسال می نماید. این کار هزاران باز تکرار می شود تا بالاخره برنامه با یکی از این رشته ها مختلف شده و درهم بشکند.

در اواسط سال ۱۹۹۹ گروهeEye Secrity برنامهٔ سرویس دهندهٔMicrosoft IIS را که یک سرویس دهندهٔ وب محسوب می شود مورد بمباران داده های طولانی قرار دادند. بعد از یک ساعت برنامهٔ IIS با بجا گذاشتن پیغام «نقض حریم حافظه» و گزارشی از محتوای رجستر ها بصورت زیر، در هم شکسته شد:

EAX=00F7FCC8	EBX=00F41130
ECX=41414141	EDX=77F9485A
ESI=00F7FCC0	EDI=00F7FCC0
EIP=41414141	ESP=00F4106C

EBP=00F4108C EFL=00000246

به مقادیر هیچکدام از رجسترها بغیر از رجیستر EIP کاری نداشته باشید. بطور طبیعی محتوی رجستر EIP آدرس دستورالعملی در حافظه است که باید اجرا شود. محتوای رجیستر EIP مقدار 41414141(Hex) یعنی 'AAAA' گزارش شده است. یعنی دقیقاً با حمله ای که به برنامهٔ ساده مثال در ابتدای این بخش شده است. بر اساس نتیجه ای که گروه eEye از این حمله گرفت حملهٔ بعدی را بگونه ای ترتیب داد تا کنترل اجرا به Command Shell منتقل شود.

مقابلهٔ ابتدایی با حمله به پشته از طریق سیستم IDS

برای کشف هر گونه تلاش که منجر به فروپاشی یک سرویس دهنده یا پروسه می شود به یک سیستم IDS(«سیستم کشف مزاحمت و تشخیص حمله») در شبکه احتیاج دارید. سیستم IDS بطور معمول مکانیزم و ویژگیهای هر یک از انواع حملات را می شناسد. بعنوان مثال سیستم IDS می تواند برای آگاهی از شروع «حمله بر علیه پشته» از دو ویژگی زیر استفاده کند:

- هرگاه طول دادهای ارسالی به یک پورت باز از یک حد مشخص و مجاز طولانی تر باشد.

هرگاه درون داده های ارسالی بر روی یک پورت خاص، کدهای NOP به وفور وجود داشته سیستم IDS می تواند جزئی از یک دیوار آتش باشد یا به طور جداگانه در شبکه طراحی و نصب شود.

فرار نفوذگر از سیستم شناسائی مزاحمت

بدلیل آنکه حملات به پشته امروزه بشدت رایج شده است، بطور نسبی استفاده از IDS نیر رواج یافته ایت. بنابراین طبیع یاست که امروزه شاهد حملات پیشرفته تر از نوع در هم شکستن پشته باشیم.در این بخش باید مکانیزمهای زیرکانه ای را تشریح کنیم که باعث می شود، سیستم IDS نتواند شروع یک حمله بر علیه پشته را کشف کند.

سیستم IDS بهترین مشخصه ای که برای کف شروع یک حمله (بر علیه پشته) در اختیار دارد وجود کدهای NOP است چرا که نمی توان روی طول زیاد رشتهٔ داده حسایت نشان داد. (زیرا بعضی از پروسه ها بطور طبیعی داده های بسیار طولانی دریافت –یا ارسال – می کند.) بنابراین ویژگی کشف کدهای nop بهمراه کدهای اجرائی ماشین، ملاک کشف شروع حمله به یک پروسه محسوب می شود.حال نفوذگر چگونه می تواند از سیستم IDS فرار کند:

تکنیک فریب سیستم IDS، برای اولین بار توسط یک گروه امنیتی در کانادا بصورت دقیق تشریح و شبیه سازی شد. گزارش تفصیلی این تکنیک جهت آگاهی در http://www.ktwo.ca/securiy.securiy.html ارائه شده است. ابزار شبیه ساز این تکنیک ADMutate نامیده می شود. مکانیزم این ابزار برای فرار از سیستمهای IDS، در سه بخش زیر خلاصه می شود:

مجموعهٔ کدهای NOP را با مجموعهٔ متنوع و کاملاً تصادفی از دستورات ماشین جایگزین می کند بگونه ای که تاثیر هر دستور مشابه دستور NOP باشد؛ مثل:

MOV EAX, EAX ← بی تأثیر

AND EAX, FFFFH ← بی تأثیر

AND EAX, 0 ← بی تأثیر

.....

جانشینی دستورات NOP با این مجموعهٔ تصادفی از دستورات بی اثر، امکان تشخیص داده های مزاحم از داده های واقعی را مشکل می سازد، چرا که اولین معیار سیستم IDS برای آگاهی از حمله، کدهای NOP است.

برای آنکه کشف مزاحمت (IDS) کدهی اجرائی را دورن داده ها تشخیص ندهد، بخش کدهای اجرائی در رشتهٔ دادها با یک کلید مشخص و تصادفی XOR می شود. سپس چند دستور برای دیکود کردن دستورات اصلی قبل از کدهای اجرائی XOR شده (بعد از دستورات NOP)، قرار می گیرد.

برای آنکه قسمت اجرایی دیکود کننده ملاک تشخیص حمله قرار نگیرد، بخش دیکود کننده بصورت متنوع و مختلفی نوشته شده و به کد اضافه می شود. (عمل XOR را به چندین روش متفاوت می تون پیاده کرد و همچنین بین دستورات بخش دیکود کننده دستورات معادل NOP اضافه می شود)

در تحقیقات عملی که نرم افزار ADMutate انجام می دهد تبدیل کدهای NOP و کدهای اجرائی نفوذگر به کدهای پلی مورفیک است بگونه ای که توسط سیستم IDS کشف نشود.

ADMutate به این نکته دقت دارد که در بین کدهائی که ایجا می کند کد '0\' وجود نداشته باشد زیرا کد '0\' برای توابع رشته ای کد پایان رشته تلقی شده و با مواجهه با چنین کدی بقیهٔ رشته به پشته منتقل نخواهد شد. ایجاد کدهای پلی مورفیک برای حمله به پشته یکی از خطرناکترین و مرموزترین انواع حمله به شمار می رود.

حمله بر علیه کلمات عبور (password Attack)

در دنیای شبکه و سیستمهای کامپیوتری یکی از روشهای تامین امنیت، استفاده از کلمهٔ عبور است. در بسیاری از سازمانها فقط کلمات عبور هستند که از داده های محرمانه و حساس حفاظت می کنند. بدین معنا که برای هر کاربر یک کلمهٔ عبور در نظر گرفته می شود و او باید در هر بار ورود به سیستم (یا شبکه) آن کلمه را وارد نماید؛ در صورت صحت کلمهٔ عبور به او اجازهٔ ورود داده خواهد شد. در حقیقت امنیت مجموعه ای از اطلاعات به امنیت یک کلمهٔ عبور. گره خورده است. فاش شدن یک کلمهٔ عبور می تواند منجر به از دست رفتن کل اطلاعات کاربر و حتی شکسته شدن حریم کل سیستم (یا شبکه) شود. لذا امنیت کلمات عبور، از اهمیت بسیار ویژه ای برخوردار است.

بسیاری از حملات مخرب علیه یک شبکه از طریق کشف کلمات عبور شروع می شد.

برای هر کاربر شبکه حداقل یک کلمهٔ عبور برای ورود به شبکه تعریف می شود. برخی از کاربران مجبورنند برای استفاده از سرویسهای مختلف دهها کلمهٔ عبور را به خاطر بسپارند: یک کلمهٔ عبور برای ورود به شبکه و استفاده از منابع اشتراکی، یک کملهٔ عبور برای بررسی نامه های الکترونیکی، یکی برای احراز هویت در سرویس دهندهٔ پراکسی،

یکی برای ورود به سرویس دهنده FTP، یکی برای ورود به سیستم از طریق TelNet و نظایر آنها.

در اکثر سیستمها برای راحتی کاربران به آنها اجازه داده می شود تا شخصاً کلمات عبور مناسب برای خود انتخاب کنند. با این کار امنیت سیستم به عملکرد و سلیقهٔ کاربران وابسته می شود و یک کاربر سهل انگار می تواند امنیت دیگران را هم به خطر بیندازد. بطور معمول کاربران حسایت زیادی نسبت به امنیت سیستم نمی دهند چون اکثراً یا ناآگاهند یا آنکه به عملکرد سیستم بیش از اندازه مطمئن هسند و لذا برای راحتی خود از کلمات عبوری استفاده می کنند که به خاطر سپردن آنها زیاد سخت نیست: کوتاه است و گاهی بامعنی! لذا در بسیار از مواقع می توان با سعی و خطا کلمهٔ عبور یک کاربر را حدس زد. زیاد دیده شده است که یک کاربر (حتی مسئول شبکه) از اسامی هفته، اسم فرزند خود، شمارهٔ ماشین خود، نام خانوادگی یا شمارهٔ شناسنامهٔ خود به عنوان کلمهٔ عبور استفاده کرده است. چنین کلمات عبوری براحتی کشف خواهند شد. دزدیدن یا حدس زدن کلمهٔ عبور یک کاربر لقمهٔ بسیار لذیذی برای نفوذ گر است چرا که وقتی جای پای او در سیستم بعنوان یک کاربر معتبر باز شد/. عملیات بعدی او می تواند به آشکار شدن کلمهٔ عبور دیگران و نهایتاً فروپاشی کل شبکه منجر شود!

از همه بدتر برخی از اوقات کاربران از کمله عبور پیش فرض و نشناخته شدهٔ یک سیستم استفاده می کنند که براحتی کشف خواهد شد. شاید باو نکنید ولی زیاد هستند کاربرانی که از کلمهٔ عبور 123 یا 1234 استفاده می کنند!!!

حدس زدن کلمات عبور بروشهای حدس و خطا توسط ابزارهای خودکار، فرآیند چندان سختی نیست و یک تازه کار هم می تواند آنها را بکار بگیرد. اکثر این ابزارها رایگاننـد و به وفور یافت می شوند. به عنوان مثال اگر یک نرم افزار قادر باشد در هر ثانیه فقط هزار کلمهٔ عبور برای ورود به یک سیستم امتحان نماید، برای پیدا کردن یک کلمهٔ عبور چهار حرفی (با ۷۰ حالت مختلفی برای هر حرف شامل حروف کوچـک، بـزرگ و علامـات) فقط شش ساعت و چهل دقیقه طول می کشد. اگـر حـروف کوچـک و بـزرگ تفـاوتی نداشته باشند این زمان به شصت و سه دقیقه کاهش خواهد یافت. اگر کـاربر از کلمـات با منی استفاده کند ، با در اختیار داشتن یک فرهنـگ لغـت چهـل هـزار کلمـه ای زمـان کشف کلمهٔ عبور چهل ثانیه خواهد بود! حتی اگر بخشی از کلمهٔ عبـور متعلـق بـه یـک شخص با معنی و بخش دیگری بی معنی و ترکیبی باشد(مثل apple4x) باز هم حـدس کلمهٔ عبور (با سرعت هزار «تلاش در ثانیه» و در اختیار داشتن فرهنگ لغت چهـل هـزار کلمه ای) حداکثر هفده ساعت خواهد بود.

حدس کلمهٔ عبور و ورود به سیستم از طریق چند خط برنامه نویسی

حتی وقتی کلمهٔ عبور پیش فر از سیستم حذف شده باشد با زهم نفوذگر ناامیـد نخواهـد شد. روش بعدی او برای وورد به سیستم آنست که چند خط برنامه نویسـی کنـد. دقـت کنید که مشخصهٔ کاربری (User ID) آشـکار اسـت و محرمانـه تلقـی نمـی شـد. (مـثلاً administrator,admin,root مشخصه های معروف و پرکاربرد هستند) او برنامه ای مـی نویسد که به طور مکرر کلمات عبور مختلف را جهـت و ورد بـه یـک سیسـتم امتحـان

نماید. این برنامه یکی از این User ID ها را در نظر گرفته و کلمات عبور مختلف را امتحان می نماید. این برنامه همچنین یک فرهنگ لغت غنی در اختیار دارد. برنامه بطور خودکار و سریع یک کلمهٔ عبور حدسی را ارسال کرده و بررسی می نماید آیا این حدس درست بوده است یا خیر. در صورت نادرست بودن حدس، کلمهٔ عبور دیگری تولید و آنرا امتحان می نماید. این روند تا پیدا شدن کلمهٔ عبور و ورود به سیستم ادامه می یابد. صدها ابزار مختلف بدین منظور نوشته شده و در اختیار عموم قرار گرفته است که مشهورترین آنها عبارتند از :

نرم افزار Authforce نوشته شده توسط Zachary P.Landau، این نرم افزار تلاش می کند تا کلمهٔ عبور لازم برای ورود به سرویس دهنده وب (سیستم احراز هویت HTTP)) را بروش سعی و خطا پیدا کند. این نرم افزار در آدرس زیر برایگان عرضه شده است:

http://Kapheine.hypa.net/authforcelindex.php

برنامه های Brute-SSL و Brute-Web برای حدس کلمات عبور HTTP و HTTPS و ورود به زور در این سرویس دهنده ها (نوشته شده توسط گروه BeastMaste). کدهای این دو برنامه در آدرسهای زیر در دسترس هستند:

http://packetstorm.securify.com/Exploit-Code-Archivee/brurte-ssl.c

http://packetstorm.securify.com/Exploit-Code-Archivee/brurte-web.c

نرم افزاری جهت حدس کلمات عبور Windows و ورود به آن از راه دور که وسط Somarsoft نوشته شده است. قابل تهیه از آدرس:

http://packetstorm.securify.com/NT/audit/nt.remotely.crack.nt.passwords.zip

نرم افزا Xavier نوشته شده توسط LithiumSoft ، ابزار قابل انعطافی برای حدس کلمات عبور بسیار یاز برنامه های کاربردی و سرویس دهنده ها محسوب می شود:

http://www.btinternet.com/~lithiumsoft/

نرم افزار Hypnopaedia نوشته شده توسط Null String که برای حدس کلمات عبور پست الکترونیکی با پروتکل POP3 کاربرد دارد:

http://packetstorm.securify.com/Cackers/hypno.zip

به منظور تحقیق

یک مجموع از ابزارهای کسشف کلمهٔ عبور شامل موارد فوق، بهمراه کدهای منبع برخی از آنها (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می باشد آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

نکته ای که باید در مورد ابزارهای حدس کلمات عبور به آن اشاره کنیم آن است که در اکثر سیستمها هر گاه یک کلمهٔ عبور اشتباه باشد بطور عمدی بین پنج تا ده ثانیه، دریافت کلمهٔ عبور بعدی به تأخیرخواهد افتاد. یعنی در این سیستمها فقط در هر دقیقه شش تا دوازده بار می توان کلمات عبور مختلف را امتحان کرد.در این حالت امتحان کلمات عبور مختلف، ممکن است روزها، ماهها یا سالها طول بکشد که عملاً ممکن نخواهد بود. نفوذگر برای ورود به یک سیستم هیچگاه حوصله صبر کردن به مدت چند ماه را نخواهد داشت!

گذشته از تأخیر مصنوعی بین دو تلاش متوالی برای ورود به سیستم، در برخی از سیستمها هر گاه دفعات تلاش ناموفق از تعداد مشخصی تجاوز کند، حساب مربوطه (Accoant) بطور کامل غیرفعال می شود. غیر فعال شدن حساب کاربری یک نفر ممکن است دائمی یا مثلاً برای ۱۵ دقیقه باشد. در صورتیکه حساب کاربری یک نفر بطور دائم غیرفعال شود. فقط مسئول شبکه قادر خواهد بود آنرا فعال کند. شاید چنین روشی را مناسب تصور کنید و احساس شود بدین نحو هیچ نفوذ گری نخواهد توانست با سعی و خطا به سیستم وارد شود ولیکن مشکل دیگری را ایجاد خواهد کرد:

یک نفوذگر می تواند بدین نحو نوع دیگری از حمله برعلیه شبکه شکل می گیرد: حملهٔ نوع DOS (Denial of Service) که در فصل نهم به آن خواهیم پرداخت.

اگر در شبکه، سیستم IDS(Intrusion Detection System) نصب شده باشد، هر گاه تلاشهای یک فرد برای ورود به یک سیستم از تعداد مشخصی تجاوز کرد به سرعت آنرا گزارش می کند.

الگوریتم شکستن کلمهٔ عبور

عمل شکستن کلمهٔ عبور منوط به آنست که در ارتباط کلمات عبور رمز شده به نحوی استراق سمع یا ربوده شود. «عملیات استخراج و کشف کلمهٔ عبور از اطلاعات رمز شده اصطلاحاً Password Crack نامیده می شود.»

برای شکستن رمز کلمه عبور نیاز نیست که «کلید رمز» (Encryption Key) بدست آید چرا که از لحاظ عملی چنین کاری ممکن نحواهد بود. مکانیزم شکستن رمز کلمات عبور بسیار ساده است:

- ابتدا یک کلمه عبور بصورت حدسی تولید شده و طبق الگوریتمی مشابه با الگوریتم اصلی رمز می شود.
- وجود آن در فایل مرز شده بررسی می شود.
- اگر آن کلمه درون فایل پیدا شد کار تمام است در غیر اینصورت یک کلمهٔ حدسی دیگر تولید رمز و جستجو می شود.

این مکانیزم در شکل (۹-۷) نشان داده شده است. اگر چه این مکانیزم ساده است ولی وسیع نیست زیرا ممکن است روند تولید کمله عبور حدسی و جستجوی آن میلیاردها بار طول بکشد.

یک ابزار شکنندهٔ کلمه عبور Password Cracher حدسهای اولیه در مورد کلمات عبور را بروشهای متفاوتی تولید و امتحان می کند حدسهای اولیه نقش بیر مهمی در احتمال موفقیت آن ابزار در کشف کلمهٔ عبور دارند. روشهای تولید و امتحان حدسهای اولیه عبارتند از:

تهیهٔ یک فرهنگ لغت بسیار غنی از کلمات معنی دارد (یا بدون معنی ولی مصطلح و محاورهای)و امتحان تک تک آنها؛ برای زبانهائی مثل انگلیسی، فرانسه، ژاپنی و ... دهها نوع فرهنگ لغت تهیه شده است که می تواند بعنوان یک بانک اطلاعاتی در اختیار ابزار

رمز شکن قرار گیرد. (بعنوان مثال به فرهنگ لغت موجود در CD جانبی نگاهی بیندازید. مجموعهٔ لغات درون آن فراتر از یک فرهنگ لغت معمولی است!)

خوشبختانه در مورد زبانهائی مثل فارسی یا عربی که از راست به چپ نوشته می شوند، کلمات رمز در بسیار از موارد باید انگلیسی انتخاب شود و کلمات رمز فارسی (از راست به چپ) مورد قبول نیست. لذا بسیاری از فارسی زبانها کلمهٔ رمز خود را کلمه ای فارسی انتخاب می کنند ولی بصورت انگلیسی می نویسند. هیچ فرهنگ لغتی که به انگلیسی تایپ شده باشد ولی کلمات آن فارسی باشند (مثل Kamran یا tajrish) تا بحال شنیده نشده است. اگر چنین فرهنگ لغتی وجود داشت ممکن بود بیش از سی درصد از کلمات رزم براحتی کشف شوند!!!

بهر حال آزمایش کلمات موجود در یک فرهنگ لغت در بسیاری از مواقع موفقیت آمیز نخواهد بود زیرا اگر چه بسیاری از کاربران از کلمات بامعنی استفاده می کنند ولی با اضافه کردن یکی دو حرف پیشوند یا پسوند و یا تلفیق دو کلمهٔ با معنی (مثل Sweetapple) شکل کملهٔ عبور بگونه ای تغییر می کند که در فرهنگ لغت پیدا نخواهد شد. لذا باید به روشهای دیگر متوسل شد:

روش بعدی برای شکستن رمز تولید حدس اولیه از طریق جایگشتهای مختلفی است که یک کمله رمز می تواند داشته باشد. به این روش «ورود به زور یا Brute Force» گفته می شود. بدین منظور برای حدس یک کلمهٔ عبور شش حرفی تمام ترکیبات مختلف آن باید امتحان شود. در ابتدا حالات مختلف هر حرف، کاراکترهای a-z و 0-9 در نظر

گرفته می شود. اگر از ترکیبات مختلف کلمهٔ عبور بدست نیامد، این ترکیبات با اضافه شدن علاماتی مثل *!@#$ و & ... تولید می شود دقتی کنید که استفاده از چنین روشی اگر چه یقیناً به جواب خواهد رسید ولیکن ممکن است روزها، سالها یا حتی قرنها به طور بینجامد. اگر کلمهٔ عبرو به قدر کافی کوچک باشد می توان در عرض چند هفته آنرا پیدا کرد.

به عنوان یک مثال فرض کنید یک ابزار قادر باشد در هر ثانیه هزار کلمهٔ عبور را امتحان کند آنگاه برای یافتن کلمات عبور ٦ حرفی (با جایگشت ۵۰ حرف) معادل ($۵۰^{٦} \div ۱۰۰۰$) ثانیه یا شش ماده خواهد بود! برای پیدا کردن کلمهٔ عبور ده حرفی این زمان معادل سی هزار قرن است! بدین ترتیب روش«ورود به زور» یا Brute Force برای کلمات عبور بزرگ ناتوان خواهد بود.

روش «مختلط» (Hybrid) ، روشی است که از تلفیق دو روش قبلی بدست آمده و احتمال موفقیت آن بیشتر از هر دو روش می باشد. در این روش در ابتدا کلمات درون یک فرهنگ لغت غنی امتحان می شود. در صورت عدم موفقیت لمات درون فرهنگ لغت با اضافه کردن پسوند یا پیشوندهای یک حرفی، دو حرفی و چند حرفی امتحان می شود. این پیشوند یا پسوند از طریق جایگشت های مختلف تولید می شوند.

تجربه نشان می دهد که روش مختلط بسیار موفق است زیرا از لحاظ روانشناسی، کاربران کلمات عبور را بهمین روش انتخاب می نمایند. به ندرت کاربری کلمهٔ عبور PG3XQW3$W را انتخاب می کند چرا که به سرعت فراموش خواهد شد ولی احتمال

انتخاب کلمهٔ عبور apple$12 را انتخاب می کند چرا که به سرعت فراموش خواهد شد ولی احتمال اتخاب کلمهٔ عبور apple$12 زیادتر خواهد بود و زیرا بسادگی به خاطر خواهد ماند.

بخاطر داشته باشید که لازم نیست ابزارهای شکنندهٔ کلمهٔ عبور، بر روی ماشین قرباتی بکار گرفته شوند بلکه ابتدا کلمات عبور رمز شده بروشهای مختلفی (مثل استراق سمع – sniffing تشریح شده در فصل هشتم) ربوده می شود؛ بدین ترتیب عملیات شکستن و کشف آن بر روی ماشین نفوذگر (با آسودگی خیال و صرف وقت کافی) انجام خواهدشد.در این روش فقط کافی است کلمات عبور رمز و درهم شده (Hashed/Encrypted) دزدیده شود که آنهم راههای متوعی دارد. لذا بدینگونه سعی و خطا از راه دور و آنهم روی ماشین قربانی (که ممکن است سیستم IDS داسته باشد و حساب کاربر را مسدود کند)، انجام نخواهد شد بلکه با سرعت بسیار بالا، بدون وقفه و بدون تاخیر شبکه، بر روی ماشین نفوذگر انجام می گیرد.ابزارهای شکنندهٔ کلمهٔ عبور با پردازنده های امروزی قادرند هزاران کلمهٔ عبور را در هر ثانیه امتحان نمایند. بعنوان مثال کلمات موجود در یک فرهنگ لغت پنجاه هزار کلمه ای در کمتر از یک دقیقه امتحان می شود. واضح است که سرعت کشف کلمهٔ عبور مستقیماً به سرعت CPU در ماشین نفوذ گر وابسته است. اگر نفوذگر از ماشینهای نوع RISC و چند پردازنده ای استفاده کند تا مثلاً در هر ثانیه بیش ار یک میلیون کلمهٔ عبور امتحان شود، آنگاه بررسی و

امتحان تمام کلمات موجود در یک فرهنگ لغت پنجاه هزار کلمهای یا جایگشتهای سه حرفی (بصورت پیشوند و پسوند) در کمتر از پنج ساعت ممکن خواهد بود!

برای سالها روشهای شکستن کلمهٔ عبور مورد توجه عموم و بالاخص دانشجویان! بوده است امروزه ابزارای بسیار قدرتمندی برای این منظور معرفی شد است که برخی از آنها شهرت جهانی دارند:

ابزار LophtCrack: ابزاری بسیار ساده و توانمند تحت ویندوز 2000 و NT . این ابزار به اختصار Lopht گفته می شود و در آدرس زیر موجود است:

http://www.Lopht,\.com/Lophtcrack/

ابزار John the Ripper: یک شکنندهٔ کلمهٔ عبور برای محیط یونیسک . این ابزار توسط گروه Solar Designer معرفی شده و در آدرس زیر موجود می باشد:

http://www.openwall.com/john/

ابزار Crack نوشته شده توسط Alec Murrett: یکی از ابزارهای قدیمی ولی قدرتمند در محیط پونیکس/. این نرم افزار در آدرس زیر رایگان در اختیار عموم است:

http://www.users.dircon.co.uk/~crpto/

ابزار Pandora: این ابزار برای شکستن کلمات عبور مربوط به Novell Netware نوشته شده و توسط گروه Simple Nomad در آدرس زیر در دسترس قرار گرفته است:

http://www.nmrc.org/pandora/

ابزار Palm Crack: این ابزار کلمات عبور یونیکس و NT را می شکند ولی باید در محیط palmOS PDA اجرا شود. این ابزار توسط گروه Noncon نوشته و در آدرس زیر ارائه شده است:

http://www.noncon.org/noncon/download.html

برای آگاهی مسئولین شبکه از عملکرد ابزارهای شکننده کلمات عبور مکانیزم و قابلیتهای ابزارهای Lopht Crack و John the Rippwe معرفی شده بالا را تشریح می نمائیم:

نکته امنیتی:

امروزه نوع بسیار پیچیده و خطرناکی برای شکستن کلمهٔ عبور مطرح شده است، بدین نحو که نفوذگر حرفه ای ابزارهای شکنندهٔ کلمهٔ عبور را بطور پنهانی و در قالب یک ویروس (با قابیت انتشار) روی شبکه پخش می کند.این ابزار ها روی ماشینهای مختلف شبکه بصورت مخفیانه جا خوش می کنند. این ابزارها در اولین فرصت آدرس ماشنی آلوده را به نفوذگر اعلام کرده و منتظر فرمان می مانند. هرگاه نفوذگر بخواهد کلمهٔ عبوری را از حالت رمز درآورده و کشف کند بخشی از تلاش را به این ماشینها محول می نماید. حال اگر ده هزار ماشین در شرایط دنیا بدین نحو آلوده شده باشند و هر کدام در ثانیه هزار کلمهٔ عبور را امتحان نمایند توانی که نفوذگر بدست می آورد معادل یک ابررایانه خواهد بود!!!

از آلوده نبوده سیستمهای خود به چنین نرم افزارهایی اطمینان حاصل کنید.

پیکربندی LophtCrack

LophtCrack مطابق با پنجرهٔ شکل (۱۰-۷) بسادگی پیکربندی و تنظیم می شود کاربر می تواند از LophtCrack در سه حالت زیر بهره بگیرد.

- روش استفاده از فرهنگ لغت؛ LophtCrack بهمراه خود یک فرهنگ لغت پنجاه هزار کلمه ای دارد! (این فرهنگ لغت در هنگام نصب آن روی سیستم ذخیره می شود)
- روش ایجاد جایگشتهای مختلف حروف و «ورود به زور» (Brute Force)
- مکانیزیم این سه روش در بخشهای قبلی تشریح شدند. کاربر ابتدا با علامت زدن در گزینه های زیر استفادهٔLophtCrack از فرهنگ لغت و همچنین نوع حمله را مشخص می نماید:
- LANMAN
- NRLM

بگونه ای که در فصل چهارم اشاره شد در ویندوز NT و 2000 از دو روش برای رمز نگاری و درهم سازی کلمات عبور استفاده شده است:

الف) روش LM: این روش ضعیف است و اگر در شبکهٔ شما رمزنگاری کلمات عبور بدین روش انجام شده باشد بسادگی هر چه تمام و با صرف اندکی وقت شکسته خواهد شد.

ب) روش ntlm (یا NT Hash): این روش محکمتر و قدرتمندتر است.

گزینهٔLANMAN برای حمله به کلمات عبور رمز شده بـه روش LM و گزینـهٔNTLM برای حمله به کلمات عبور رمز شده به روش NTLM است که توسط کـاربر انتخـاب و فعال می شود.

کاربر ابتدا گزینه Brute Hybrid را فعال کرده و سپس تعیین مـی نمایـد کـه هـر کلمـهٔ فرهنگ لغت با جایگشتهای مختلف چند کاراکتر به عنوان پیشوند یا پسوند امتحان شود. مثلاً اگر عدد ۲ انتخاب شود برای هر کلمهٔ موجود در فرهنگ لغت مثـل apple حـالات مختلف applexx که مشتمل بر ۲۵۰۰ حالت (۵۰×۵۰) است، آزمایش می شود.

در LophtCrack می توان مبنای تولید کلمات عبور و امتحان آنها را صرفاً روش Brute Force انتخاب کرد. در این حالت کاربر در بخش Brute Force Attack از شکل (۱۰-۷) ضمن فعال کردن گزینهٔ Enabled، بایـد مجموعـهٔ کاراکترهـائی را کـه جایگشـتهای مختلف آن امتحان خواهد شد، تعیین نماید.

ابزار John the Ripper بر علیه کلمات عبور یونیکس

ابزار LophtCrack بسیار قدرتمند است ولی فقط کلمات عبور مبتنی بر سیسـتم عامـل windows را رمزگشائی می کند و برای محیطهای دیگر از جمله یونیکس و لینوکس بـه کار نمی آید. امرزوه بسیاری از سرویس دهده های مهم مبتنی بر سیستم عـالم یـونیکس (یا محیط های سازگار با آن مثل سولاریس یا لینوکس)پیکربرندی می شوند. لـذا بـرای حمله به کلمات عبور در این سیستمها ابزاهـاری دیگـری معرفـی شـده اسـت. یکـی از

قویترین آنها برنامهٔ John the Ripper است که در مقالات یا کتب به اختصار John (یا Ripper) نامیده می شود. ابزار John در محیطهای زیر قابل اجرا است:

- یونیکس
- لینوکس
- FreeBSD
- OpenBSD
- سولاریس
- Digital UNIX
- IRIX
- HP-UX
- AIX
- DOS
- Win 9x
- Win NT/20

به غیر از انعطاف باورنکردنی این ابزار، یک مشخصهٔ ویژهٔ در «John» گنجانده شده است:

ابزار John بگونه ای راحی شده تا برای استفادهٔ بهینهٔ از سرعت CPU، از قابلیتهای MMX در پردازنده های اینتل و همچنین مشخصه های پردازندهٔ K6 محصول AMD به بهترین نحو استفاده کند. چنین مشخصه ای، زمان آشکارسازی و شکستن کلمات عبور را کاهش خواهد داد.

گروه Dug Song نویسندهٔ برنامهٔ FragRouter (که در فصل ششم تشریح شد) نرم افزار John را بگونه ای توسعه داده است که می توان از آن برای آشکارسازی کلمات عبور مبتنی بر مکانیزمهای زیر استفاده کرد:

- S/Key One-password System
- AFS/Kerberos Ticket Ganting

درو کردن حسابهای کاربری در وب ۱۱ (Account Harvesting)

مثال بسیار خوبی از حملات خطرناک و مخرب بر علیه وب، حمله از نوع «برداشت حسابهای کابران وب» است. از ابتدای معرفی وب شاهد چنین حملاتی بوده ایم ولی در سالهای اخیر رواج وحشتناکی یافته است. در این تکنیک، نفوذگر تلاش می کند با توسل به روشهای زیرکانهٔ مشخصه های کاربری (User ID) و کلمات عبور کاربران وب را بروده و از آن برای اهداف خود استفاه کند. دقت کنید که نفوذ گر این عملات را صرفاً بر اساس روشهائی که در بخشهای قبلی بدان اشاره شد انجام نمی دهد بلکه از ضعف برنامهٔ کاربردی وب بهره می گیرد.

یکی روشهای کشف کلمات کاربری بررسی پیغامهای خطائی است که برنامهٔ کاربری تحت وب، در هنگام اشتباه وارد شدن مشخصهٔ کاربری یا کلمهٔ عبور بر می گرانند.

به شکل (۱۹-۷) و (۰-۷) دقت کنید، این صفحات وب ساختگی، مربوط به یکبانک فرضی بنام Mock Bank هستند. وقتی کاربری می خواهد به عنوان یک شخص مجاز به وب سایتی وارد شود در صفحهٔ اصلی، باید User ID و سپس کلمهٔ عبور خود را وارد نماید تا احراز هویت شود. شکل (۱۹-۷) نشان می دهد که وقتی مشخصهٔ کاربردی (User ID) شخص غلط است و چنین کاربری اصلاً وجود ندار، در پاسخ صفحهٔ وبی ارسال می شود که محتویات ظارهی آن در محیط مروگر به کاربر اعمال می کند که مشخصات کاربر جهت ورود به سایت غلط بوده و باید از نو وارد شود. شکل (۲۰-۷) هم از لحاظ شکل و هم محتوای صفحه، مشابه با شکل (۱۹-۷) است و از دیدگاه یک

کاربر معمولی هیچ تفاوتی با یکدیگر ندارند، ولیکن این دو صفحه در یک نکتهٔ ظریف با هم تفاوت دارند که از چشم یک حرفه ای پنهان نخواهد ماند.

X URL ظاهر شده در خط آدرس – خطر بالای مرورگر – در یک رقم با هم متفاوتند و همین رقم سرشار از معناست !!»

وقتی یک کاربر، User ID خود را غلط وارد کند، در URL ظاهر شده، رشتهٔ error=1 ظاهر شود در حالی که اگر User ID درست ولی کلمهٔ عبور اشتباه باشد در URL رشتهٔ error=2 دیده می شود. حال نفوذ گر از این اشکال کوچک استفاده بزرگ می کند!

بر اساس تفاوت دراین پیغامهای خطا، نفوذگر چند خط برنامهٔ ساده می نویسد تا بجای مرورگر با برنامهٔوب (در سمت سرویس دهنده) ارتباط برقرار کرده و با ارسال مشخصه های کاربری متفاوت، فهرست تمام کاربرانی که روی سیستم حساب دارند را کشف نماید. مکانیزم کشف مشخصه های کاربری به شرح زیر است.

در مرحلهٔ اول با استفاده از یک فرهنگ لغت و تولید جایگشتهای مختلف آن فهرستی از کلمات که بعنوان User ID، روی آن سیستم تعریف شده اند، کشف و ثبت می شود.

برای اینکار تمام حدسهائی که باید بعنوان User ID، روی آن سیستم تعریف شده اند، کشف و ثبت می شود . برای اینکار تمام حدسهائی که باید بعنوان user ID امتحان شوند، در قالب قواعد پورتکل HTTP و با یک کلمهٔ عبور صد در صد علط (مثل «1») به سمت آن برنامهٔ کاربردی ارسال می شود. هر گاه پیغام خطا بصورت error=2

برگشت داده شود به معنای آنست که User ID معتبر است ولی کلمهٔ عبور غلط می باشد؛ بنابراین User ID در فایلی ثبت می شود. سپس مرحلهٔ بعدی آغاز می شود.

User ID های معتبر بصورت جداگانه از فایل قبیل استخراج شده و با کلمات عبور مختلف امتحان می شوند. بطور معمول برنامه های کاربردی و به تعداد دفعات تلاش ناموفق برای ورود یک کاربر به سیستم با کلمات عبور متفاوت را اندازه گیری نمی کنند و لذا تلاش ناموفق با هر تعدادی حساب کاربر مربوطه را قفل نخواهد کرد؛ زیرا اگر چنین کاری انجام شود یک نفوذگر آماتور هم قادر خواهد بود تا تمام حسابهای کاربری کشف شده را قفل کند، بگونه ای که کاربران واقعی نتوانند به سیستم وارد شوند و این عمل خودش یک نوع حملهٔ دیگر محسوب می شود. بدین ترتیب هر تعداد تلاش برای وردد به سیستم پاسخی واقعی می گیرد و یک نفوذگر با صرف وقت کافی می تواند مجموعه ای از حسابهای کاربردی و کلمات عبور را به نفع خود درو و برداشت نماید!

راه چاره یست؟

روش مبارزه با دوری حسابهای کاربران وب

نقطه ضعف برنامهٔ کاربردی در مثال فوق بسیا ساده و شفاف است و راه حل آن نیز ساده خواهد بود. برای تمام برنام های کاربردی تحت وب که قرار است مشخصه کاربری و کلمهٔ عبور کاربران را بررسی کنند باید بدقت مراقب بود که پیغامهای خطار هیچ نماد روشن و بامعنی در خود نداشته باشند. سعی کنید صفحات پیغام خطا دقیقاً یکسان باشند و پیغامهای کوتاه و گویا روی آن نقش بینند برای خطاهائی که در هنگام احراز هویت

یک شخص تولید می شود پیغامهای مجزا و تفصیلی تولید نکنید. کار خودتان ساده تر خواهد بود!

حمله بر علیه مکانیزمهای نشست کاربران وب

بسیاری از برنامه های کاربردی تحت وب که مجبور به استفاده از Session ID هستند، در مورد ایجاد نشست و نظارت بر آن به شدت ضعف دارند. یک نفوذگر ممکن است یک نشت با برنامهٔ کاربردی برقرار کرده و یک Session ID معتبر دریافت کند؛ سپس آنرا عوض کرده و خودش را در قالب یک کاربر دیگر جا بزند و از آن به علل روند عملیات او را ادامه بدهد! برنامه های کاربردی تحت وب که Session IDرا به گونه ای ضعیف تولید می کنند به شدت ناامن خواهند بود زیرا اگر نفوذگر بطور عمدی Session ID را به گونه ای ضعیف تولید می کنند به شدت ناامن خواهند بود زیرا اگر نفوذگر بطور عمدی Session ID یککاربر دیگر را حدس زده و از آن بجایSession ID خود استفاده کند، برنامهٔ کاربردی وب گمان می کند که نفوذ گر همان کاربر مجاز است و از آن به بعد با نفوذگر مجاوزه می کند! یعنی با تعویض Session ID نفوذگر به یک کاربر مجاز دیگر تبدیل خواهد شد و بجای او سرویس خواهد گرفت. (البته خود کاربر اصی هم سرویس می گیرد؛ نفوذگر نیز مثل او و به نیابت از او سرویس می گیرد!!)

بسیاری از برنامه های کابردی وب که دهها هزار نسخهٔ از آنها در سر تا سر دنیا و منطبق با مقاصد و اهداف شرکتها و موسسات توسعه داده شده اند، از این نقطع ضعف رنج می روند و به محض آنکه نفوذگرSession ID یک کاربر مجاز را حدس زده و استفاده کند،

آن برنامهٔ کاربردی بسادگی به او سرویس می دهد. این نقطهٔ ضعف می تواند خسارات جبران ناپذیری به بار بیاورد. بعنوان مثال یکبانک خودکار و تحت وب را در نظر بگیرید که کاربران آن پس از برقاری یک نشست با برنامهٔ کاربردی آن بانک، عملیات مالی خود را انجام می دهند.اگر نفوذگر بتواند با حدس session ID متعلق به یک کاربر، خودش را بجای او حا بزند می تواند از حساب آن کاربر برداشت کند. حملاتی بدینگونه بسیار زیاد گزارش شده اند!

مکانیزیم حمله به نشستهای وب به شرح زیر است:

اولین چیزی که نفوذگر به آن نیاز دارد. تعیین Session ID یک کاربر مجاز است که در حال حاضر او نیز یک نشست برقرار کرده است. برای اینکار نفوذ گر ابتدا خودش یک نشست مجاز با آن برنامهٔ کاربردی برقرار می نماید و مقدار Session ID اختصاص داده شده به خودش را مشاهده می کند. سپس بررسی می کند که طول Session ID چند کاراکتر است، نوع کاراکترها چیست (عددی، الفبائی یا هر ترکیب دیگر) و در نهایت آرایش و ساختار آنرا تعیین می کند.

در دومین مرحله، نفوذگر چند خط برنامهٔ ساده می نویسد تا هزاران بار با کلمهٔ عبور خودش بعنوان یک کاربر مجاز به آن برنامهٔ کاربردی وارد شود (یعنی Login کند) و یک Session ID دریافت نماید. او این مقادیر را در جائی ذخیره می کند.

در سومین مرحلهٔ ، نفوذگر مقادیرSession ID جمع آوری شده را مورد تحلیلهای آماری قرار می دهد و سعی می کند تا فرمول تولید آن را بیابد، شاید بتواند Session ID یک کاربر دیگر را حدس بزند. موفقیت در این مرحله کار را برای او تمام می کند! زیرا:

در مرحله بعدی او با حساب شخصی و مجاز خود به آن برنامهٔ کاربردی وارد می شود و پس از دریافت یک Session ID معتبر، آنرا با مقدار متعلق به یک کاربر دیگر عوض کرده و بجای او (و در کنار او !!)سرویس می گیرد.

مراحل بدست آوردن Session ID های مختلف و استخراج آن بسیار ساده است:

اگرSession ID بهمراه URL باشد او URL های مختلف را درون یک فایل ذخیره می کند تا بعداً فیلد مربوط به مشخصهٔ نشست را تشخیص داده و با چند خط برنامهٔ نویسی آنها را از URL جدا کرده و در فایل دیگری بریزد.

اگرSession ID درون صفحهٔ وب ارسالی پنهان و جاسازی شده باشد کل صفحات وب را ذخیره کرده و سپس با چند خط برنامه نویسی در درون صفحات، برچبش های «<INPUT» را که دارای گزینهٔ «TYPE=HIDDEN » هتسند، جستجو و استخراج می کند.

اگر از «کوکی دائم» استاده شده باشد کار بسیار ساده تر است چرا که نفوذگر براحتی قادر خواهد بود فایلهای کوتاه و متنی کوکی را مشاهده و ویرایش نمایند. در مرورگر NetScape تمام فایلهای «کوکی دائم» در فایلی به نام Cookies.txt ذخیره می شوند و

بسادگی می توان توسط برنامهٔ آنها را دید. شکل (۲۲-۷) این فایل را در محیط Notepad نشان می دهد.

در محیط IE(Internet Explorer) فایلهای «کوکی دائم» که از سرویس دهنده های مختلف ارسال می شوند، بطور مجزا و با اسامی مختلف و بصورت «فقط خواندنی» روی شاخه ای بنام Cookies ذخیره می شوند. برای سوء استفاده از کوکیهای دائم نفوذگر با حساب کاربری خود به سرویس دهنده وارد شده و یک Session ID برای خود بدست می آورد؛ در این صورت یک فایل کوکی برای او ایجاد می شود سپس آنرا توسط یک ویرایشگر مثل Notepad باز کرده و مقدار آنرا بصورت دستی شبیه شکل (۲۲-۷) با session ID یک کاربر دیگر عوض کرده و نهایتاً انرا ذخیره می کند. پس از بستن مرورگر و اجرای مجدد، کوکی تقلبی ملاک کار قرار می گیرد که در آن Session ID یک کاربر دیگر عوض کرده و نهایتاً آنرا ذخیره می کند. پس از بستن مروگر و اجرای مجدد، کوکی تقلبی ملاک کار قرار می گیرد که در آن Session ID، متعلق به کاربر دیگر است. دقتی کنید که نفوذگر قبل از تغییر کوکی باید مروگر را ببندد و پس از آن، دوباره آنرا اجرا نماید زیرا فایل کوکی دائم فقط در هنگام بسته شده مروگر ذخیره و در هنگام باز شدن مروگر، مجدد باز می شود. روش تغییر «کوکی دائم» ساده و پیش پا افتاده است ولی بزرگترین معضل نفوذگر تغییر کوکی های موقت است که درون حافظهٔ مرورگر (RAM) ذخیره می شوند و نمی توان به آن دسترسی داشت.

حمله به کوکی های موقت (Persession Cookies)

در بخش قبلی اشاره شد که کوکی های موقت درون حافظهٔ مرورگر ذخیره می شوند و بنابراین نفوذگر دسترسی مستقیم بدان نخواهد داشت و لذا تغییر در Session ID ممکن نخواهد بود از دیدگاه برنامه نویسان وب، کوکی های موقت قابل اعتماد و غیرقابل تغییر هستند ولی در این بخش می خواهیم روش حمله به کوکی های موقت را معرفی کنیم تا این برنامه نویسان متقاعد شوند که در هر حالت باید برای تولید Session ID دقت کافی و وسواس داشته باشند، زیرا حتی کوکی های موقت که بصورت رمزنگاری شده و از طریق سوکتهای SSL منتقل می شوند، ناامن و قابل تغییر هستند!

نرم افزار Achilles یکی از قدرتمندترین ابزارهای حمله به کوکی های موقت (یا هر فیلد پنهان HTTP) محسوب می شود. Achilles توسط گروه امنیت سیستم Digizen در اکتبر سال ۲۰۰۰ معرفی شد. این ابزار (برای محیط ویندوز) در آدرس زیر ارائه شده است:

http://www.digizen-security.com/

(ابزار Achilles بهمراه مستندات آن در دیسک جانبی کتاب موجود است.)

بگونه ای که از شکل (۲۳-۷) مشخص است،Achilles در نقش یک پراکسی بین مروگر (متعلق به نفوذگر) و سرویس دهندهٔ هدف قرار می گیرد و داده های مبادله شده (منطبق با پروتکل http) را دریافت و در اختیار نفوذگر قرار می دهد. این داده ها در محیطی قابل ویرایش و تغییر، به نفوذگر نشان داده خواهد شد و چونSession ID درون

فیلدهای ارسالی و دریافتی قابل مشاهده است لذا نفوذگر فارغ از آنکه از چه روشی برای پیاده سازی Session ID استفاده شده می تواند آنرا ببیند و قبل از ارسال به سرویس دهنده، مقدار آنرا تغییر داده و سپس دستور ارسال بدهد! حال حتی اگر از کوکی نوع موقت هم استفاده شده باشد، مقدار Session ID روی صفحهٔ خروجی Achillles نقش خواهد بست!

Achilles قادر است با پروتکلهای HTTPS HTTP کار کند و عملاً نقش کنسول حمله به نشستهای وب را ایفا می نماید. با استفاده از ابزار Achilles به تمام انواع Session ID می توان دسترسی داشت و آن را دستکاری کرد.

از آنجایی که Achilles در قالب یک سرویس دهنده پراکسی کوچک پیاده سازی شده است لذا نفوذگر می تواند مروگر و پراکسی Achilles را همزمان بر روی ماشین خود اجرا کند یا آنکه پراکسی Achilles را بر روی ماشینی مجرا نصب نماید.در هر دو حالت باید تنظیمات Proxy Server در مرورگر به درستی تنظیم شده باشد تا مرورگر مجبور شود بجای ارتباط مستقیم با سرویس دهندهٔ وب، ترافیک داده های HTTP یا HTTPS بصورت واضح و قابل ویرایش پیش روی نفوذگر قرار گرفته است! هرگاه مرورگر یا سرویس دهنده، اطلاعاتی را مبادله کنند، Achilles دربین راه آنها را متوقف می کند تا نفوذگر تغییرات لازم را به آنها بدهد. پس از فشردن کلید Send (در پائین پنجره، سمت چپ) نفوذگر اجازه عبور آنها را صادر می کند.

Achilles از HTTPS نیز حمایت می کند. اطلاع دارید که HTTPS همان پروتکل HTTP است که در لایه زیرین از سوکتهای امن و رمزنگاری شده SSL استفاده می کند. بگونه ای که از شکل (۲۵-۷) ملاحظه می کنید بین Achilles و مرورگر نفوذ کننده، بطور همزمان دو ارتباط SSL برقرار می شود:

- یک ارتباط SSL بین مرورگر و Achilles
- یک ارتباط SSL بین Achilles و سرویس دهنده

Achilles بصورت درونی دارای یک «گواهینامه دیجیتالی – Digital Certificate» است تا بتواند با مروگر ارتباط SSL برقرار نماید! سرویس دهندهٔ وب نمی تواند بفهمد یک پراکسی بین او و مرورگر قرار گرفته است. فقط مرورگر نفوذ کننده از این موضوع مطلع می شود و با نمایش یک پیغام هشدار اعلام می کند گواهینامهٔ دیجیتالی Achilles مورد تائید نیست. با این حال چون مرورگر تحت اعتبار و متعلق به نفوذگر است با یک لبخند از این پیغام چشم پوشی خواهد کرد: همه چیز در اختیار نفوذگر است!

مقابله با حملات علیه Session ID

برای آنکه برنامه های کاربردی وب از حمله بروش دستکاری Session ID در امان بمانند برنامه نویسان وب باید به موارد زیر دقت داشته باشند. مورا امنیتی زیر به روشی که برای پیاده سازی مکانیزم Session ID بکار گرفته شده هیچ ارتباطی ندارد و حتی الامکان باید رعایت شوند:

اطلاعات مربوط به یک نشست باید طبق یکی از روشهای معمول امضای دیجیتالی یا یکی از روشهای رمزنگاری بصورت رمز شده یا در هم (Hashed) ارسال شوند.

- حتی اگر SSL استفاده می کنید، ترجیحاً اطلاعات مربوط به «نشست» که درون URL یا عناصر پنهان صفحهٔ HTML یا کوکی ها -cookies- جاسازی و نگهداری می شوند را قبل از ارسال رمزنگاری کنید.
- مطمئن شوید که مقدار Session ID بقدر کافی طولانی است. توصیهٔ مؤکد آنست که Session ID حداقل ده کاراکتر باشد.
- روالی که برای تولید Session ID انتخاب می کنید باید بقدر کافی پویا باشد و برای صفحات مختلفی که کاربر مرور می کند بطو مداوم عوض شود.
- در تولید Session ID حتماً به نحوی از پارامتر زمان (که بطور مداوم تغییر می کند) استفاده کنید و پس از تولید حتماً آنرا رمز نمایید.

رعایت دو مورد آخر باعث خواهد شد که نفوذ گر نتواند براحتی Session ID متعلق به یک کاربر دیگر را حدس بزند.

- موارد فوق الذکر را نه تنها برای Session ID، بلکه برای هر فیلد یا اطلاعاتی که مایل نیستند کاربر آنرا دستکاری کرده و برایتان پس بفرستد، رعایت کنید. تمام این موارد باید در سمت سرویس دهنده و در برنامهٔ کاربردی شما (به عنوان برنامه نویس وب) رعایت شود زیرا کاربران نمی توانند در مروگرشان هیچ تغییری ایجاد نمایند.

- پس از آنکه برنامهٔ کاربردی تحت وب خود را تکمیل کردید سعی کنید، با استفاده از Achilles روند تولید Session ID را بررسی کرده و خودتان سعی در نفوذ بـه آن نمائید تا نقاط ضعف آن آشکار شود!

نکتهٔ امنیتی:

اگر برنامه نویس وب از دادهائی که برای مرورگر ارسال می شود محافظـت نکنـد، هـیچ امنیتی برای داده و سای وب نمی توان متصور شد و حمله به آن سایت، بسـادگی امکـان پذیر است. اگر وب سایت شما به منظور عملیات حسـاس نظیـر بانکـداری یـا تجـارت الکترونیکی طراحی شده است وظیفهٔ شما چند برابر خواهد بود زیرا یـک نقطـه ضـعف کوک منجر به بحرانهای بسیار بزرگ می شود.

نکتهٔ امنیتی:

برخی از برنامه نوسان وب برای رمزنگاری و مخفی نگاه داشتن اطلاعات مربوط به یـک نشست، زحمت زیادی متقبل می شـوند ولـی ممکـن اسـت یـک اشـتباه کوچـک تمـام زحمات آنها را برباد بدهد این اشتباه کوچک آنست کـه وقتـی برنامـهٔ وب آنهـا مجبـور است صفحات مختلفی را به عنوان خروجی و به منظور فعل و انفعال با کاربر تولید کنـد باید در تمام صفحات Session ID تغییر کند و در ضمن رمز نگاری شـده باشـد. حـال فرض کنید از مجموع دهها صفحه وب فقط در یکی از آنها رعایت نکـات امنیتـی نشـده باشد و نفوذگر بتوانند با مشاهدهٔ Session ID، خودش را در قالب یک کاربر دیگـر جـا بزند. در این حالت به محض ربوده شدن نشست از یک کاربر مجاز، کنتـرل آن نشسـت

به طور کامل به دست نفوذگر خواهد افتاد زیرا اگر حتی در یکی از صفحات نشست، یک کاربر ربوده شود، آن نشست در مراحل بعدی با Session ID جدید و مربوط به کاربر دیگر ادامه خواهد یافت. لذا به عنوان برنامه نویس وب هوشیار باشید!

نکتهٔ امنیتی:

بسیاری از کاربران پس از ورود (Login) به برنامهٔ کاربردی وب فراموش می کنند تا Logout کنند و بهمین دلیل نشست آنها معلق باقی خواهد ماند. بعنوان مثال برخی از کاربران به محض آنکه کارشان با برنامهٔ تحت وب تمام شد مروگر خود رابسته و ارتباط خود را قطع می کنند در حالیکه نمی دانند اگر کسی بعد از آنها مرورگر را باز کرده و پس از وصل ارتباط ، با Session ID آنها وارد شود میتواند خودش را بجای آنها جا بزند.

لذا شما بایستی این سهل انگاری کاربران را به نحوی در برنامهٔ کاربردی خود جبران کنید. راه حل این موضوع آنست که برای Session ID تولید شده برای یک کاربر طول عمر در نظر بگیرید (مثالً ۱۵ دقیقه) و پس از گذشت زمان عمر آن نشست، تمام اطلاعات مربوط به آن نشست شامل Session ID پاک شود و کاربر برای ورود مجدد محبور باشد از نو کلمهٔ عبور وارد کند. (یعنی از نو Login کند).

به نحو مقتضی در سایت وب خودتان عمل Logout را برای کاربران تشریح کرده و از آنها بخواهید موکداً قبل از بستن مرورگر حتماً logout کنند.

نکتهٔ امنیتی:

اگر سایت وب شما عملیات حساسی را برعهده گرفته است و نگران حمله به آن هستید میتوانید با پرداخت هزینه، نرم افزار AppScan را تهیه کنید. این نرم افزار بطور خودکار سایت وب شما را پویش کرده و در آن نقاط ضعف برنامه نویس را که عالم نفوذپذیر آن خواهد شد، جستجو می کند:

http://www.sanctuminc.com/

حمله به برنامه های کاربردی وب بروش SQLPiggybacking

تقریباً در اغلب برنامه های کاربردی تحت وب به نحوی از بانک اطلاعاتی استفاده می شود. این برنامه ها از طریق وب دادهای ورودی کاربر را دریافت می کنند و بر اساس آن یک QUERY روی بانک اطلاعاتی خود ارسال می نمایند. در حقیقت فعل و انفعال برنامهٔ کاربردی تحت وب با بانک اطلاعاتی بر اساس تولید Query از داده های ورودی کاربر انجام می شود. بسیاراز برنامه های کاربردی برای فعل و انفعال با بانک اطلاعاتی خود از زبان SQL (Structured Query Language) استفاده می کنند. (فارغ از آنکه بانک اطلاعاتی از چه نوعی است: SQL,Access اوراکل یا پاراداکس) این برنامه های کاربردی از طریق صفحات وب یا کاربر در ارتباطند، داده های او را جمع آوری کرده و برای جستجو در بانک اطلاعاتی، تغییر یک فیلد یا رکورد (براساس عمل مورد تقاضای کاربر)، اقدام به صدور یک یا چند Query روی آن بانک اطلاعاتی می نمایند.

اگر چه همه چیز عادی و بی خطر به نظر می رسد ولی نقطه آسیب پذیر این گونه از برنامه ها آنجاست که تولید Query بر اساس داده هائی که کاربر مستقیماً در فیلدهای ورودی صفحهٔ وب وارد کرده انجام می شود. یعنی برنامهٔ کاربردی بصورت کاملاً ساده و معمولی محویات یک فیلد را که توسط کاربر پر شده است، در جلوی یک دستور SQL چسبانیده و آنرا جهت اجرا روی بانک اطلاعاتی ارسال و اعمال می کند. یک نفوذگر حرفه ای که با زبان SQL آشناست می تواند محتویات این فیلد را که مختص به ورود داده های خام است را با دستورات و گزینه های SQL پر می کند. چون از این داده ها مستقیماً برای تولید Query استفاده می شود ممکن است ناخودآگاه آن Query تبدیل به یک فرمان خطرناک شده و پس از اجرا، اهداف نفوذگر را برآورده نماید.

در حقیقت روش حمله به برنامه های کاربردی تحت وب با تکنیک SQL Piggybaking بر این اساس استوار است که نفوذگر بصورت هوشمندانه فیلدهای ورودی صفحه وب را با گزینه ها و دستورات زبان SQL به نحوی پر کند تا پس از تولید و اجرای یک Query بر اساس این فیلدها، اهداف او برآورده شود. یک نفوذگر زیرک با نام (مستعار) Rainforest Puppy خودش یک سایت Packetstorm Security را مورد حمله قرار داد. (جالب اینجاست که Packetstorm خودش یک سایت «نفوذگری – امنیتی» محسوب می شود!) او در یک مقالهٔ مشهور عملکرد خود را تشریح نموده است. این مقاله را در آدرس زیر مطالعه کنید:

"How 1 Hacked Packetstorm"

http://www.wiretrip.ner/rfp/p/doc.asp.id=42

مکانیزیم حمله بروش SQL Piggybaking ، به شرح زیر است:

در مرحلهٔ اول تک تک صفحات وب که شامل حداقل یک فیلد ورودی هستند بررسی شده و از فیلدهائی که با استخراج محتویات آنها مستقیماً یک Query ساخته خواهد شد، فهرست تهیه می شود. این فیلدهای ورودی می توانند مثالهای زیر باشند:

- فیلد User Name
- فیلد Account Number
- فیلد Product Name
- یا هر فیلد مشابه که کاربر باید آنرا با داده پر کند!
- در مرحله دوم این یلدها با کاراکترهائی نظیر مثالهای زیر مقدار دهی می شوند:
- کاراکتر (Quotation)
- کاراکتر (Double Quote)
- کاراکتر کاما
- کاراکتر سمی کالون؛

این کاراکتر ها در زبان SQL عملکردهای خاص دارند. بعنوان مثال علامت سمی کالون؛ در SQL برای تفکیک دو Query از هم که بصورت همزمان روی بانک اطلاعاتی ارسال، اعمال می شوند، کاربرد دارد. یا مثلاً علامت کوتیشن برای خاتمه دادن به عبارت SQL

تعریف شده است. وجود هر یک از این کاراکترهای کنترلی درون یک SQL Query ممکن است یک خطا با یک عملکرد غیرطبیعی را بهمراه داشته باشد.

پس از مقدار دهی فیلدها با کاراکترهای خاص و ارسال آن به سمت برنامهٔ کاربردی وب، نفوذگر واکنش سرویس دهندهٔ بانک اطلاعاتی و برنامهٔ کاربر را (که در قالب یک صفحهٔ وب پاسخ داده می شد) به دقت بررسی می نماید.

هر گونه واکنش غیرطبیعی نشان دهندهٔ یک نقطهٔ ضعف است و نفوذگر با تمرکز بر آن سعی می کند تا از طریق ارسال دستورات و عبارت SQL (در قالب داده های ورودی) عملیات مورد نظرش را روی بانک اطلاعاتی اعمال نماید. (یعنی نفوذگر در فیلدی که باید نام خانوادگی خود را وارد کند، زیرکانه یک دستور SQL مینویسد و کلید SUBMIT را فشار می دهد!)

برای هر نقطهٔ ضعف در برنامهٔ کاربردی، مجموعه ای از دستورات SQL بصورت سعی و خطا امتحان شده و پاسخ آنها بررسی می شود تا عملیات موفق و قابل انجام مشخص گردد.

نهایتاً نفوذگر با در اختیار داشتن نام فیلدها و همچنین فهرست فرامین قابل ارسال SQL، شروع به فعل و انفعال با بانک اطلاعاتی می نماید و اهداف خود را دنبال خواهد کرد!

شکل (۲۶-۷) یکی از صفحات وب سایت یک بانک ساختگی به نام MOCK BANK را نشان می دهد. در این صفحهٔ وب یک فیلد ورودی جهت درج شمارهٔ حساب کاربر علیه شده است . کاربر مجاز است تا شمارهٔ حساب بانکی خود را در این فیلد وارد کرده

و کلید SEARCH را فشار بدهد. در این شکل نفوذگر برای آنکه پاسخ برنامهٔ کاربردی وب را بداند یک شماره حساب بانکی نادرست و دروغین را در درون یک فیلد وارد کرده و کلید Search را فشار بدهد.

در این شکل نفوذگر برای آنکه پاسخ برنامه کاربدی وب را بداند یک شمارهٔ حساب بانکی نادرست و دروغی را درون یک فیلد وارد کرده وکلید Search را فشار می دهد. (مثلاً یک شمارهٔ بزرگ با تمام ارقام 1 وارد می کند.) پاسخ برنامهٔ کاربردی وب، در شکل (۲۷-۷) نشان داده شده است.

در صفحهٔ وب شکل (۲۷-۷) یک پیغام ساده مبنی بر آنکه شمارهٔ حساب مورد نظر یافت نشده، به کاربر نشان داده شده است. از دیدگاه نفوذگر اطلاعات بسیار با ارزشی در خط Lcaation (یا همان خط Address در مرورگر IE) وجود دارد. رشته ای که بر اساس آن یک جستجو روی بانک اطلاعاتی انجام شده در این خط مشاهده می شود.

حال نفوذگر شروع به بازی با این فیلد می کند و درون آن ترکیبات مختلف کاراکترهای کنترلی را درج می کند تا پاسخ سرویس دهندهٔ سمت مقابل را ببیند. پیغامهای خطا سرشار از نکات فنی هستند! مثلاً نفلوذگر رشتهٔ 11111111111111 را درون فیلد «شماره حیات » وارد کرده و آنرا جهت جستجو به سمت سرویس دهنده ارسال می نماید. همانطور که در شکل (۲۸-۷) مشخص شده یک پیغام خطا تولید و برای مرورگر برگشته است. این پیغام خطا همان چیزی است که نفوذگر به دنبال آن می گردد.

نفوذگر متوجه می شود که چون او یک کاراکتر (qoute) به انتهای شماره اضافه کرده این مشکل پیدا آمده است.دلیل آن این بوده که برنامهٔ کاربردی در سمت مقابل آنـرا گرفتـه و مستقیماً از آن یک Query به شکل زیر توسط برنامهٔ کاربردی سمت مقابل تولیـد و روی بانک اطلاعاتی ارسال و اعمال شده است.

SEECT *from account WHERE (userid= ‘10001’ and numbr=

چون در این Query دو علامت متوالی کوتیشن پشت سر هم ظـاهر شـده، آن Query در زبان SQL معتبـر نیسـت و اعمـال آن روی بانـک ، سیسـتم مـدیریت بانـک اطلاعـاتی (DBMS) را مجبور به رد Query و ارسال پیغام خطا کرده است. این پیغام خطا نفـوذگر را متوجه می کند که برنامهٔ کاربردی درسمت سرویس دهنده چگونه بـا بانـک اطلاعـاتی فل و انفعال دارد. او فرمول زیر را استنتاج وقتی نفـوذگر متوجـه شـود کـه مقـادیر وارد شده در فیلدهای ورودی یک صفحهٔ وب، مستقیماً و بدون هیچ بررسی خاص درون یک SQL Query جاسازی میوشد، کار تمام است و نفوذگر به مراد خود خواهد رسید! زیـرا او قادر خواهد بود با وارد کردن دستورات SQL بصورت حساب شده درون فیلد موردی شروع به فعل و انفعال با بانک اطلاعاتی نماید.

اطلاعات ارزشمند دیگری که از پیغام خطای ظاهر شه روی مرورگر، قابل استخراج است نام واقعی بانک اطلاعاتی و نام فیلدهای این بانک اطلاعاتی است. در مثـال قبـل نفـوذگر متوجه می شود که :

نام بانک اطلاعاتی حساب کاربران، account است.

نام فیلد مشخصهٔ کاربری userid است.

نام فیلد شماره حساب بانکی number است.

حال نفوذگر با داشتن این اطلاعات ، می تواند بصورت زیر عمل کند: فرض کنید او می داند UserID یک نفر که در Mock Bank شماره حساب بانکی دارد، ۱۰۰۲ است ولی کلمهٔ عبور او را نمی داند. او براحتی و در جلوی URL موجود (در خط Location مرورگر) عبارت زیر را جلوی عبارت =account درج می کند:

1111111111111 '+OR+USERID%3D' 10002

آنهائی که با HTTP و MIME آشنا هستند می دانند که علامت + در یک URL جانشین فاصلهٔ خالی (Blank) می شود و 3d% بجای علامت = قرار می گیرد. پس از ارسال رشتهٔ فوق به سمت برنامهٔ کاربردی، از رشتهٔ فوق مستقیماً یک SQL Query بصورت زیر ساخته شده و برای اعمال به سوی سیستم مدیریت بانک اطلاعاتی ارسال می شود:

SELECT * FROM account WHERE (userid='10001'

And numver= '1111111111111' or userid = '10002')

رشته ای که نفوذگر تنظیم و ارسال کرده است.

آنهائی که حتی اندکی با SQL آشنا هستند متوجه می شوند که یک Query به شکل فوق چقدر خطرناک و مضر است. عبارت فوق بیان می کند که از بانک اطلاعاتی ACCOUNT تمام مشخصات مربوط به کسی که کلمهٔ کاربری او '10001' و شمارهٔ حساب بانکی او 111111111111111 است انتخاب شود یا کسی که کلمهٔ کاربری او

'10002' است!!! همان چیزی که نفوذگر می خواهد . نتیجه این جستجو مشخصات حساب کاربری کسی است که شمارهٔ او ۱۰۰۰۲ است. بدون آنکه کلمهٔ عبور او لازم باشد. (در ضمن کسی که شمارهٔ کاربر او ۱۰۰۰۱ و کلمهٔ عبورش '11111111111111' است هم وجود ندارد!)

با ارسال Query فوق، مشخصات حساب بانکی شخصی با شمارهٔ ۱۰۰۰۲، مطابق با شکل (۲۹-۷) روی خروجی ظاهر می شود.

Piggybacking

Piggybacking به معنای سواری گرفتن یا کولی مفت گرفتن از کسی یا چیری است!

SQL Piggybacking بدین معناست که نفوذگر بجای ورود داده هائی که درون یک Query از نوع SQL جاسازی می شد، Query های مورد نظر خود را قرار می دهد و از آن داده ها در راستای اهداف خود سواری می گیرد.

خبر بد: در مثال فوق، با تنظیم دقیق یک رشتهٔ عملیاتی، از دستور SELECT سوء استفاده شده است. (بدون نیاز به کلمهٔ عبور، اطلاعات حساب شخصی یک کاربر استخراج شده است.)

خبر بدتر: اگراز دستور UPDATE سوء استفاده شود. نفوذگر براحتی می تواند اهداف خود را اعمال و درون بانک اطلاعاتی ذخیره کند!!

خطرناکترین نوع حمله از نوع فوق آنست که نفوذگر از طریق علامت جدا کننده ؛ (سمی کالون) چند عبارت SQL را بعنوان ورودی تنظیم کرده و برای سرویس دهنده

بفرستد . این کار در SQL امکان پذیر و قابل انجام است. بدین نحو نفوذگر قادر خواهد بود تا هر کاری انجام بدهد! تنها محدودیت نفوذگر آن خواهد بود که برنامهٔ کاربردی تحت وب از بین نتایج اجرای عبارات SQL فقط آنهائی را برخواهد گرداند که برنامه نویس وب آنها را تعیین کرده است لذا در بسیاری از مواقع نفوذگر پس از تنظیم و ارسال عبارات SQL، نتیجهٔ اجرای آنرا روی صفحهٔ وب مرورگر خود مشاهده نخواهد کرد. لذا او مجبور است حملات خود را بدون هیچ کنسول خروجی و نشانگر وضعیت به انجام برساند که البته برای یک حرفه ای چندان دشوار نیست. بدین نحو آیا هیچ امنیتی برای داده ها می توان متصور شد؟ پس به راهکارهای مبارزه با چنین حملاتی دقت کنید:

مقابله با حلات از نوع SQL Piggybacking

بهمان اندازه که حملهٔ نوع SQL Piggybacking خطرناک است ، مقابلهٔ با آن ساده و سهل خواهد بود. لذا برنامه نویسان وب باید به نکات امنیتی زیر دقت داشته باشند:

بهیچوجه به داده های ارسال شده توسط کاربر اعتماد نکنید و مستقیماً از آن Query تولید ننمایید. این داده ها حتی ممکن است بصورت سهوی دارای کاراکترهای کنترلی با نتایج غیر منظره باشند. بنابراین قبل از تولید Query از داده های کاربر، باید بدقت آنها را غربال کرده و هوشمندانه آنرا تحلیل (Purse) نمائید. این کار برای یک برنامه نویس حرفه ای وب کار دشواری نخواهد بود.

وقتی که داده های کاربر از طریق سرویس دهندهٔ وب تحویل برنامهٔ کاربردی شما می شوند باید همیشه آنها را آلوده فرض کنید، مگر خلاق آن ثابت شود. وقتی شما انتظار

داشته اید که کاربر یک مقداری عددی درون یک فیلد درج کرده و ارسال نماید ، در اولین مرحله باید عددی بودن آن ثابت کنید. هیچ کاراکتر های غیر عددی نباید در مقدار ارسالی کاربر مشاهده شود. وقی کاربر موظف به ورود اطلاعات دورن یک فیلد می شود که ماهیت آن غیر عددی است (مثل نام، آدرس ، کلمهٔ عبور و ...) باید درون رشتهٔ داده های رسالی، عدم وجود کاراکترهای زیر بررسی شوند:

تمام انواع کوتیشن (شامل ' و "): معنای آن در SQL خاتمهٔ یک رشته (string) است.

علامت سمی کالون (;) : معنای آن در SQL اتمام دستور جاری و شروع دستوری جدید است .

علامت * : معنای آن در SQL «انتخاب همگانی » یا «Wildcard selector» است.

علامت _ (underling)

کاراکترهای ویژه مثل (\r \n $ [ ] ( ) ^ % <> ~ ? * | \ &) تمام این علامات در SQL عملکرد تعریف شده ای دارند. بعضی از آنها عملکردهای محاسباتی و برخی منطقی و برخی نمادهای خاص SQL هستند.

برنامهٔ شماره در اولین گام باید این کاراکترها را غربال کرده و دور بریزد. برای بار سوم در این فصل اشاره می کنیم که استفاده از HTTPS (آ روی SSL) هیچ امنیتی در مقابل حملهٔ نوع SQL Piggybacking ایجاد نخواهد کرد . چرا که نفوذگر بروشی که در بخش قبل اشاره شد با استفاده از پراکسی Achilles از HTTPS فرار خواهد کرد.

نکتهٔ امنیتی:

بعنوان برنامهٔنویس وب یا مسئول امنیت شبکه بایستی اطلاعات بسیار جامعی در مورد ضعف برنامه های کاربردی تحت وب کسب کنید. نباید مشکلاتی که دیگران آزموده اند را شما از نو تجربه کنید . در آدرس زیر اطلاعات بسیار با ارزشی در این موارد به شمار عرضه می شود. این اطلاعات توسط گروه امنیت وب وابسته به «کنسرسیوم جهانی وب» تدوین شده و مهمترین مرجع در این زمینه به شمار می رود:

http://www.w3.org/Security/Faq/www-sequrity-faq.html

گام سوم: نفوذ از طریق استراق سمع در سطح لایهٔ شبکه

در فصل قبل روشهائی را که نفوذگر برای رخنه به سیستم از طریق برنامه های کـاربردی و سیسم عالم بکار می گیرد، بررسی کردیم.در این فصل نظر خود را معطوف به حملات از طریق لایه های زیرین به شبکه می نمائیم . اگر چه مـی تـوان نفـوذ بـه سیسـتم را از طریق لایه های لائی مثل بنرامه های کاربردی و سرویس دهنده هائی مثل وب، FTP یـا DNS برنامه ریزی کرد ولیکن نفوذ به سیستم از طریق لایه های زیرین شـبکه (مـثلاً در سطح لایهIP یا Data Ling) بدلیل انعطاف زیاد، بیشتر مورد توجه نفـوذگران حرفـه ای است. حمله در این سطح گاهی بسیر مخرب و خطرناکتر از حملات در لایه های بـالاتر است چرا که استراق سمع یا تحریف اطلاعات در سطح لایه های زیرین فقط متکـی بـه داده های یک سرویس دهندهٔ خاص در لایهٔ کاربرد نخواهد بود و نفوذگر از ایـن طریـق، برای رخنه و حمله به تمام سرویس دهنده های موجود روی یک ماشین (یـا حتـی روی یک شبکه) آزادی عمل پیدا خواهد کرد. به عنوان مثال در صـورت اسـتراق سـمع بسـته های IP، ترافیک تمام سرویس دهنده ها در اختیار نفوذگر قرار می گیـرد. در ایـن فصـل اینگونه حملات را کالبد شکافی کرده و تکنکیهائی کـه امـروزه بـه نامهـای Spoofing, Sniffing یا Sessinon Hijacking مشهورند و همچنین ابزارهائی نظیر Net Cat تشریح خواهیم کرد.

استراق سمع از هاب: Passive Snffing

بدلیل قیمت مناسب و کارآئی قابل قبول، بخش عظیمی از شبکه های اترنت با استفاده از هابهای معمولی پیاه سازی شده اند.در شبکهٔ مبتنی بر هاب، فریمی کـه هـر ایسـتگاه بـر روی کانال قرار می دهد توسط هاب دریافت شده و مجدداً بر روی بقیهٔ کانالهـا تکـرار[1] می شود. در یک عبارت ساده هاب هر چه را دریافت می کند بصورت فراگیـر[2] بـر روی تمام خروجیها ارسال می نماید. بنابراین ترافیک تولیـد شـده بـر روی یکـی از کانالهـای ورودی هاب روی تمام کانالهای خروجی شنیده خواه شد. ایـن مفهـوم در شـکل (۲-۸) نشان داده شده است.

در چین ساختاری هر گاه بر روی یکی از ماشینهای متـل بـه هـاب، یـک ابـزار Sniffer نصب شده و فعال باشد، بسادگی قادر به ربودن کلی فریمهای ارسالی از تمام ماشـینهای متصل به هاب خواهد بود. بسیاری از نرم افزارهای Sniffer بـرای محیطهـای مبتنـی بـر هاب معمولی نوشته شده اند. این ابزارها بطور عام «اسنیفر غیرفعال» (Passive Sniffer) نامیده می شوند و بطور آرام و مخفیانه فریمهای جاری بـر روی LAN را اسـتراق سـمع می نمایند.

استراق سمع از سوئیچ: Active Sniffing

برخلاف هابهای معمولی در شبکهٔ اترنت که فریم ارسالی روی یک کانـال را روی تمـام کانالها بصورت فراگیر ارسال می نمایند، عملکـرد سـوئیچها بسـیار هوشـمندتر از هـاب

---

[1] Repeat
[2] Broadcast

است، بدین نحو که با دریافت یک فریم از روی یکی ازروی یکی از کانالهای ورودی آنرا بر روی همهٔ کانالهای خروجی ارسال نمی کند بلکه قبل از ارسال، فیزیکی مقصد (destination Address) را بررسی کرده و فقط آنرا بر روی کانالی ارسال می کند که ماشین مقصد بدان کانال متصل است. بدین طریق یک ایستگاه متصل به سوئیچ قادر نخواهد بود ترافیک متعلق به ایستگاههای دیگر را بشنود. شکل (٤-٨) مفهوم کلی سوئیچ را نشان می دهد.

با چنین ساختاری در شبکهٔ اترنت مبتنی بر سوئیچ، نرم افزارهای Sniffer قادر نخواهند بود ترافیک جاری شبکه را دریافت کرده و در اختیار نفوذگر قرار بدهند، لذا استفاده از ابزارهای Snor، Sniffit و tcpdump در شبکه های مبتنی بر سوئیچ چندان فایده ای نداشته و فقط فریمهای ارسالی برای همان ماشینی که روی آن نصب شده را دریافت و در اختیار قرار می دهند و بقیهٔ فریمهای در حال تبادل روی شبکه از دید niffer مخفی می ماند.

برای جبران این مشکل که «اسنیفرها قادر به استراق سمع اطلاعات دیگران از سوئیچها نیستند». ابزارهای مفید و در عین حال خطرناکی تهیه شده است که سوئیچهای ضعیف و قدیمی را گمراه می کند. این نوع از سوئچها به وفور در بازار فروخته شده اند و ممکن است سوئیچ شبکه ای که شما مسئول آن هستید از همین نوع باشد.

استراق سمع از HTTPS و SSH

در آخرین نسخهٔ Dsniff قابلیت بسیار عجیبی برای گمراه کردن HTTPS گنجانده شده است همانطور که در فصل دوم اشاره شد، [1] SSL روشی برای انتقال مطمئن داده ها از طریق سوکتهای امن می باشد. این امنیت از طریق روشهای رمزنگاری و احزار هویت ایجاد شده است. [2] HTTPS همان پورتکل HTTP است با این تفاوت که از سوکتهای امن SSL در لایهٔ زیرین بهره می گیرد، بگونه ای که انتقال صفحات و داده های وب بصورت رمزنگاری انجام می شود.

سؤال کلیدی آنست که : ابراز Dsniff چگونه اطلاعات رمز شده مبتنی بر سوکتهای امن SSL را استراق سمع می کند؟

خوشبختانه روشی که در سوکتهای امن برای رمزنگاری و احراز هویت بکار گرفته شده، هنوز قابل شکست و رمزگشائی نیست. اگر گزارشی مبنی بر شکستن رمز اطلاعات رمز نگاری شدهٔSSl شدند بدین معنا که کسی توانسته اطلاعات ارسالی از طریق سوکتهای SSL را رمزگشائی کند، یا آن گزارش کاملاً دروغ است یا آنکه فصل جدیدی در دنیای ریاضیات و رایانه باز شده است!

قبل از آنکه روش Dssniff را برای استراق سمع از سوکتهای SSL، تشریح کنیم روش برقراری یک ارتباط HTTPS را یادآوری می کنیم. وقتی یک مرورگر با یک «سرویس دهنده مطمئن» ارتباط HTTPS بقرار می کند ، سلسلهٔ فعل انفعالات زیر اتفاق می افتد:

---

[1] Secure Socker Layer
[2] HTTP running over SSL

در اولین مرحلهٔ، سرویس دهنده گواهینامهٔ دیجتالی (Digiatal Certificate) خود را برای مروگر ارسال می کند تا هویت سرویس برای مرورگر محرز شود. این گواهینامهٔ دیجیتالی به مثابهٔ گواهینامهٔ رانندگی شما در یک جاده است.

دردومین مرحلهٔ، مرورگر تلاش می کند تا ابتدا اعتبار این گواهینامهٔ دیجیتالی را توسط یک سرویس دهندهٔ معتبر و قابل اعتماد جهانی که «مرکز گواهی امضا یا «Trusted Certificate Authority» نامیده می شود، تائید نماید.

اگر صحبت این گواهینامهٔ دیجتالی تائید شد یک ارتباط SSL بین مرورگر و سرویس دهندهٔ مورد نظر برقرار شده و یک کلید رمز تصادفی برای رمز تصادفی برای رمزنگاری اطلاعات در خلال نشست SSL در نظر گرفته می شود. مبادلهٔ کلید رمز، امن —Secure است و قابل اکشف و استراق سمع نخواهد بود.

پس از توافق روی این کلید رمز مبادلهٔ داده ها بین مرورگر و سرویس دهندهٔ بصورت رمزنگاری شده شروع می شود و چون رمزگشائی آنها از لحاظ عملی فعلاً ممکن نیست لذا یک نشست امن برقرار شده است!

با این توضیح کوتاه متوجه شده اید که Dsniff هرگز قادر به رمز گشائی اطلاعات و استراق سمع آنها نیست. تکنیک Dsniff برای استراق سمع از سوکتهای SSL به شرح زیر است:

Dsniff نصب شده روی ماشین نفوذگر، بصورت یک واسط بین سرویس دهنده و مرورگر قربانی می نشیند و خود را بجای سرویس دهندهٔ اصلی وانمود می کند. بدین

منظرو با استفاده از تکنیک dnsspoofing، ماشین قربانی بگونه ای فریب داده می شود تا از آدرس IP ماشین نفوذگر بجای آدرس سرویس دهندهٔ واقعی استفاده کند!

مرورگر قربانی به اشتباه با ماشین نفوذگر (بعنوان سیستم میانی که دزد داده ها است) یک ارتباط HTTPS (مبتنی بر سوکتهای امن) برقرار می کند.

Dsniff بلافاصله گواهینامهٔ دیجتالی خود را برای مرورگر قربانی ارسال می نماید؛ واگذر چهارمین گواهینامهٔ دیجیتالی معتبر است ولی قطعاً متعلق به سرویس دهندهٔ اصلی نیست .(مثل آنکه شما گواهینامهٔ معتبر دوستتان را به افسر راهنمائی و رانندگی نشان بدهید. با یک نگه به عکس روی آن و چهره شماه متوجه می شود که متعلق به خودتان نیست).

مرورگر قربنی، بلافاصله به کاربر اعلام می کنند که گواهینامهٔ دیجیتالی با آدرس حوزهٔ مورد نظر او تناقص دارد. (چون این گواهی دیجیتالی متعلق به Dsniff بوده است) متاسفانه مرورگرها بعد از اعلام یک هشدار، اختیار را به دست کاربر می دهند و از او کسب تکلیف می کنند که آیا برقراری ارتباط را با وجود چنین تناقضی ادامه بدهند یا خیر!

معملاً کاربران معملی اطلاعات زیادی در مورد گواهینامهٔ دیجتالی و اهمیت آن ندارند و بطور معمول یاد گرفته اند از پیغامهای هشدار چشم پوشی کرده و بدون مطالعهٔ پیام آن، سریعاً کلید Continue (ادامه کار) یا کلید Proceed را فشار بدهند تا سریعاً به کارشان بسند. دقت کنید که تعداد چنین کاربرانی زیاد است و بیش از ۹۰ درسد از استفاده

کنندگان اینترنت را شامل می شود! بدین نحو کاربر به واسط میانی که همان Dsniff است اعتماد کرده و پس از مبادلهٔ کلید رمز، یک نشست HTTPS با آن برقرار می کند.

از طرف دیگر Dsniff در نقش مرورگر ، یک نشست امن HTTPS با سرویس دهندهٔ اصلی برقرار می نماید و یک کلید رمز بین او و سرویس هنده توافق می شود . بدین نحو Dsniff عملاً در نقش یک سوریس دهندهٔ کوچک پراکسی (Proxy) دو ارتباط همزمان HTTPS برقرار خواهد کرد:

بین خودش بعنوان سرویس دهنده و مرورگر قربانی

بین خودش بعنوان مرورگر و سرویس دهندهٔ اصلی

Deniff داده های ارسالی از مروگر را گرفته و با کلید خودش رمز گشائی می کند و ضمن استراق سمع و ذخیره، آنها را مجدداً با کلید رمز دیگرش رمزنگاری کرده و و به سمت سرویس دهندهٔ اصلی ارسال می نماید. در مورد داده های ارسالی از سرویس دهنده نیز همین کار را انجام می دهد بدین نحو کاربر متوجه هیچ اختلالی در این جستجو و گشت و گذار در سایت وب مورد نظرش نخواهدشد در حالیکه Dsniff تمام اطلاعات او را ربوده و در اختیار نفوذگر قرار داده است!

برای درک عملکرد Dsniff به شکل (۸-۸) دقت کنید: در این مقال ماشین نفوذگر بین ماشینهای Bob و Alice قرار گرفته و اطلاعات آن دو را بین طر فین مبادله و استراق سمع می نماید. در ابزار Dsniff به قابلیت استراق سمع از سوکتهای SSL اصطلاحاً web mitm و ssh mitm گفته شده است:

web mitm برای استراق سمع از HTTPS (داده های وب)

ssj mitm برای استراق سمع از SSH (Secure Shell)

Mitm[1] اصطلاحی است که گروه Dug Song رانج کرده است و یک اصطلاح مسخره و انحرافی برای حمله نوع Person-in-the-Middle است. در تمام حملات نوع mith، یک ماشین با نیرنگ و بطور مخفیانه، خودش را بین یک سرویس دهندهٔ و مشتری قرار می دهدو همانند یک پراکسی (وکیل)، اطلاعات را بین آن دو مبادله می کند در حالی که یک نسخهٔ از آنها را در اختیار نفوذگر نیز قرار می دهد!

در ادامه، تکنیک web mitm را در ابزار Dsniff تشریح می نمائیم.این تکنیک بطور خلاصه در شکل (۹-۸) به تصویر کشیده شده است. مطابق با این شکل حمله از نوع web mitm در پنج مرحله انجام می شود:

در مرحله دوم نفوذگر طبق روشی که بنام Dns Spoof در قبل تشریح شد، ماشین قربانی را به گونه ای گرماه کرده تا مرورگر او آدرس حوزهٔسایت وب مورد نظر کاربر را به آدرس ماشین میانی بنگارد و به اشتباه نشست HTTPS را با ماشین نفوذگر برقرار نماید. فرض کنید او می خواسته با آدرس www.xyz.com نشست برقرار کند در حالی که با ارسال بسته های جعلی DNS توسط DNSSpoofing گمراه شده و آدرس IP ماشین نفوذگر را با آدرس IP سرویس دهندهٔ واقعی، اشتباه می گیرد.

[1] Monkey-in-th-Middle

در مرحلهٔ سوم، مرورگر قربانی یک ارتباط HTTPS مبتنی بر سوکتهای SSL با نفوذگر برقرار می کند . تمام داده های ارسالی او به برنامهٔ web mitm روی ماشین نفوذ گر تقدیم می شود.

در مرحلهٔ چهارم، ماشین نفوذگر با استفاده از نرم افزار web mitm نقش یک پراکسی را بازی کرده و پس از برقرری ارتباط با مرورگر قربانی، خودش یک ارتباط SSL با سرویس دهندهٔ مورد نظر کاربر برقرار می کند. حاللا ارتباط کاربر با سرویس دهندهٔ مورد نظرش بقرار شده و شروع به مبادله داده با آن خواهد کرد در حالی که ارتباط کاربر با سرویس دهدهٔ مورد نظرش برقرار شده و شروع به مبادله داده با آن خواهد کرد در حالی که نفوذگر این داده ها را می بیند.

در مرحلهٔ چهارم: web mith تمام داده های ارسالی از سرویس دهنده به مرورگر (و بالعکس) را دریافت کرده و در یک پنجره ساده نشان می دهد. نفوذگر می تواند آنها را ذخیره کند یا درون آنها به دنبال اطلاعات حساس و هم مثل کلمات عبور بگردد.

نکته:

موفقیت نفوذگر در این جبهه منوط به آنست که کاربر پیغامهای هشدار در مورد تغییر گواهینامهٔ دیجتالی سرویس دهنده را نادیده بگیرد.

اندکی دقت توسط کاربر می تواند از موفقیت چنین حمله ای جلوگیری کند. بگونه ای که اشاره شد تمام مرورگر ها در مواجعه با گواهینامهٔ دیجتالی که متعلق به یک سرویس دهندهٔ وب نیست، بلافاصلهٔ پیغامهای هشدار صادر و از کاربر کسب تکلیف می کنند.

مرورگر Netscape چند پنجره هشدار نظیر آنچه در شکل (۱۰-۸) می بینید به کاربر عرضه می کند . اگر کاربر روی این چهار پنجره کلیدهای Next یا Contimue را فضار بدهد عملاً قربانی یک توطئه شده و نفوذ گر در استراق سمع داده هایش موفق گردیده است! شکل (۱۱-۸) پیغام هشدار مرورگر Internet Explorer را نشان می دهد که بسیار ساده تر از پیغامهای Netscape است و متاسفانه شرایط روانی برای چشم پوشی کاربر از آنرا فراهم کرده است.

نکته:

web mitm و DNS Spoof هر دو بعنوان اجزای نرم افزار Dsniff محسوب می شوند که در چند بخش گذشته قابلیتهای دیگر آنرا بررسی کردیم.

در شکل (۱۲-۸) پنجرهٔ خروجیweb mitm در حین استراق سمع از سوکتهای SSL نشان داده شده است.در این پنجره تمام آنها بین مروگر قربانی و سرویس دهنده رد و بدل می شود به نفوذگر نشان داده شده است. ولی بطور معمول بخش کوچکی از آنها برای نفوذگر مفید است: کلمهٔ عبور و مشخصهٔ شناسائی کاربر! او در درون داده های وب، اطلاعات مورد نظرش را پیدا می کند و کار برای قربانی تمام است.

مشابه با همین روش، ابزار ssh mtim از نشستهای مطمئن مبتنی بر SSL استراق سمع می کند. اگر چه برای برقراری نشست SSH گواهینامهٔ دیجیتالی ارائه نخواهد شد ولی باید یک کلید رمز برای نشست انتخاب گردد. نفوذگر دقیقاً مثل روش web mith بین سرویس دهنده و کاربر قرار می گیرد. Ssh mitm دو نشست همزمان برقرار می کند:

یک نشست SSH بین خودش و کاربر

یک نشستSSH بین خودش و سرویس دهنده

نشست بین ماشین نفوذگر که برنامۀ ssh mith روی آن اجرا شده و ماشین سرویس دهنده مشکل خاصی ندارد ولیکن برنامۀ مشتری (SSH Client) یک پیغام هشدار به کار بر خواهد داد مبنی بر آنکه کلید رمزی که سرویس دهنده پیشنهاد کرده را نمی شناسد. پس از این پیغام منتظر صدور فرمان از کاربر می شود. برنامه های مختلف SSH، پیغامهای هشدار متفاوتی ارائه می کنند. به عنوان مثال OpenSSH در محیط یونیکس پیغام زیر را به کاربر نشان می دهد:

WARNING: Host Indentification has changed!

IT POSSIBLE THAT SOMONE IS DOING SOMETHING NASTY!

Someone could be eavesdropping on your right now

(man-in-the-middle attack)! It has also possible that the nost-key has just

been changed. Please contact ypur system administrator.

کاربران بی دقت و بی تجربه ممکن است پس از اندکی تامل دستور ادامۀ کار را صادر کنند. در این حالت نشست بین او و سرویس دهندۀ اصلی، تحت نظر نفوذگر قرار گرفته و داده های او بسادگی استراق سمع خواهد شد.

نکته امنیتی:

به تمام کاربران و همکاران خود در شبکه هشدار بدهید که هر گاه با پیغامهای امنیتی در مورد عدم اطمینان به گواهینامهٔ دیجیتالی (Digital Certificate) یا تغییر کلید رمز مواجه شدند، از ادامهٔ کار منصرف شده و با شما (به عنوان مسئول شبکه) تماس بگیرند.

نکتهٔ آخری که باید به آن اشاره کنیم آنست که Dsniff فعلاً از پروتکل نسخهٔ SSH Version 1.0 حمایت می کند و در مواجهه با نشستهای مطمئن مبتنی بر SSHVersion2.0 ناتوان است.

مقابله با استراق سمع (مقابله با اسنیفرها)

بگونه ای که ملاحظه کردید نفوذگر از طریق ابزارهائی چون Dsniff قادر است تمام ترافیک شبکهٔ را مورد دستبرد و تجاوز قرار بدهد و در چنین حالتی حریم شبکه عملاً پایمال شده است و هیچ امنیتی برای جریان اطلاعات نمی توان متصور شد. بنابراین مسئول شبکه باید موارد زیر را به دقت مد نظر قرار بدهد. در صورتی که امکان داشته باشد هیچ انتقال و مبادلهٔ داده ای انجام نشود مگر آنکه رمزنگاری شده باشد؛ بدین منظور از پروتکلهای HTTPS (بجای HTTP) برای مبادلهٔ داده های وب، SSH)f[hd Telnet kasjihd lul,gd یا S/MIME (FTP یا PGP برای رمزنگاری نامه های الکترونیکی استفاده شود.

سعی کنید برای رمزنگری در سطح لایهٔ شبکه از IPSec بهره بگیرید. امروزه بسیاری از سیستمهای عامل مثل ویندوز از IPSec حمایت می کنند.

بعنوان مسئول یک شبکه با عضو گروه امنیت سیستم) هیچگاه از راه دور با دیوار آتش، مسیریاب یا سرویس دهنده های حساس خود، نشست TelNet برقرار نکنید. زیرا در این صورت داده های ارسالی یا دریافتی بر روی کانال اشتراکی شبکه ظاهر می شود و ممکن است کلمات عبور و مجوزهای دسترسی استراق سمع شود. در این صورت قلب سیستم شما در اختیار نفوذگر قرار می گیرد. برای بکارگیری و کارکردن با مسیریاب یا دیوار آتش مستقیماً از خود سیستم یا ترمینال متصل بدان استفاده کنید تا داده ها از طریق کانال اشتراکی شبکه مبادله نشود. هیچگاه در این مورد جانب احتیاط را فرو نگذارید.

هر گاه پیغامهای هشدار در مورد یک نشست SSL گزارش می دهد که کلید عمومی سرویس دهنده (public key) بطور مرموزی تغییر کرده، نگران شده و تا دلیل آنرا نفهمیده اید راحت ننشینید.

اگر بودجهٔ شما اجازه می دهد حتی الامکان بجای استفاده از هاب از سوئیچ استفاده نمائید زیرا نفوذگر در مواجهه با سوئیچ بسیار محدود شده و قادر به استراق سمع تمام ترافیک شبکه نخواهد بود.

برای آنکه نفوذگر قادر نباشد از طریق تکنیک Arpspoofing، ترافیکی شبکه را به سمت ماشین خود برگرداند، سعی کنید جدول ARP را از حالت پویا (Dynamic) درآورده و بصورت دستی آنرا تنظیم کنید تا عمل Arp spoofing بی تأثیر شود. اگر چه تنظیم جدول ARP برای شمار بسیار دشوار خواهد بود. (بالاخص در شبکه های محلی برزگ) ولی مصیبت یک حمله و استراق سمع گاهی شدید تر است!

فریب دادن ماشینها با آدرسهای IP ماشین مبداء (نفوذگر) بطور جعلی و دروغین تنظیم می شود. این تکنیک که IP Address Spoofing نامیده شده است، میتواند دلایل متعددی داشته باشد:

نگر با آدرس دهی اشتباه، امکان تعقیب و کشف ماشینس را از طرف مقابل می گیرد چرا که بسته هائی که از طرف ماشین او ارسال می شوند آدرس مبداءی دارند که متعلق به یک ماشین بیگناه (یا موهوم) در شبکه است.

از طریق آدرس دهی دروغ، نفوذگر گاهی موفق به عبور بستهای IP خود از لابه لای فیلتر یا دیوار آتش یک سیستم که به آدرسهای IP حساسیت دارند، خواهد شد.

در بخشهای قبلی بون آنکه به تکنیک IP Spoofing اشاره ای شود روشهائی را توضیح دادیم که به این تکنیک متکی هستند. به عنوان مثال در توضیح ابزار Nmap (فصل ششم) اشاره شد که نفوذ گر برای آنکه ناشناس بماند. سعی می کند بسته هائی را با آدرس IP جعلی ارسال نماید تا سیستمهای کشف مزاحمت (IDS) نتوانند هویت واقعی او را کشف کنند یا مثلاً ابزار Dsniff (تشرح شده در همین فصل) در تکنیکDnsspoof از همین حق استفاده می کند، زیرا بستهٔ پاسخ DNS (ارسالی از طرف نفوذگر) دارای آدرس جعلی IP متعلق به ماشین DNS است.

مقابله با انواع فریبکاری متکی به آدرسهای IP جعلی

روشهای عملی بسیار مؤثری برای مقابله با حملات متکی به آدرسهای IP جعلی وجود دارد:

مطمئن شوید فرمول که پروتکل TCP برای تولید Seq No بکار می رود، غیرقابل تخمین بوده و استخراج فرمول تولید آن بسیار سخت باشد. بدین منظور همیشه با شرکت طراح سیستم عامل خود در تماس بوده و آخرین نرم افزارهای اصلاحی (Patch) را بر روی سیستم خود نصب کنید. جهت بررسی بیستر موضوع از نرم افزار Nmap که در فصل ششم معرفی شده استفاده نمائید. این نرم افزار قادر است فرمول تولید عدد تصادفی بار Seq No در بسته های SYN-ACK را تحلی کرده و اگر قادر است فرمول تولید عد تصادفی برای Sep No را بر علیه سیستم خود آزمایش کردید و قاعده ای بر آن حاکم باشد آنرا کشف کند.اگر Nmap را برعلیه سیستم خود آزمایش کردید و قاعده ای شفاف را به شما ارائه داد، نگران شوید و به سرعت به شرکت طراح سیستم عامل خود گزارش بدهید و مشورت بخواهید.

اگر از سیستم عامل یونیکس بهره می گیرید. سیستم را بگونه ای پیکربندی کنید تا تمام دستورات راه دور (r-commands) از طریق نشست های امن (SSH) برقرار شود. بدین ترتیب احراز هویت طرفین ارتباط ، بروشهای مدرن رمزنگاری (مثل Keberos) انجام خواهد شد و لذا هیچ ماشین ثالثی نمی تواند با تکنیکهای ضریب (Spoofing) خودش را بجای دیگری وانمود کند.

از فیلترهای anti-spof بر روی مسیریابها یا دیوار آتش شبکهٔ خود بهره بگیرید. این فیلترها از ایده بسیار ساده و در عین حال قدرتمندی بهره می گیرند. برای درک این ایده به شکل (۱۸-۸) دقت کنید. هر گاه بسته ای از بیرون شبکه بخواهد بدرون آن وارد شود

در حالی که که آدرس مبداء آن متعلق به ماشینی در دورن شبکه است، قطعاً اشکالی وجود دارد و فیلتر anti-spof آنرا حذف خواهد کرد. ممکن است این اشکال ناشی از یک حمله باشد یا آنکه پیکربندی ماشین مبداء صحیح نبوده است. بهردلیلی که به بسته خارجی از آدرسهای متعلق به ماشینهای درون شبکه بعنوان آدرس مبداء استفاده کرده باشد، باید حذف شود. نکته ای که اشاره بدان خالی از لطف نیست آنست که فیلترهای anti-spoof معمولاً دارای دو قسمت هستند:

الف) فیلتر پردازش کنندهٔ ترافیک ورودی به شبکه که اصطلاحاً ingress گفته می شود: برای آنکه حملات از بیرون بر علیه شبکه شما هدایت نشود نصب چنین فیلتری برای شبکهٔ شما الزامی است. چرا که باید جلوی ورود تمام بسته های حاوی آدرسهای جعلی به شبکه گرفته شود.

ب) فیلتر پردازش کننده ترافیک خروجی از شبکه که اصطلاحاً egess گفته می شود. اگر چه ممکن است نصب چنین فیلتری چندان مهم به نظر نرسد ولیکن به دو دلیل ، نصب آنرا پیشنهاد می کنیم. اولاً ممکن است که نفوذگر با توسل به یک روش خاص؛، در ییاز سرویس دهنده های شبکه شما مثل DNS یا web Server رخنه کرده باشد و تمایل داشته باشد از از طریق شبکهٔ شما حمله ای را برعلیه شبکهٔ دیگر ترتیب بدهد. ثانیاً ممکن است یکی از کاربران مجاز شما هوس هدایت یک حمله از درون شبکه شما بر علیه یک شبکهٔ دیگر به سرش بزند که بدلایل قانونی برایتان دردسرساز خواهد شد.

مسیریابهای شبکهٔ خود را به گونه ای پیکربندی کنید تـا از گزینـهٔ Source Routing در بستهٔ IP حمایت نکنند. بعنوان مثال تمام مسـیریابهای سـیکو (Cisco) از فرمـان « no ip sourceroute» حمایت می کنند که عمل Source Routing را غیرفعال می نماید.

فیلترهای مسیریاب خود را بگونه ای تنظیم که تمـام بسـته هـائی کـه در فیلـد Option گزینهٔ Source Routing دارند را حذف نماید. هر چند ممکن اسـت ایـن عمـل اشـکال زدائی از شبکهٔ شما را اندکی مشکلتر کند ولی ارزش آنرا دارد.

حتی الامکان برای راحتی خود از مفهوم «اعتماد در سیستم عالم Unix Trust–» یونیکس استفاده نکنید. زمانی می توانید از این قابلیت استفاده کنید که به عملکرد فیلترهای نصب شده روی دیوار آتش یا مسیریاب خود مطمئن باشید.

NarCat: نرم افزار همه منظور و قدرتمند شبکه

تکنیکهای «استراق سـمع» (Eavesdropping/ Sniffing)، «اسـتفاده از آدرسـهای دروغ» (IP Spoofing) و «ربودن نشست» (Session Hijacking) روشهای موثری در خـدمت به نفوذگر محسوب می شوند. نفوذگر بـا اسـتفاده از ایـن ابزارهـا در سـطح پروتکلهـای TCP/IP نفوذ خودبه شبکهٔ و بهره برداری از آنرا پایه ریزی می کنـد. ایـن فصـل کامـل نخواهد بود مگر آنکه ابزار NetCat را معرفی کنیم. ایـن نـرم افـزار همـه منظـوره بـرای مسئول شبکه. دانشجویان و نفوذگران! بسیار مفید و قابل انعطاف است.

Netcat

سخن رایجی در مورد Netcat نقل می شود که «اگر یک نفوذگر یا مسئول شبکه مجاز به انتخاب فقط و فقط یک ابزار باشد. بهترین انتخاب همین NetCat است.» این نرم افزار که در عین قدرت و قابلیت بالا بسیار ساده و کوچک است گاهی با عنوان« Swiss Army knife of Nerwork Tools » عرفی می شود! این ابزار با قابلیتهای بسیار قدرتمند ، زیر 100Kyte حجم دارد.

هدفی که این نرم افزار دنبال می کند آنست که بسادگی بتوان داده های را بین دو ماشین روی شبکه مبادله کرد. NetCar قادر است هر گونه داده ای را بر روی هر پورت TCP یا UDP مبادله کند . این نرم افزار به سادگی در حالت مشتری با یک شماره پورت ارتباط برقرار کند.

NetCar طبق معمول برای محیطهای مبتنی بر یونیکس (شامل یونیکس، لینوکس، اولرتریکس، SunOS ،Solaris، ALX و IRIX) طراحی و در سال ۱۹۹٦ معرفی شد. این ابزار به رایگان در آدرس http://www.Lopht.com/users/lopht/nt/nc110.tgz در دسترس همگن قرار دارد. در سال ۱۹۹۸ نسخهٔ مبتنی بر ویندوز آن توسط Weld Pond نیز عرضه شد. این نرم افزار نیز در آدرس http://www.lopht.com/~weld/netcat/ قابل تهیه است. نسخهٔ یونیسک و ویندوز کاملاً با هم سازگارند و لذا در محیطهای متفاوت به سادگی می توان مبادلهٔ داده ها را از طریق NetCat ممکن ساخت. در سمیناری به سال ۱۹۹۹ در لاس وگاس یکی از

پرهیاهوترین و مفیدترین سخنرانی ها، ارائه مقاله ای بود با نام « NetCat دوست شماست!» در طی آن قابلیتهای این نرم افزار ساده برای آزمون دیوار آتش و بررسی قواعد آن تشریح شد.

به منظور تحقیق

برنامهٔ NetCat بهمراه کدهای منبع آن(به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می باشند. آدرس موقعیت آنها بر روی CD، در انتهای فصل مشخص شده است.

با استفاده از NetCat می توان از طریق هر پورت TCP یا UDP با هر پورت TCP یا UDP دیگر، داده مبادله نمود. بگونه ای که در شکل (۲٤-۸) دیده می شود، NetCat بعنوان یک برنامهٔ واحد در دو حالت مشتری (Client Mode) و حالت سرویس دهنده (حالت شنود- Listener) قابل اجراست.

در حالت «مشتری»، NetCat قادر است ارتباطی را با یک پورت TCP (یا UDP) روی ماشین دیگر برقرار کند. داده هائی که باید پس از برقراری ارتباط ارسال شود از طریق ورودی استاندارد (شامل صفحه کلید، فایل یا یک برنامهٔ دیگر) تامین می شود.

در حالت سرویس دهده (که با گزینهٔ /- در هنگام فراخوانی فعال می شود)، NetCat می تواند هر پورت دلخواه TCP یا UDP را بر روی ماشین باز کند، به آن گوش کرده و منتظر دریافت داده شود داده های دریافتی از شبکه مستقیماً به خروجی استاندارد (صفحهٔ نمایش، قابل ، یا هر برنامهٔ دیگر) هدایت می شود.

NetCat از گزینه Source Routing حمایت می کند لذا اگر مسیریابها نیز از این گزینه حمایت کنند می توان از آن برای عملیات Spoofing(ارسال بسته های IP با آدرس جعلی) که در همین فصل تشریح شده استفاده کرد. در حقیقت سادگی و انعطاف پذیری این بازار باعث شده که از آن بتوان در حملات متفاوتی استفاده کرد. به دلیل مشابه می توان برای عملیات مفید و نظارتی نیز از آن بهره گرفت. در ادامه قابلیهای مهم و بنیادین نرم افزار NetCat را تشریح می کنیم. فقط بخاطر داشته باشید که این نرم افزار از طریق خط فرمان اجرا می شود و شامل چندین گزینه (Switch) است. نام نرم افزار به اختصار nc است و قابلیتهای مهم آنرا در چند بخش معرفی می کنیم. (دقت کنید که مثالهایی که ادامه ارائه می شوند به ظاهر در محیط یونیکس اجرا شده اند در حالی که نسخهٔ ویندوز NetCat، تفاوتی با نسخهٔ یونیکس آن ندارد.

مقابله باNetCat

بگونه ای که در طی این چند بخش تشریح شده NetCat نرم افزاری همه منظوره و کاملاً منعطف است و به همان اندازه که در خدمت نفوذگر است می تواند برای مسئول شبکه و همچنین دانشجویان و محققین مفید باشد. لذا وقتی بحث مقابله با NetCat به میان می آید اشاره به روشهای است که بر اساس آنها:

اولاً: راه سوء استفاده از NetCat بسته شود.

ثانیاً: از حملاتی که با استفاده از NetCat پایه ریزی می شود، پیشگیری شود.

لذا راهکارهائی که در ذیل معرف شده چند تازگی ندارد و قبلاً نیز توصیه شده است:

جلوگیری از انتقال فایل از طریق NetCat : باید دیوار آتش یا فیلتر مسیریاب خود را به نحوی پیکربندی کنید که به همان صورتی که تمام ارتباطات TCP از بیرون به درون را نظارت و مسدود می کند، تمام ارتباطات زائد از درون به یرون را نیز مسدود نماید. فقط پورتهائی را باید باز بگذارید که صراحتاً به آنها احتیاج دارید (مثل HTTP 80 , DNS 53 یا هر سرویس دهنده مجاز دیگر)

بعنوان مسئول شبکه باید تک تک پروسه هائی که روی سیستم فعال هستند را بشناسید و دلیل موجهی برای فعال بودن آن داشته باشید. هر گاه یک پروسهٔ ناشناس روی ماشین شما فعال شده و به یک پورت خاص گوش می هد، زنگ خطر به صدا درآمده است. بررسی کنید که چنین پروسه ای چرا و چگونه فعال شده است و به سرعت تکلیف آنرا مشخص کنید.

مقابله با NetCat در هنگام پویش پورتهای باز: هیچ ماشینی نباید پورتهای باز بی هدف داشته باشد. اگر یک ماشین فقط به یک پورت گوش می دهد صراحتاً تمام پورتهای دیگر باید بسته باشند. آتکا به دیوار آتش یا فیلتر برای جلوگیری از ارتباط با پورتهای غیرمجاز چندان هم مطمئن نیست.

مقابله با NetCat در هنگام جستجوی نقاط آسیب پذیر: بدین منظور اولاً خودتان از NetCat برای بررسی نقاط آسیب پذیر سرویس دهنده های خود استفاده کنید و ثانیاً همیشه سیستم خود را با استفاده از آخرین نسخه های patch (اصلاح کنندهٔ نقاط ضعف) به روز نگهدارید.

در مجموع هیچ نسخه واحدی را نمی توان برای مقابله با NetCat تجویز کرد. مسئول شبکه موظف است تدابیر امنیتی لازم را بر روی فیلترها و دیوار آتش پیاده کند تا هیچ ترافیک زائد و مشکوکی مبادله نشود.

حملات DoS (اخلال در سرویس دهی)

بگونه ای که در فصل اول بدان اشاره شد هر حمله ای هدف خاص خود را دنبال می کند و هدف حمله مکانیزم آنرا تعیین می نماید. هدف بسیاری از حملات، ایجاد اخلال و وقفه در سرویس دهی یک ماشین در شبکه است. سودای نفوذگر از چنین حملاتی، وارد کردن ضربات اقتصادی و سیاسی به یک گروه، سازمان، دولت یا شبکهٔ اطلاع رسانی است.[1] هر گاه سیستم مورد حمله به شرکتهای بزرگ یا بناگه های اقتصادی (که عملیات مالی خود را به شبکه محول کرده اند) تعلق داشته باشد، نتایج این حمله ممکن است به یک بحران جدی بینجامد، لذا توجه مسئولین امنیت شبکه را به فراگیری روشهای مقابله با اینگونه حملات جلب می نمائیم.

حملهٔ نوع DOS علیرغم مشابه آن با سیستم عاملی به همین نام، مخفف کلمات Denial of Service یا «اخلال در سرویس دهی» است و بر اساس آن نفوذگر تلاش می کند به یکی از روشهای عملی اعملی، مانع از سرویس دهی یک سرویس دهنده در شبکه بشود.

[1] حملات DOS گاهي اوقات بصورت قانوني و توسط نهادهاي امنيتي و دولتي براي جلوگيري از نشر اكاذيب و موارد خلاف اخلاق و مصالح ملي انجام مي شود چرا كه برخي افراد از گستردگي شبكة اينترنت در سطح جهان سوء استفادة كرده در گوشة مناسبي از اين كره خاكي پناه مي گيرند و برخلاف وجدان عمومي و مصالح ملي اقدام به ايجاد سايتهاي ضد امنيتي يا ضد اخلاقي مي كنند. اين حملات نه تنها مذموم نيستند بلكه توسط افكار عمومي ستايش مي شوند.

این حملات نه از لحاظ تکنیکی زیبا هستند و نه از لحاظ روان شناختی مهیجند؛ بهمین دلیل ملعبهٔ دست بچه های آماتور نیست(!) و علاوه بر آن به دلیل نیاز به خطوط با پهنای باند بالا هزینه زیدی دارند.

نکته امنیتی

بدلی هزینه بالای حملات نوع DOS، اگر شبکهٔ شمابا چنین حمله ای مواجه شد، می توان گفت که به احتمال زیاد یک دشمن قسم خورده در پی نابودی شماست و تکرار چنین حمله ای بسیار محتمل است. بطور معمول برای رهایی از چنین حمله ای باید از نیوی انتظامی یا دستگاههای ذیربط کمک بگیرید زیرا بسادگی نمی توان در مقابل حملات DOS ایستادگی کرد.

هدف حملهٔ DOS صرفاً در هم شکستن یک سرویس دهنده است بگونه ای که سیستم مجبور به راه اندازی مجدد شود یا مدتی از شبکه خارج بماند . حملهٔ DOS می تواند حالت خفته و هدایت شده از قبل، داشته باشد و از درون یک شبکهٔ محلی آغاز شود (Locally DOS) یا آنکه با هدایت مستقیم نفوذ گر از بیرون شروع شود.

(Remotely Dos)

از دیدگاه ، حملات Dos به دو دسته زیر تقسیم می شوند:

- حمله ای که مستقیماً منجر به توقف سرویس های ماشین شود.
- حمله ای که منجر به اشباع یک سرویس دهنده و تلف شدن منابع آن شود.

در حملات Dos از نوع اول، هدف حمله مستقیماً «پروسه های سرویس دهنده» هستند.

این نوع حمله به روشهای زیر امکان پذیر است:

- حمله از درون شبکه از طریق نابود کردن پروسه های در حال اجرا (Process Killing)
- تغییر در پیکربندی که سیستم یا سرویس دهنده (Configuration Modifying)
- در هم شکستن یک پروسه (process Crashing)
- حمله از بیرون با ارسال بسته های ناقص(Malformed Packet Attack)

در مورد این نوع حملات به تفصیل توضیح خواهیم داد.

در حملات DOS از نوع دوم، هدف حمله آنست که منابع سیستمی یک سرویس دهنده مثل حافظهٔ اصلی، حافظهٔ جانبی و امثال آن بگونه ای تلف شود که حتی با وجود فعال بودن سرویس دهنده، کاربران امکان سرویس گرفتن از آنرا نداشته باشند.

اینگونه حملات می تواند به روشهای زیر انجام شود:

- ایجاد متوالی پروسه های فرزند و تکرار فرآیند Fork تا جایی که جدول پروسه ها پر شود. (حمله از درون)
- اشباع سیستم فایل با اطلاعات بیهوده (حمله از دورن)
- جاری کردن سیل بسته به سمت سرویس دهنده (حمله از درون)

روش مقابله با توقف سرویس دهنده ها

به عنوان مسئول شبکه برای پیشیری از حملاتی که از درون تدارک دیده می شود بایستی به موارد زیر دقت داشته باشید:

- هر سیستم عاملی که برای سرویس دهنده های خود در نظر گرفته اید دارای نقاط آسیب پذیر است که متأسفانه پس از عرضهٔ آن به بازار آشکار می شود. بطور معمول شرکت طراح سیستم عامل برای رفع یک شکاف یا اشکال، بلافاصله برنامه های اصلاح کننده آن را ارائه می دهد. لذا همیشه بایستی با شرکت طراح سیستم عامل خود در ارتباط باشید و این نرم افزارها را که بنام Patch عرضه می شوند، جهت رفع نقاط ضعف سیستم بکار بگیرید. بعنوان مثال اشکالات سیستمهای زیر آشکار گردیده و برای اصلاح آنها Patch عرضه شده است: (این شکافها حتی به یک آماتور قدرت ضربه زدن به سیستم را می دهد).
- II-5.0
- Windows XP
- Winsows Swrver 2000
- BIND (Unix DNS Server)
- Unix FTP Server (نسخه های قدیمی)
- سطوح دسترسی به سیستم را برای کاربران مختلف، بدقت تنظیم کنید. هیچ کاربری بجز مسئول شبکه نیازمند ورود به سیستم در عالیترین سطح یعنی Super-

user (یا adm.n) نیست. در اعطای مجوز دسترسی بـه منـابع سیسـتم سـخاوت بـه خرج ندهید و برای هر کاربر حداقل منابعی را که نیار دارد در نظر بگیرید. یک کاربر امین ممکن است با سهل انگاری در حفظ کلمهٔ عبور خود راه را بـرای خطرنـاکترین حملات باز کند.

- بطور منظم و متوالی تنظیمات سیستم را بررسی کرده و آنها را یادداشت نمائید. از نرم افزارهای کشف تغییر در نظیمات سیستم استفاده کنید. (مثل نرم افـزار Tripwire تشریح شده در فص دهم) این نرم افزارها هر گونه دستکار و تغییـر در برنامـه هـای سیستم و همچنین فایلهای پیکربندی را گزارش می دهند.

حملهٔ نوع SYN Flood

در فصل دوم گفتیم که یک ارتباط (اتصال) TCP، طبق قاعدهٔ دست تکـانی سـه مرحلـه ای (3-Way Handshake) شکل می گیرد. اولـین مرحلـه بـرای تقاضـای یـک ارتبـاط، ارسال بسته TCP با تنظیم بیت SYN-1 ، برای ماشین سرویس دهنده می باشد. (ارسـال بستهٔ SYN) به شرطی که در سمت سرویس دهنده پورت مربوطه باز باشد و پروسـه ای به آن پورت گوش بدهد در مرحلهٔ دوم یک بسته با مشخصـات SYN=1 و ACK=1 از سمت سرویس دهنده برای متقاضی ارتباط ارسال می شود. (ارسال بسته SYN-ACK) مسئلهٔ بسیار حیاتی آنست که پروسهٔ TCP در سمت سرویس دهنده، مجبور است پس از دریافت بستهٔ SYN، محتوای فیلد Sewuence Numbwer آنرا (که اختصـاراً ISN مـی نامیم) در جائی ذخیره کند تا آنکه در مرحلهٔ سوم از ان اسـتفاده شـود. پروسـهٔ TCP بـه

ازای دریافت هر بستهٔ SYN یک قطعهٔ کوچک از فضای حافظه خود را بدین منظور اختصاص می دهد و اطلاعات لازم از هر بستهٔ SYN را در آن ذخیره می کند، تا در آینده، هنگام تکمیل مرحلهٔسوم دست تکانی از آن استفاده کند. هر بار سرویس دهنده، یک بستهٔ SYN دریافت کند یک ارتباط «نیمه باز وغیرفعال» (Half open) بوجود می آید که باید جهت تکمیل مرحل و فعال شدن آن در آینده، اطلاعاتی از فیلدهای آن بسته در حافظه ذخیره شود. بدین منظور پروسهٔTCP، از تمام ارتباطات نیمه باز، یک صف (Queue) تشکیل می دهد و اطلاعات لازم از هر ارتباط نیمه باز را درون این صف ذخیره می کند . نقطهٔ ضعف سیستم از همین جا ناشی می شود:

فرض کنید که یک بسته SYN به مقصد برسد و اطلاعات لازم از آن درون صف ذخیره شود ولی مراحل بعدی آن ادامه نیابد. اطلاعات درون صف تا مدت زمان زیادی باقی خواهد ماند تا آنکه زمان سپر شده از حدی بگذرد و آن اطلاعات از صف دور انداخته شود. مدت زمانی که TCP برای تکمیل یک ارتباط نیمه باز صبر می کند بسیار زیاد است. (چیزی بین ٤٥ ثانیه تا ٣٦٠ ثانیه)

در حملهٔ SYN Flood (سیل بسته های SYS)، نفوذگر در یک حلقهٔ بی نهایت بسته های SYN را تولید و با آدرسهای IP دروغین به سمت هدف ارسال می کند. در سرویس دهنده به ازای دریافت هر یک از آنها یک عنصر به صف و یکی به تعداد ارتباطات نیمه باز اضافه می شود. چون هیچگاه این ارتباط نیمه باز TCP تکمیل نخواهد شد لذا صف مربوطه شروع به رشد می کند و تا سپری شدن زمان انقضاء . صف خالی

نخواهد شد. اگر نفوذگر پهنای باند کافی در اختیار داشته باشد تا بتواند تعداد بسیار زیادی بسته SYN را به سمت سرویس دهندهٔ هدف بفرست بگونه ای که قبل از زمان انقضاء و خالی شدن صف فضای مجاز حافظهٔ آن پر شود، در صف جانی برای پذیرش تقاضای برقراری ارتباط TCP باقی نمی ماند و سرویس دهنده عملاً از کار می افتد و این همان هدفی است که نفوذگر دنبال می کرده است! شکل (۲-۹) حملهٔ SYN Fliid را نشان می دهد.

شکل (۲-۹) حمله SYN Flood

اگر چه پس از انقضای زمان، صف تخلیه می شود ولی اولاً این زمان بسیار زیاد است و ثانیاً نفوذگر قادر است این روند را تکرار کند یعنی اگر این عمل در فواصل زمانی مشخص تکرار شود سرویس دهنده بطور دائم مختل خواهد شد.

حملهٔ نوع SYN0Flood یک مشکل دیگر هم بوجود می آورد: با دریافت یک بسته SYN گذشته ذخیره سازی اطلاعات لازم در صف مربوطه، در پاسخ به آن یک بسته بنام SYN-ACK بر می گردد که در حقیقت مرحلهٔ دوم از «دست تکانی سه مرحله ای» محسوب می شود. در نظر بگیرید که نفوذگر هنگام ارسال بسته های SYN، از آدرسهای IP جعلی متعلق به ماشینهایی استفاده کند که در شبکهٔ اینترنت موجود و فعال (Alive) باشند؛ در این حالت سرویس دهندهٔ تحت حمله بسته های SYN-ACK را به سمت آنها

هدایت می کند. چون این ماشینها انتظار دریافت چنین بسته ای را نداشته اند در پاسخ به آن بستهٔ RESET ارسال خواهند کرد. صدمات این حمله در قالب از دست رفـن پهنـای باند ظاهر می شود ولی معمولاً نفوذگران کمتر سعی می کنند که از آدرسـهای IP جعلـی ولی فعال در شبکه استفاده کنند زیرا وقتی آن ماشـین بسـتهٔ RESET پـس مـی فرسـتد کمک خواهد کرد که صف ارتباطات نیمه باز شروع به خالی شدن نمایـد چـون تکلیـف یک ارتباط نیمه باز با دریافت بستهٔ RESET مشخص می شود و اطلاعات مربوط بـه آن از صف حذف می شود. (شکل ۳-۹) مفهوم این نوع حمله را به تصویر کشیده است.

اگر سرویس دهندهٔ هدف، RAM بسیار بالایی در اختیـار داشـته باشـد و صـف پروسـهٔ TCP، بتواند اطلاعات هزاران هزار ارتباط نیمه باز را در خود جا بدهـد هیچگـاه اشـباع نخواهد شد.در این حالت نفوذگر مجبور است که حملهٔ SYN Flood را بـا ایـن هـدف ادامه بدهد تا کل پهنای باند کانال سرویس دهنده، با ترافیک بسته های SYN تلف شود. بنابراین نفوذگر باید خودش پهنای بلتذی بیش از سرویس دهندهٔ هدف در اختیار داشـته باشد. مثلاً اگر سرویس دهندهٔ هدف، یک خـط T1 بـا ظرفیـت 1.544Mbps در اختیـار داشته باشد، نفوذگر به خطی با ظرفیت بیش از خـط T1 نیـار دارد تـا بتوانـد کانـال T1 سرویس دهنده را اشباع کند. تا موقعی که سیل بسته های SYNادامه داشته باشد نفوذگر به هدف خود رسیده است.

مقابلهٔ حدی با حملهٔ Flood SYN

- بهترین و در عین حال پرهزینه ترین راه مقابله با حملهٔ SYN Flood در اختیار داشتن پهنای باند کافی و کانلاهی متعدد برای سرویس دهنده هاست.
- سرویس دهنده ها باید منابع حافظهٔ بسیار زیاد و سخاوتمندانه ای در اختیار داشته باشند.
- اگر یک سرویس دهندهٔ حساس و کلیدی در شبکه وجود دارد که باید دائماً فعال باشد، لازم است دارای یک ماشین پشتیبان در نقطهٔ دیگری از اینترنت باشد. یعنی سایتتان را در دو ISP مجزا سازماندهی کنید که دومی پشتیبان اولی ولی فعلاً پنهان است. با شروع حملهٔ SYN Flood اجازه بدهید این حمله توسط نفوذگر ادامه یابد ولی کاربران را به سایت پشتیبان هدایت کنید. معمولاً وقتی حملهٔ SYN Flood به یک ماشین شروع شد آن حمله تا انتها به همان نقطه ادامه خواهد یافت در حالی که کاربران مجاز که از آدرسهای حوزه بصورت www.siternname.com استفاده می کنند به آدرس IP جدید ارجاع داده می شوند؛ به این عمل « تغییر جهت (Redirection)» گفته می شود. عمل «تغییر جهت» و فراهم کردن شرایط آن هزینهٔ زیادی را بر صاحبان شبکه تحمیل می کند ولیکن برای سازمانهای بزرگ لازم است.
- برای دفاع در مقابل اشباع شدن صف ارتباطات نیمه باز، سازندگان سیستمهای عامل روشهای متفاوتی را در پیش گرفته اند. برخی از آنها اندازهٔ صف مورد استفاده را تا حد نهایت بزرگ در نظر گرفته اند و برخی دیگر زمان انتظار برای تکمیل این

ارتباط TCP را کوتاهتر فرض کرده اند. اگر علاقه مندی که بدانید سیستم عامل شما از چه روشی استفاده می کند و آیا در مقابل حملهٔ SYN Flood آسیب پذیر است یا خیر به آدرس زیر مراجعه کنید:

http://www.nationwide.net/~alephl/FAO

زیباترین روش پیشگیری از اشباع شدن صف روشی است که در سیستم عامل Linux بکار گرفته شده است . در Linux بطور کلی صف ارتباط نیمه باز حذف شده وا صلاً چنین صفی تشکیل نمی شود. پس در Linux اطلاعات لازم از هر ارتباط نیمه باز کجا ذخیره می شود.

در لینوکس اطلاعات لازم که اصلی ترین آن seq No از بستهٔ SYN است به طرز محرمانه در بستهٔ پاسخ یعنی SYN-ACK جاسازی و ارسال می شود. حال اگر ارتباط TCP بطور صحیحی انجام شود در مرحلهٔ سوم مجدداً این اطلاعات باز می گردد؛ ولی اگر حمله ای در کار باشد پاسخ بسته های SYN-ACK باز نخواهد گشت؛ خوشبختانه صفی در کار نیست که اشباع شود! لینوکس این مکانیزم را SYN Cookie نامگذاری کره است. در مکانیزیم SYN Cookie برای تولید و جاسازی فیلد Seq No در بستهٔ SYN-ACK، مبتنی بر مقادیری که فیلدهای زیر در بستهٔ SYN دریافتی دارند یک مقدار چهار بایتی محاسبه می شود:

۱. Source IP Address
۲. Destination IP Address

۳. Sorrce Port Number

٤. Destination Port Number

٥. مقدار فعلی زمان (Time)

٦. یک کلید رمز محرمانه

این مقدار چهار بایتی که بر اساس تابعی از متغیرهای ششگانهٔ فوق محاسبه می شود، به عنوان Seq No پیشنهادی در بستهٔ SYN-ACK قرار گرفته و ارسال می شود و بدین ترتیب ذخیره کردن Seq No بستهٔ SYN در حافظه ضرورتی ندارد. دقت کنید که طبق اصول پروتکل TCP ، فیلد . Seq No بطور طبیعی باید یک مقدار تصادفی داشته باشد ولی در مکانیزیم SYN Cookie، این مقدار غیر تصادفی و حاوی اطلاعات مفید (و محرمانه) است؛ در عین حال هیچ ناسازگاری با عملکرد TCP ندارد.

به شکل (٤-۹) دقت کنید.در این شکل مکانیزم SYN Cookie تبیین شده است:

- در شرایط طبیعی وقتی ماشین Alice یک ارتباط صحیح TCP را شروع می کند بسته ای با مشخصات $SYN(A,ISN_A)$ برای ماشین مقصد ارسال می کند. ISN_A یک عدد تصادفی است که ماشین Alice (A) آن را به عنوان Seq No پیشنهاد می کند.
- ماشین Bob درسمت مقابل از روی مشخصات ششگانه، عدد غیر تصادفی ISN_B (B) آنرا به عنوان Seq No خود پیشنهاد می نماید.

- حل وقتی ماشین Alice در مرحلهٔ سوم با ارسال بستهٔ $ACK(B,ISN_B)$ مراحل ارتباط TCPرا تکمیل می کند مقدار ISN_B را همراه دارد. لذا برای بدست آوردن اطلاعتی که باید در صف ذخیره می شد (یعنی ISN_A) عکس عمل تابع بر روی ISN_B انجام می شد.

وقی که یک حمله پایه ریزی می شود پس از دریافت بستهٔ SYN(X,ISNX) ، پاسخ SYN-ACK آن به سمت یک نقطهٔ نامشخص از شبکه ارسال خواهد شد؛ چون ISN_A در بستهٔ SYN-ACK جاسازی ذخیره شد است در ماشین Alice، صفی که اشباع شود در کار نخواهد بود!

اگر از سیستم عالم لینوکس استفاده می کنید برای فعال کردن این مکانیزم، باید از فرمان زیر استفاده کنید:

```
echo 1> /proc/sys/net/ipv4/tcp_syncookies
```

حملات توزیع شدهٔ Dos (Ddos = Distributed Dos)

حملات Dos، نفوذگر را به موفقی کامل نمی رسانند مگر آنکه شخص نفوذ در خطوط با پهنای باند بالا در اختیار داشته باشد و بالطبع باید برای این کار هزینهٔ بالائی را متقبل شود. در ضمن برای آنکه حملاتی مثل SYN Flood، یک سرویس دهنده را دچار مشکل کند، مدت حمله باید طولانی باشد که می تواند منجر به آشکار شدن هویت نفوذ گر و نهایتاً دستگیری او شود.

«حملات توزیع شدهٔ Dos» که به اختصار آنرا Ddos می نامیم نوعی از حملات هوشمند و زیرکانهٔ Dos است که:

- هزینه ای برای نفوذگر در پی نخواهد داشت.
- هویت نفوذگر پنهان می ماند.
- احتمال موفقیت آن بسیار بالا است.
- می تواند طولانی مدت باشد.
- مبارزه با آن بسیار مشکل است.

این نوع حمله برای اولین بار در تابستان سال ۱۹۹۹ در صحنهٔ اینترنت ظاهر شد و پس از آن بطرز فزاینده ای اینترنت را میدان یک جنگ تمام عیار کرد. بگونه ای که سایتهای eBay،E*Trade، Yanoo، CNN، ZDNet و Amazon.com (که در سطح دنیا شهرت دارند) نیز از این حملات در امان نماندند و ناتوانی و فروپاشی آنها گزارش شد.[1]

در حملات Ddos نفوذگر سعی می کند از ماشینهائی که در سراسر اینترنت پراکنده هستند، برای حمله به یک هدف در شبکه استفاده نمایند. بعنوان مثال اگر هدف حمله بمباران یک سرویس دهنده به روش SYN Flood باشد، نفوذگر به نحوی چندین ماشین پراکنده را برای رسیدن به هدفش بسیج می کند. در اینجا اگر به فرض صد ماشین، هر کدام پهنای باند مفید 15Kbps را برای این منظور صرف کنند. قربانی این

[1] http://www.attrition.org/mirror/attrition

توطئه با سیلی معدل 1.5Mbps مواجه خواهد شد که در این صورت کل پهنای باند خط T1 را اشغال خواهد کرد.

ماشینهای زامبی (Zombie)

به رایانه ای که بدون اطلاع صاحبش وسط یک نفوذگر بـه عنـوان ابـزار حمـلات Dos مورد سوء استفاده قرار می گیرد، «ماشین زامبـی —zombieگفتـه مـی شـود. ماشـینهای زامبی بطور ناخودآگاه در خدمت یک نفوذگر بدخواه قرار می گیرنـد و نفـوذگر آنهـا در حمله به یک هدف در شبکه بسیج، هدایت و فرماندهی می کند.

اولین سؤال آنست که ماشینهای بی گناه به چه نحو در خدمات یک نفوذگر بدخواه قـرار می گیرند؟ معمولاً نرم افزارهای Zombieدر قالب برنامه های رایگان، زیبا و جذاب ولی آلوده، در سراسر شبکهٔ اینترنت توزیع می شوند. بخشـی از کـاربران آمـاتور کـه آگـاهی خاصی از این موضوع ندارند، با اجرای این برنامه ها، ارسال بخشی از بسته های بمباران کننده مثل SYN Flood را بر عهده می گیرند و بدین نحو و اطلاع ، با اهـداف نفـوذگر همسو می شوند !

به هر ترتیب نفوذگر، نرم افزار زامبی را بر روی ماشینهای مختلف توزیع می کند. پس از توزیع گسترده و سرسری این نرم افزارهای آلوده، ممکن است صدها یـا هـزاران ماشـین آلوده بسیج شده و برای شروع یک حمله آماده باشند. بطور معمول نرم افزارهـای زامبـی پس از اجرا بر روی یک ماشین منتظر صدور فرمان می مانند. وقتی تعدا ماشینهای زامبی زیادباشد، نفوذگر قدر به هدایت و فرماندهی همـهٔ آنهـا نیسـت لـذا در حمـلات Dos ،

ماشینهای زامبی در قالب تعدادی «گروه» دسته بندی می شوند. هر گروه از ماشینهای زامبی توسط یک ماشین «سرگروه» فرماندهی می شوند و ماشینهای سرگروه تحت فرمان ماشین نفوذگر هستند.

در ادامه یک نوع حملهٔ بسیار مشهور و خطرناک Ddos را برای آگاهی مسئولین شبکه معرفی می کنیم و سپس به روشهای مقابله با آن خواهیم پرداخت.

حملهٔ Ddos از نوع TFN2K(Tribe Floos Network 2000)

به شکل (٦-٩) دقت کنید. این شکل مکانیزم حملهٔ TFN2K را (که توسط گروه Mixter طراحی شده) نشان می دهد.

در TFN2K نفوذگر از نرم افزار NetCat برای ارتباط با سرگروه ها استفاده می کند. هر گروه یک مجوعه از ماشینهای زامبی را رهبری می کند و مستقیماً تحت فرمان نفوذگر است. (ماشینهای سرگره اصطلاحاً CLENT نامیده می شوند) نفوذگر از طریق ابزار NetCat، فرمان حمله را صادر می کند و آنها نیز به ماشینهای زامبی تحت فرمان خود، دستور شروع حملهٔ Ddos را بر علیه هدف مشخص در شبکه ابلاغ می نمایند.

ماشین های سرگروه (یا CLIENTS) نیز ماشینهای بی گناهی هستند که ناخودآگاه در اختیار نفوذگر از هر نقطه نامعلوم در شبکه قادر به صدور فرمان حمله خواهد بود.

در TFN2K ماشینهای زامبی می توانند از مکانیزمهای زیر بر علیه ماشین هدف استفاده کنند:

- UDP Flood: ارسال سیل آسای بسته های UDP به سمت یک قربانی در شبکه به نحوی که تمام پهنای باند در اختیار او، با این حجم از بسته های UDP تلف شود.
- SYN Foold: ارسال سیل آسای بسته های SYN که قبلاً بطور مفصل توضیح داده شد.
- ICMP Flood : ارسال سیل آسای بسته های Ring (ICMP Echo Reques)
- حملهٔ Smurf: ارسال فراگیر بسته بسته های Ping یا UDP

- حملهٔ Mix: شامل ارسال سیل آسا و همزما بسته های SYN، UDP و ICMP به سمت یک هدف
- حملهٔ Targa: این حمله شامل ارسال سیل آسای بسته های ناقص IP به سمت یک هدف است که توسط گروه Mixter (طراح اصلی TFN2K) ابداع شده است.

کدهای برنامهٔ TFN2K بهمراه اطلاعات تفصیلی؟، در سایت http://mixter/warrior2k.com. در دسترس است. کدهای این برنامه در دیسک ضمیمهٔ کتاب موجود می باشد.

با توجه بدانکه در TFN2K از شش مکانیزم حملهٔ Dos استفاده می شود لذا نفوذگر قادر است که حمله ای مثل ICMP Foold را شروع کند و سپس بررسی نماید که آیا موفق بوده است یا خیر. اگر حمله موفق نباسد به ماشینهای تحت فرمان خود دستور تغییر نوع حمله (مثلاً به SYN Foold) را صادر می کند.

یکی از شگفت ترین نکات بکار رفته در TFN2K آنست که در ماشینهای آلوده به این نرم افزار، هیچ پورت ، یا UDP باز نمی شود و بنابراین با نرم افزارهای پویش پورت (Port Scan) و یا اجرای فرمان netstat-na، نمی توان متوجه شد که پورت خاص و مشکوکی بر روی ماشین باز شده است و بدین نحو بسیار مخفی عمل می کند.

سؤال اینجاست که فرمان حمله چگونه صادر می شود؟ در TFN2K برای فرمان دادن به ماشینهای زامبی از بسته های Echo Reply packer استفاده می شود.[1] برای آنکه فقط

[1] تاکید می کنیم که از بستهٔ Echo Reply یعنی بستهٔ پاسخ جدول استفاده می شود نه Enho Request :

شخص نفوذگر بتواند فرمان حمله را صادر کند و شخص دیگری نتواند فرماندهی حمله را بست بگیرد، در هنگام ارسال بستهٔ Echo Reply Packer در قسمت حمل داده[1] یک کلمهٔ عبور رمز شده قرار می گیرد.

دلیل استفاده از بستهٔ Echo Reply آنست که اکثر مسیریابها یا دیوارهای آتش اجازه می دهند چنین بسته ای به درون شبکه وارد شود زیرا فرض می کنند این بسته در پاسخ به دستور ping(Echo Request) باز گشته است.

نکات بسیار خطرناک TFN2K عبارتند از:

- عدم باز کدرن پورت باز روی ماشین زامبی و مخفی ماندن از چشم نرم افزارهای PortScan
- استفاده از بسته های ICMP Echo Reply جهت عبور از دیوار آتش
- آدرس مبداء صدور بسته ICMP Echo Reply به طور اشتباه و جعلی تنظیم می شود تا ماشین سرگروه و ماشین نفوذگر کشف نشوند.
- ماشینهای زامبی نیز در هنگام حمله به یک هدف از آدرسهای مبداء اشتباه و جعلی استفاده می کنند تا دیرتر کشف شوند.
- پیگیری محل ماشینهای زامبی سودی ندارد زیرا اولاً در سطح دنیا (یا یک کشور) پراکنده اند و به ISPهای متفاوت و مسیریابهای متفاوتی متصل هستند. ثانیاً بی گناه هستند!! ثالثاً بسیار زیادند و نمی توان تمام آنها را غیرفعال کرد.

---

[1] Payload

- پیگری ماشین فرماندهی حمله و جستجوی نفوذگر بسیار مشکل خواهد بود و نفوذگر وقت کافی دارد تا بعد از صدور فرمان حمله از شبکه خارج شود.
- در TFN2K هر ماشین سرگروه یک فایل مخفی از آدرسهای IP متعلق به ماشینهای زامبی تحت فرمان خود در اختیار دارد و به منظور آنکه در صورت آشکار شدن هویت سرگروه ماشینهای زامبی لو نروند، این فایل بصورت رمز شده بر روی ماشینهای سر گروه ذخیره می شود.
- در TFN2K نفوذگر می تواند نسخهٔ نرم افزار نصل شده بر روی ماشینهای زامبی را به روز کند
- نفوذگر قادر است ماشینهای زامبی را وادار به پاک کردن نرم افزار TFN2K از دیسک سخت کنند (عمل خودکشی)، یا آنکه در لحظات خاتمه ماموریت کل اطلاعات دیسک سخت ماشین زامبی را پاک نماید (انفجار قرارگاه).

اگر درست بیندیشید حملهٔ TFN2K به مثابهٔ یک جنگ واقعی در میدان نبرد عمل می کند!!!

بغیر از TFN2K نرم افزارهای زیر نیز بعنوان ابزار حملات Ddos مطرح هستند:

- Bilitznet,by Phreeon
- Mstream
- Trin00
- Win Trin00

- Freak88
- Trinity
- Shaft
- Stacheldraht,by Randomizer(German Goroup)

مقابله با حملات Ddos

هر چند مقابله با حملات Ddos دشوار به نظر می رسد ولی لازم است مسئولین شبکه سیاستهای امنیتی زیر را به اجرا بگذارند:

چون هیچ مسئول شبکه ای تمایل ندارد ماشینهای شبکهٔ تحت مدیریت او در اختیار یک بدخواه قرار بگیرد لذا بایستی بطور مداوم ماشینهای شبکه آزمایش شوند.

مسئول شبکه باید در جریان آخرین اخبار و گزارشها در مورد انتشار نرم افزارهای زامبی Zombio قرار بگیرد چون ممکن است گروهی قبل از عملیاتی شدن یک نرم افزار زامبی آنرا کشف کند.

از نرم افزارهای Anti-Spoof دو طرف بر روی مسیریابهای خود بهره بگیرید. این نرم افزارها اجازهٔ عبور بسته های IP با آدرسهای جعلی را نمی دهند. بدین ترتب هیچ بسته ای حق ندارد از درون یک شبکه که NetID آن مشخص است با آدرس جعلی بیرون برود.

اگر شک دارید که آیا ماشینهای شبکهٔ شما آلوده به نرم افزارهای زامبی هستند می توانید با سایتهائی که بطور رایگان خدمات مشاوره ای ارائه می دهند و ابزارهای مناسبی در اختیار دارند تماس بگیرید. یکی از این سایتهای بسیار ارزشمند و مفید، در آدرس زیر در دسترس می باشد:

http://www.nipc.gov.

این سایت که توسط دولت آمریکا راه اندازی شده آخـرین گزارشـهای خبـری در مـورد فعالیتهای نفوذگران و هشدارهای امنیتی را به رایگان در اختیار علاقمندان قرار می دهـد. در شکل (۷-۹) صفحهٔ اصلی سایت وب NIPC [1] نشان داده شده است.

استفاده از نرم افزارهای Zombie Zapper

توصیه می شود:

http://razor.bindview.com/tools/ZombieZapper form.shtml

اگر سایت حساسـی داریـد بایـد در نقطـهٔ دیگـری از شـبکهٔ اینترنـت و روی یـک ISP متفاوت ، پشتیبان داشته باشد تا به محض کشف حملهٔ Ddos تغییر سایت بدهید.

[1] National Infrastructure Protection Center (nipc)