سیستمهای کشف مزاحمت (IDS)

سیستم کشف مزاحمت که به اختصار IDS نامیده می شود ، برنامه ایست که با تحلیل ترافیک جاری شبکه یا تحلیل تقاضاها سعی در شناسائی فعالیتهای نفوذگر می نماید و در صورتی که تشخیص داد ترافیک ورودی به یک شبکه یا ماشین از طرف کاربران مجاز و عادی نیست بلکه از فعالیتهای 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 عملکرد IDS در سطوح مختلف، روشهای فرار نفوذگر از آنرا نیز بررسی خواهیم کرد.سیستم IDS در دو سطح ((لایه شبکه)) و ((لایه کاربرد)) عمل می کند و مکانیزم هر یک با دیگری متفاوت است.

عملکرد سیستم IDS مبتنی بر لایه شبکه

در این نوع سیستم کشف مزاحمت،IDS تمام بسته های IP وارده به شبکه 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 بسته های IP (و محتویات آنها شامل بسته های TCPوUDP)را مرتب کرده و بروز واقعی یک حمله را تشخیص بدهد.

بطور معمول سیستمهای IDS یک بانک اطلاعاتی از الگوی حملات مختلف در اختیار دارند.

(به این بانک اطلاعاتی، بانک ویژگیها و امضای حمله Signatures& Features گفته می شود) در حقیقت اکثر سیستمهای IDS تحلیلهای خود را بر Attack تطابق الگوهای حمله با ترافیک موجود در شبکه متمرکز کرده اند و هرگاه الگوی ترافیک جاری در شبکه با ویژگی یکی از حملات منطبق باشد یک حمله گزارش خواهد شد.لذا نفوذگر برای فرار از IDS سعی می کند به روشهای مختلف 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 جو(Pager) یا به صدا درآوردن بوق آژیرآنرا به اطلاع مسئول شبکه می رساند و در عین حال به تعقیب حمله ادامه می دهد.شکل (۱۹-٦) یک سیستم IDS معمولی (در سطح شبکه) را نشان می دهد.

در این شکل سیستم IDS در حین نظارت برترافیک شبکه متوجه تلاش برای ارتباط با پورتهای ۸۰و۲۳شده است.این سیستم تلاش برای برقرای ارتباط با پورت ۲۳(مربوط به TelNet) را اصلاً طبیعی نمی داند و آنرا به عنوان علائم یک حمله گزارش می کند.یا مثلاً سیستم IDS با تحلیل جریان بسته های IP متوجه می شود که چند هزار

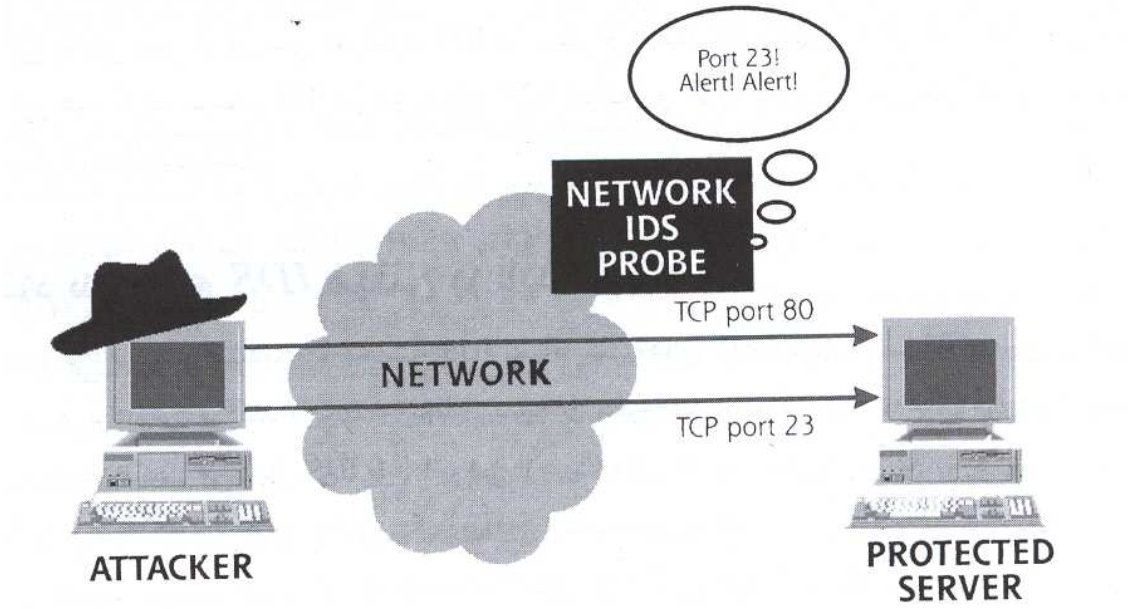
بسته SYN با فیلد Source IP یکسان و با شماره های مختلف پورت به شبکه ارسال شده است. این مسئله قطعاً علامت بروز یک حمله است.

حال باید دید نفوذگر به چه نحوی تلاش می کند از IDS مبتنی بر لایه شبکه فرار کند؟

نفوذگر از مکانیزمهای زیر برای فرار از IDS (IDS Evasion) بهره می گیرد:

ترافیک ارسالی به شبکه هدف بگونه ای تنظیم می شود که با الگوی 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 نهائی (Host) مشخص شود.



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 بسته IP را به همراه دارد.قطعات کوچکتر (Fragment) شکسته شود.هر بسته شکسته شده سرآیند بسته IP را به همراه دارد.قطعات مختلف از طریق شبکه ارسال شده و نهایتاً در ماشین مقصد بازسازی خواهند شد.وقتی سیستم IDS با بسته های قطعه قطعه شده IP مواجه می شود باید همانند ماشین نهائی آنها را دریافت و بازسازی نماید.نفوذگر می تواند بسته های 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 شده 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 ۲۰۰۰تقریباً 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 بسته TCP یا UDP) از سیستم IDS فرار می کرد. بعنوان مثال ابزار Snort(که یک نرم افزار Open Source و رایگان است) بعنوان یک سیستم IDS بسیار معروف تا سال ۲۰۰۰ در مقابله با بسته های قطعه قطعه شده ناتوان بود!

در ضمن نفوذگر می تواند قطعه قطعه کردن بسته IP را به روش های نامتعارف انجام بدهد بگونه ای که سیستم IDS نتواند بدرستی آنرا بازسازی کند. مکانیزم این نوع حمله به شرح زیر است:

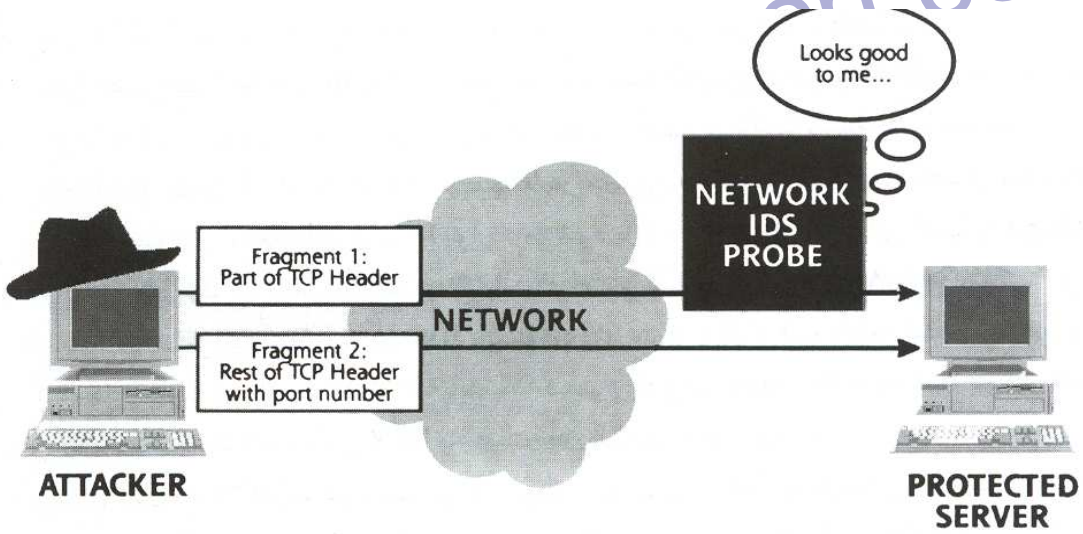
حمله به IDS بر اساس قطعات کوچک و قطعات هم پوشان IP

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

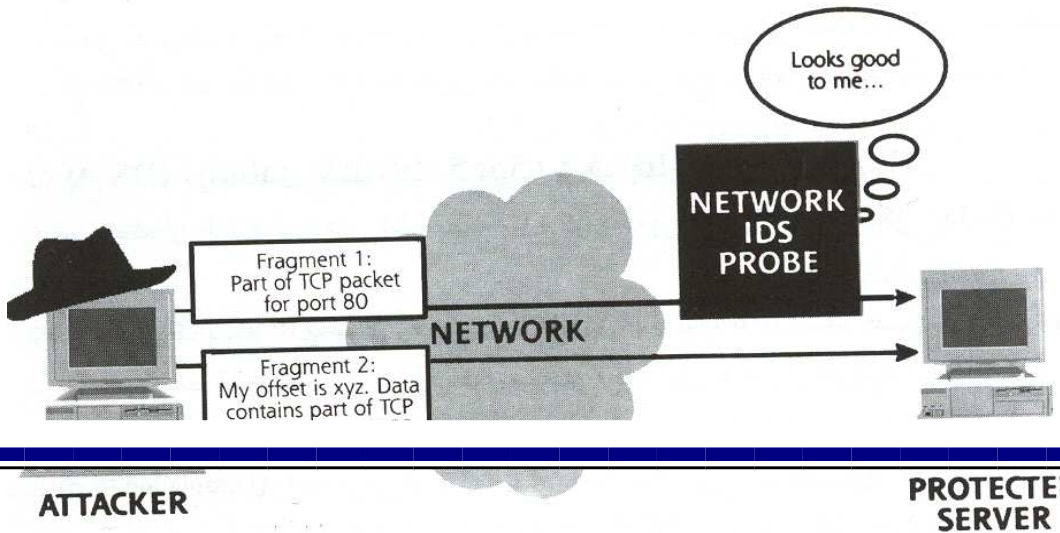
فرض کنید یک بسته IP محتوی یک بسته TCP(در فیلد Payload) باشد.چون بخش Payload از هر بسته IP می تواند قطعه قطعه شود لذا بطور عمدی قطعه اول به قدری کوچک در نظر گرفته می شود که فقط دو بایت اول از بسته TCP را شامل شود و بنابراین دو بایت دوم از بسته TCP که شماره پورت مقصد (Destination Port) را در برمی گیرد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

TCP و شماره پورت مقصد را ندارد معمولاً IDS آنرا معمولی درنظر گرفته و از آن می گذرد.بدینصورت نفوذگر IDS را دور می زند.نوع دیگر حمله به IDS حمله براساس قطعات همپوشان (Fragment Overlap) است که با دستکاری و تغییرات عمدی در فیلد Fragment Offset (از بسته IP) انجام می شود.بگونه ای که در محل قرار گرفتن قطعه جاری را در دیتاگرام اصلی مشخص می کند.به شکل (۲۱-۶) دقت کنید.قطعات همپوشان با مکانیزم زیر تنظیم و ارسال می شوند:

اولین قطعه بسته IP که شامل سرآیند بسته TCP است دارای شماره پورت مجاز است.(مثل HTTP-TCP80).



شکل (۲۰-۶) حمله به IDS مبتنی بر مکانیزم قطعات کوچک



دومین قطعه بگونه ای تنظیم می شود که پس از بازسازی بر روی بخشی از قطعه های قبل نوشته شده و مقادیر قبلی را بازنویسی کند لذا شماره پورت واقعی در قطعه دوم مشخص می شود.شماره پورتی که در قطعه اول درج شده است و بعداً بازنویسی خواهد شد.

چون احتمالاً فقط قطعه اول از هر بسته IP توسط IDS بررسی می شود لذا قطعه دوم که قطعه اول را بازنویسی می کند توسط IDS تشخیص داده نخواهد شد!

قطعات همپوشان پس از بازسازی در ماشین هدف بسته TCP اصلی را با شماره پورت واقعی تشکیل می دهد!

FragRouter:ابزاری برای فرار از چنگ سیستم IDS

ابزار FragRouter که توسط گروه Dug Song طراحی شده انواع حملات به سیستم IDS را بر اساس تکنیک قطعه قطعه سازی بسته های IP پیاده سازی کرده

است.این ابزار در آدرس http://www.anzen.com/research/nidsbench/ در دسترس قرار گرفته و در محیطهای Solaris-Linux و BSDقابل اجرا است.(این برنامه و کدهای آن (به زبان C)در CD جانبی کتاب ضمیمه شده است.)

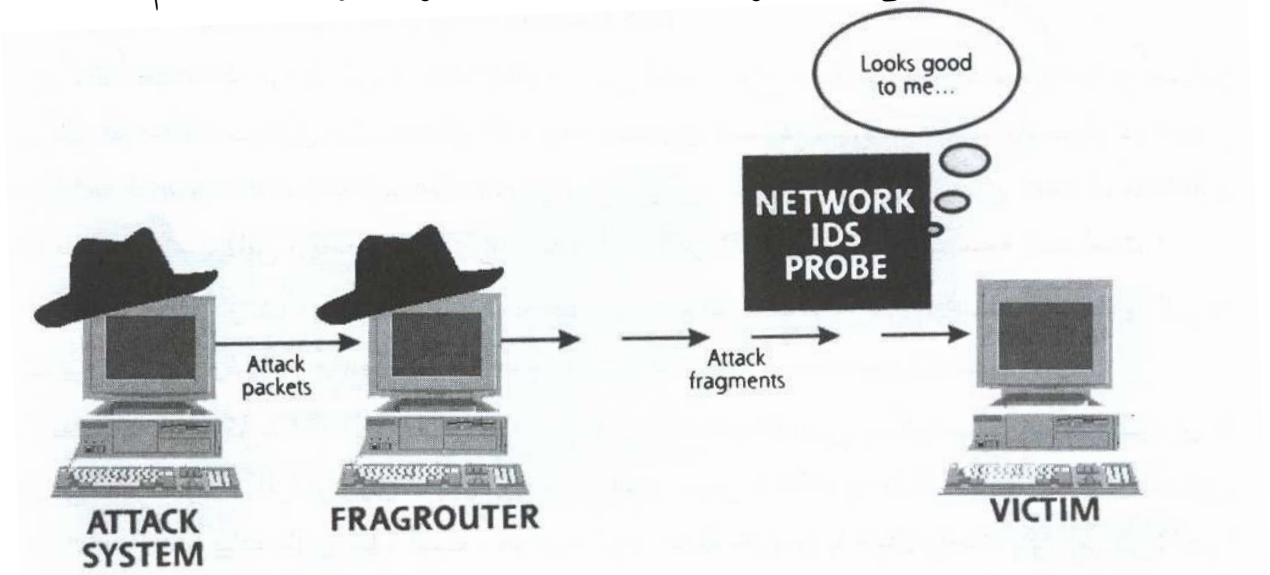
FragRouter بیش از ۳۵ روش مختلف را در قطعه قطعه سازی بسته های **IP**(یا TCP)پیاده سازی کرده است.برخی این روشها در جدول (۲-٦) معرفی شده اند.

بگونه ای که در شکل (۲۲-٦) دیده می شود نرم افزار FragRouter نقش یک مسیریاب نرم افزاری را بازی می کند.نفوذگر آنرا بر روی ماشینی نصب کرده و سپس با اتکاء به مکانیزمهای قطعه قطعه سازی که در FragRouter پیاده سازی شده است هر گونه ابزار حمله خود را با خیال راحت بکار می گیرد چرا که این مسیریاب پس از دریافت بسته های حمله آنرا به نحو مناسب و دلخواه نفوذگر تکه تکه خواهد کرد بگونه ای که حتی الا مکان سیستمIDS را تحریک ننماید!

مکانیزم قطعه قطعه سازی بسته های IP	نام Flag برای تنظیم	نام مکانیزم قطعه قطعه سازی
بسته های IP را به قطعات ۸ بایتی تقسیم کرده و سپس ارسال می نماید .	-F1	Frag-1
بسته های IP را به قطعات ۲٤ بایتی تقسیم کرده و سپس ارسال می نماید .	-F2	Frag-2
بسته های IP را به قطعات ۸ بایتی تقسیم کرده و سپس ارسال می نماید ؛ با این ویژگی که یکی از بسته ها خارج از ترتیب ارسال می شود .	-F3	Frag-3
ابتدا مراحل " دست تکانی سه مرحله ای " را تکمیل کرده و یک ارتباط TCP برقرار می کند . سپس بسته های جعلی RST و FIN با Checksum اشتباه ارسال می نماید . (بسته ای که کد کشف خطای غلطی در آن تنظیم شده است .) سپس داده ها را بصورت بسته های یک بایتی ارسال می کند .	-T1	Tcp-1
ابتدا مراحل " دست تکانی سه مرحله ای " را تکمیل کرده و یک ارتباط TCP برقرار می کند . سپس داده ها را بصورت بسته های دو بایتی ارسال می کند . بین هر دو بسته دو بایتی ، بسته ای یک بایتی ارسال می شود که پس از بازسازی بایت دوم از بسته قبل را بازنویسی می کند . بدین ترتیب از هر بسته دوبایتی ، بایت دوم توسط بعدی تغییر خواهد کرد . (Overlapping Segment)	-T5	Tcp-5

		.
Tcp-7	-T7	ابتدا مراحل " دست تکانی سه مرحله ای " را تکمیل کرده و یک ارتباط TCP برقرار می کند . سپس داده ها را بصورت بسته های یک بایتی ارسال می کند . بین هر دو بسته یک بایتی ، بسته ای پوچ ارسال می شود که دارای Sequence Number کاملاً غلطی است .

حدول (۲-٦) برخی از مکانیزمهای FragRouter برای گول زدن سیستم IDS



شکل (۲۲-٦) فرار از سیستم IDS توسط ابزار FragRouter

بگونه ای که در شکل (۲۲-٦)دیده می شود نرم افزار FragRouter نقش یک مسیریاب نرم افزاری را بازی می کند.نفوذگر آنرا بر روی ماشینی نصب کرده و سپس

با اتکاء به مکانیزمهای قطعه قطعه سازی که در FragRouter پیاده سازی شده است هرگونه ابزار حمله خود را با خیال راحت بکار می گیرد چرا که این مسیریاب پس از دریافت بسته های حمله آنرا به نحو مناسب و دلخواه نفوذگر تکه تکه خواهد کرد بگونه ای که حتی الامکان سیستم IDS را تحریک ننماید!

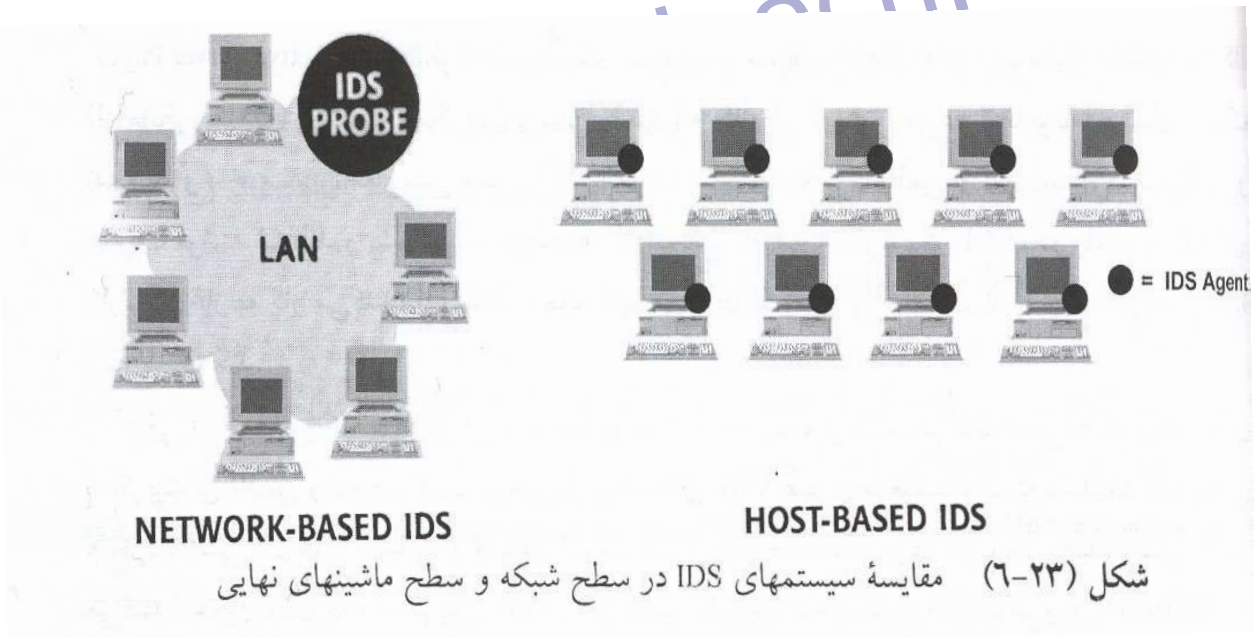
بنابراین نرم افزار FragRouter ابزاری برای فرار از سیستم IDS محسوب می شود و هیچ کاربرد دیگری ندارد.مستقل بودن آن(در قالب یک مسیریاب نرم افزاری)به نفوذگر اجازه می دهد تا بدون واهمه از بصدا درآمدن آژیرهای خطر توسط IDS از ابزارهای موردنظر خود استفاده کند.یعنی نفوذگر می تواند پس از نصب و فعال کردن آن ابزارهای نقشه برداری از شبکه(مثل Cheops)ابزارهای پویش پورتهای باز(مثل Nmap)ابزارهای کشف قواعد دیوار آتش(مثل Firewalk)و ابزارهای کشف نقاط ضعف سیستم(مثل Nessus) را بکار بگیرد.تمام بسته های تولید شده توسط این ابزارها قبل از خروج از شبکه به FragRouter تحویل داده می شوند تا عملیات لازم بر روی آن انجام شود.

عملکر سیستم IDS در سطح کاربرد و روشهای فرار از آن

سیستم کشف مزاحمت در لایه کاربرد تمام تقاضاهای ارسالی هر برنامه کاربردی را بررسی کرده و در صورت تقاضاهای نامتعارف و خطرناک هشدار می دهد.(بدین ترتیب هر پروتکل لایه کابرد سیستم IDS مستقلی نیاز دارد.به شکل ۲۳-۶ دقت

کنید.)بعنوان مثال تقاضاهای HTTP می تواند خطرناک باشند.فرض کنید یک تقاضای HTTP از نوع GET با قالب زیر صادر شده باشد:

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



شکل (۲۳-۶) مقایسهٔ سیستمهای IDS در سطح شبکه و سطح ماشینهای نهایی

این تقاضا می تواند باعث شود تا یک برنامه آسیب پذیر از نوع CGI بر روی سرویس دهنده هدف اجرا شود.IDS می تواند چنین تقاضاهائی را کشف کرده گزارش بدهد.IDS در سطح کابرد قواعدی را جهت بررسی تقاضاها در اختیار دارد و براساس آن تقاضاهای خطرناک را تشخیص می دهد. نفوذگر برای فرار از IDS

روشهای متنوعی را بکار می گیرد که دانستن این روشها برای مسئولین شبکه بسیار حیاتی است چرا که حمله در سطح لایه کابرد بسیار مخرب است.یکی از این حملات در لایه کاربرد حمله به برنامه های CGI است که در ادامه آنرا بررسی کرده و روشهای فرار نفوذگر زا IDS متناظر با آن را توضیح می دهیم.

حمله بر علیه IDS جهت کشف برنامه های CGL نوشته می شوند.اسکریپت های CGL کدهای اجرائی خاصی هستند که بر روی سرویس دهنده وب اجرا می شوند.بعنوان مثال یک اسکریپت CGL می تواند داده های ارسالی یک کاربر را دریافت کرده و در یک بانک اطلاعاتی ذخیره کند.

یک کاربر اطلاعات لازم را از طریق مرورگر خود(Browser) در صفحه وب مربوطه وارد می کند.(محل دریافت داده از کاربر در صفحه وب فرم (Form)نام دارد.)وقتی داده های هر فرم برای سرویس دهنده وب ارسال می شود سرویس دهنده اسکریپت مربوطه را اجرا کرده و داده های ارسالی را برای آن می فرستد.اسکریپت CGL پس از پردازشهای لازم بر روی داده های فرم پاسخ مناسب را به کاربر باز می گرداند.

اگر کمی با تامل به عملکرد CGL های معمولی نگاه کنیم اکثر آنها عملیات زیر را انجام می دهند:

جستجو در یک بانک اطلاعاتی به دنبال یک آیتم خاص و مورد نظر کاربر

ذخیره اطلاعات ارسالی توسط کاربر در یک بانک اطاعاتی

انجام یک محاسبه فوری و بلادرنگ (Online)

اکثر برنامه های مبتنی بروب براساس اسکریپتهای CGL یا روشهای مشابه به مثل ASP (Active Server Page) یا Perl نوشته می شوند.بسیاری از سرویس دهنده های وب مثل Apache یاIIS (از مایکروسافت) روشهای اسکریپت نویسی را با ارائه مثالهائی آموزش می دهند تا برنامه نویسان از یک نقطه شروع خوب برنامه نویسی وب را آغاز نمایند.متاسفانه بخش اعظمی از این اسکریپتها ناقص و ضعیف نوشته شده اند و بشدت آسیب پذیرند. از آنجائی که این اسکریپتهای CGL در پاسخ به تقاضای کاربران آغاز به کار می کنند لذا نقاط ضعف آنها می تواند نفوذگر را برای حمله به یک سرویس دهنده وب کمک کند.

یکی از عملیاتی که نفوذگر در مرحله پویش و جستجوی سیستم انجام می دهد پیدا کردن اسکریپتهای ناقص و ضعیف است. برخی از برنامه های CGL بقدری ضعیف نوشته شده اند که بدون هیچ پردازشی داده های ارسالی از صفحه وب مشتری را مستقیما روی سرویس دهنده اصلی اجرا می کنند.بعنوان مثال یک اسکریپت CGL را درنظر بگیرید که یک فرمان پرس و جو (Query) را از صفحه وب مشتری گرفته و آنرا مستقیما به SQL Server می فرستد.اگر چه کاربران بطور معمول آن اسکریپت را از طریق صفحه وب راه اندازی می کنند ولی نفوذگر نه از طریق صفحه وب بلکه از

طریق برنامه نویسی با پورت ۸۰ از سرویس دهنده ارتباط برقرار می کند و تقاضای راه اندازی یک اسکریپت را با متود GET بصورت مصنوعی تولید و به سرویس دهنده می فرستد.اسکریپتهای CGL ضعیف هرچه که ارسال شده باشد(با فرض آنکه از صفحه وب و از طریق فرم آمده) اجرا می کنند! بنابراین یکی از اهداف نفوذگر یافتن همین نقاط ضعف در اسکریپتهاست.برای جستجوی اسکریپتهای آسیب پذیر ابزارهای متنوعی وجود دارد که به نظر می رسد قویترین آنها Whisker است. زیرا بغیر از جستجوی آسیب پذیری سعی می کند سیستم IDS را نیز گول بزند.Whisker که در سایت وب Rainforest Puppys با آدرس http://www.wiretrip.net/rfp/ در دسترس است مبتنی بر Perl بوده و روی تمام محیطهائی که زبان Perl را پشتیبانی می کنند قابل اجرا است.(اگرچه در نرم افزار Nessus که قبلا معرفی شد برای بررسی نقاط ضعف اسکریپتهای CGL امکاناتی در نظر گرفته شده ولی Nessus در سطح TCP/IP قدرتمند است در حالیکه Whisker در سطح لایه کاربرد و وب)

Whisker نقاط ضعف بیش از ۵۰۰ نوع اسکریپتCGI و ASPرا می شناسد و آنها را آزمایش می کند.در ضمن قادر است با سرویس دهنده های وب مجازی (Virtual Webserver)که بر روی یک ماشین واحد اجرا می شوند ارتباط برقرار کند.در ضمن روشی برای حدس User ID و کلمه عبور که در هنگام احراز هویت (Web Authentication) درخواست می شود پیاده سازی کرده است.

بزرگترین ویژگی Whisker آنست که برای پویش و جستجوی نقاط ضعف اسکریپتهای CGL از روشهای خاصی استفاده می کند تا سیستم IDS وب(Web IDS) را گول بزند.بسیاری از سیستمهای IDS در سطح لایه کاربرد تمام تقاضاهای GET و PUT و POST و DELETE را بازسازی کرده و در هنگام دریافت یک تقاضا که منجر به اجرا شدن یک اسکریپت CGL می شود اعلام خطر می نمایند.بگونه ای که اشاره شد وقتی تقاضای زیر مبتنی بر متود GET توسط سیستم IDS دریافت می شود IDS کشف می کند که کسی سعی در اجرای اسکریپت broken.cgi از شاخه /cgi-bin/ را دارد:

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

بدین ترتیب زنگ خطر را به صدا درآورده و مانع از اجرای اسکریپت مربوطه خواهد شد.

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:(درتمام این روشها منظور از تقاضا ارسال یک فرمان HTTP به سرویس دهنده وب برای فعل و انفعال با برنامه CGL تلقی شده است.)

URL Encoding:قسمت آدرس در URL ارسالی با کدهای معمولی ASCII ارسال نمی شود بلکه ابتدا هر کاراکتر با معادل یونی 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%67%69%2d%62%69%6e/broken.cgi HTTP/1.0

نکته:طبق استانداردMIME برای معادلسازی کاراکترها در رشته URL ابتدا کاراکتر ٪ و سپس کد هگزادسیمال عدد قرار می گیرد!

%63-C / %67-g / %69-i / %2d- - / %62-b / %69-i / %6e-n

%63%67%69%2d%62%69%6e - cgi-bin

Directory Insertion ./: URL ارسالی شامل کاراکترهای ، است که در برخی از سرویس دهنده های وب به این شکل تعبیر و تفسیر می شود:"لطفا به شاخه جاری تغییر مسیر بدهید!" تغییر مسیر به شاخه جاری هیچ خاصیت یا ضروری 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 نمی 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 / broken . cgi / HTTP / 1.0

آنهایی که با پروتکل HTTP آشنا هستند صحت 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 معنی در سرویس دهنده 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 /thisisabunchofiunktomaketheURLlonger / .. / cgi-bin / broken. Cgi HTTP / 1.0

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

GET / index . htm1?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 مجاز بداند.(برخی از سیستمهای IDS بدین نحو گمراه شده و به آن تقاضا اجازه اجرا می دهند 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 انگلیسی 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 اجرا است. بدین ترتیب سیستم IDS فریب می خورد و تقاضای ارسالی اجرا می شود. مثال:

GET / CGI-BIN / broken . cgi HTTP / 1.0

Windows Delimiter: در سیستم عامل ویندوز استفاده از علامت \ بجای / (جدا کننده شاخه) مجاز شمرده می شود در حالیکه برخی از سیستمهای IDS به آن حساسیت ندارند لذا در مورد شکل URL گمراه می شوند. مثال:

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

Session Splicing: بر خلاف نِه روش قبلی مکانیزم Session Splicing در سطح لایه TCP پیاده سازی شده است. تقاضا ابتدا تکه تکه شده و در بسته های جداگانه یک تا سه کاراکتری ارسال می شود. بسیاری از سیستمهای IDS توانایی

بازسازی قطعات URL را ندارد و لذا اجبارا 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 ازسیستمهای IDS برای تحلیل رشتهURL ازتوابع رشته ای استفاده می کنند. حال اگر در بین رشته URL کاراکتر 00% (NULL Character) وجود داشته باشد توابع رشته ای آنرا بعنوان خاتمه رشته تلقی می کنند و بدین نحو IDS گمراه می شود در حالی که URL اعتبار خود را از دست نخواهد داد.(اکثر سرویس دهنده های وب از کاراکتر 00% چشمپوشی می کنند.) مثال:

GET %00 / cgi-bin / broken . cgi HTTP / 1.0

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

مکانیزمهای Whisker:

در Whisker برای گول زدن و فرار از IDS سعی شده است تا رشته ارسالی در قالب متود GET (که منجر به فراخوانی اسکریپت CGL 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.

مقابله با نفوذگرانی که قصد گول زدن سیستم IDS را دارند

برای بالا بردن امنیت شبکه توصیه می کنیم که بر روی شبکه خود هر دو نوع سیستم IDS را نصب نمائید یعنی :

IDS مبتنی بر لایه شبکه (Network-Based IDS)

IDS مبتنی برلایه کاربرد(قابل نصب برروی ماشینهای نهایی وسرویس دهنده ها)

IDS نوع اول سعی می کند که بطور عموم ترافیک شبکه را نظارت کند و با مقایسه ترافیک جاری شبکه با الگوی حملات مختلف بروز یک حمله را گزارش بدهد.

IDS نوع دوم با بررسی تقاضاهای رسیده به هر سرویس دهنده تقاضاهای خطرناک را تشخیص داده و اعلام می نماید. لذا مجبور هستید IDS نوع دوم را روی ماشین سرویس دهنده نصب کنید یعنی IDS نوع اول عمومی است و IDS نوع دوم بطور خاص یک سرویس دهنده را نظارت می کند.

نیاز به نصب هر دو IDS از آنجا ناشی می شود که نفوذگر برای پویش و جستجوی نقاط ضعف یک سرویس دهنده بطور همزمان از نرم افزارهائی مثل FragRouter

وWhisker استفاده می نماید.(FragRouter برای فرار از IDS مبتنی بر لایه شبکه و Whisker برای فرار از IDS مبتنی بر لایه کاربرد)توصیه موکد ما آنست که هر دو نوع IDS را به روز نگاه دارید چرا که گزارشها حاکی از آنست که IDS های نوع جدید(بعد از سال ۲۰۰۰) نسبت به روشهای دهگانه Whisker هوشمند تر شده اند.

نکته بدیهی آنست که خودتان Whisker را بر علیه برنامه های CGI سرویس دهنده خود بکار بگیرید و نقاط آسیب پذیر را کشف کرده و آنها را رفع کنید.