NetCat : چاقوی همه کاره

حسین بلندقامت آذر – مرکز تحقیقات مخابرات ایران – گروه امنیت شبکه

مقدمه

نرم‌افزار کوچک ولی بسیار قدرتمند Netcat صاحب چنان آوازه‌ای شده است که به ندرت بتوان متخصص امنیتی را یافت که این نرم‌افزار را نشناسد و یا از آن استفاده نکرده باشد. این نرم‌افزار در رده‌بندی‌های ابزارهای امنیتی همیشه در بالاترین رده‌ها قرار دارد. طبق گزارش یک بررسی انجام گرفته از سوی insecure.org در سال ۲۰۰۰ به منظور تعیین ابزارهای امنیتی برگزیده کاربران ، Netcat در رده دوم قرار گرفت. در سال ۲۰۰۳ نیز Netcat توانسته است رده سوم را به خود اختصاص دهد.

Netcat نرم‌افزاری است که قادر است داده‌ها را از روی شبکه از طریق اتصالات tcp و یا udp خوانده یا بنویسد. Netcat می‌تواند به عنوان یک پویشگر پورت، backdoor، port redirector، port listener، banner grabber و ... عمل کند. البته هیچ وقت بهترین ابزار برای هر کدام از این عملیات‌ها نیست بلکه مزیت آن این است که تمام این قابلیت‌ها را یکجا در اختیار کاربر قرار می‌دهد. از این رو لقبهای مختلفی مانند "چاقوی جیبی شبکه TCP/IP" و یا "چاقوی همه‌کاره سوئیسی" به این نرم‌افزار داده شده است. این چاقو با دو لبه تیز در اختیار متخصصان امنیتی و همچنین نفوذگران می‌باشد. متخصصان امنیتی باید با توانایی‌های بلقوه این نرم‌افزار آشنا باشند تا در آینده بتوانند تدابیر امنیتی لازم را به‌کار بندند.

معرفی Netcat

Netcat نرم‌افزار بسیار ساده‌ای است که قادر است با استفاده از پروتکل‌های tcp و udp داده‌ها را بر روی شبکه ارسال و دریافت کند. Netcat به صورت ابزاری مطمئن برای back-end طراحی شده است به طوریکه می‌تواند مستقیماً و به‌سادگی توسط سایر برنامه‌ها و اسکریپتها اجرا شده و مورد استفاده قرار گیرد. در عین حال، Netcat ابزاری است با ویژگیهای متنوع برای اشکال‌زدایی و شناسایی شبکه، به طوری که می‌تواند تقریباً هر نوع اتصالی را که مورد نیاز کاربر باشد برقرار سازد و همچنین دارای چندین قابلیت جالب توجه دیگر به صورت توکار می‌باشد.

Netcat توسط Hobbit و برای سیستم‌عامل Unix نوشته شد و در سال ۱۹۹۶ منتشر شد. نسخه تحت ویندوز این نرم‌افزار نیز در سال ۱۹۹۸ توسط Weld Pond ارایه شد. هر دو نسخه تحت NIX* و ویندوز این نرم‌افزار با لیسانس freeware از آدرس http://www.atstake.com/research/tools/network_utilities قابل دریافت می‌باشد.

برخی از ویژگیهای Netcat در زیر لیست شده است:

- امکان برقراری اتصال و نیز قبول اتصال بر روی tcp و udp از/به هر پورت ممکن
- بررسی کامل ارسال DNS و ارسال معکوس DNS همراه با اعلان اخطارهای مناسب
- امکان استفاده از هر پورت مبدا
- امکان استفاده از هر آدرس شبکه مبدا
- قابلیت پویش پورت به صورت توکار همراه با تولیدکننده اعداد تصادفی
- قابلیت خواندن پارامترهای خط فرمان
- حالت ارسال کند، یک خط در هر N ثانیه
- امکان اجازه به سایر برنامه‌ها برای سرویس‌دهی به ترافیک ورودی

کامپایل و نصب

کامپایل کداصلی Netcat بسیار ساده و سرراست است. کافیست با نگاهی به فایل Makefile نـوع سیستم (systype) خود را تشخیص داده و دستور make <systype> را صـادر کنیـد. فایـل اجرایـی nc ظاهر خواهد شد. اگر در بخش systype گزینه مناسب برای سیستم خود را نیافتیـد از گزینـه‌ی generic استفاده کنید. برای سیستم‌عامل ویندوز نیازی به کامپایل نیست و نسخه باینری نرم‌افزار ارایه شده اسـت. در ادامه‌ی متن هر جا صحبت از Netcat به میان می‌آید، منظور اجرای آن در محیط NIX* می‌باشد.

بررسی ویژگیهای Netcat

Netcat بسیار ساده و در عین حال همه‌کاره است. برای توصیف تمام کارهایی که می‌شود بـا ایـن چاقوی همه‌کاره کرد، بایستی پا را فراتر از بررسی ویژگیهای آن نهاد و با ارایه مثالهایی کاربردی، تصـور بهتری از امکانات این نرم‌افزار ارایه کرد. به همین منظور مثال‌هایی در بخش انتهایی ارایه شده است.

اگر هیچ گونه پارامتری در ادامه nc به کار نرود، Netcat تقاضای پارامتر می‌کند و آمـاده دریافـت آن از طریق ورودی استاندارد خواهد بود. بعد از دریافت یک سطر ورودی، آن را به پارامترهای تشـکیل دهنده شکسته و مورد پردازش قرار می‌دهد. مزیت این روش مخفی نگه‌داشتن پارامترهای دسـتور اجـرا شده از چشم ps می‌باشد.

میزبان می‌تواند به هر دو صورت اسم یا آدرس ذکر شود. اگر n– به کار گرفته شود، Netcat تنهـا آدرسهای IP به صـورت عـددی را قبـول خواهـد کـرد و بنـابراین هـیچ‌گونـه DNS Lookup صـورت

نمی‌گیرد. ولی اگر n– ذکر نشده و v– به کار گرفته شود، Netcat به طور کامل هر دو گونه Lookup مستقیم و برعکس اسم و آدرس را برای میزبان انجام می‌دهد و در صورت عدم تطابق نام‌ها در DNS، پیغام اخطاری نمایش می‌دهد. این باعث می‌شود جهت برقراری اتصال وقت بیشتری صرف شود.

برای برقراری اتصال به خارج (Outbound)، بایستی پورت مقصد ذکر شود. پورت مقصد نیز می‌تواند به صورت عددی یا اسمی (این اسامی در /etc/services ثبت شده‌اند) درج شود. اینجا نیز اگر سوییچ n– ذکر شده باشد، تنها مقادیر عددی برای پورت مورد قبول خواهد بود. ذکر بیش از یک پورت باعث بروز رفتار دیگری می‌شود که در ادامه (پویش پورت) تشریح خواهد شد.

به طور کلی سوییچ v– میزان اعلان جزئیات را مشخص می‌کند. معمولاً بیشتر اوقات Netcat با این سوییچ اجرا می‌شود تا کاربر اطلاعاتی در مورد ارتباط جاری دریافت کند. با استفاده از سوییچ w– <timeout> می‌توانید زمان لازم برای برقراری یک اتصال را کاهش دهید. استفاده از این سوییچ به شکل w– 3 همراه با سوییچ v– بسیار معمول است و عملی مشابه telnet انجام می‌دهد. استفاده از سوییچ v– به شکل دوبار پشت سرهم (v– v–) باعث می‌شود Netcat اطلاعات بیشتری ارایه کند. اگر سوییچ v– ذکر نشود، Netcat به طور ساکت و بی‌سروصدا کار وظایف خود را انجام می‌دهد مگر اینکه با خطایی مواجه شود که در این صورت خطا را توصیف کرده و با وضعیت خروجی غیرصفر خارج می‌شود.

با استفاده از سوییچ u– ، به جای اتصال TCP، اتصال UDP برقرار خواهد شد که فی نفسه اتصال محسوب نمی‌شود زیرا که UDP پروتکل بدون اتصال است. با این حال Netcat از مکانیزم connected UDP socket که توسط بسیاری از کرنل‌ها پشتیبانی می‌شود، به‌منظور برقراری ارتباط UDP استفاده می‌کند. به هنگام ایجاد اتصال UDP عملاً تا زمان خواندن از ورودی استاندارد چیزی ارسال نمی‌شود و Netcat پورت UDP را پورت باز فرض می‌کند. تنها این موقع است که می‌توان فهمید که آیا در سمت

مقابل اتصال، سرویس‌دهنده‌ای در حال سرویس‌دهی بوده است یا نه! حتی گاهی آن را هم نمی‌شود فهمید (قطعی نیست) !!

به‌منظور تهیه یک فایل Hex از داده‌های مبادله شده از سوییچ o– logfile استفاده می‌شود. سطرهای فایل با دو علامت < و > به‌ترتیب به نشانه به سمت شبکه، و از شبکه نشانه‌گذاری می‌شود. ذخیره اطلاعات مبادله شده باعث کندی فعالیت Netcat می‌شود، بنابراین در مواردی که سرعت ارتباط عامل مهمی محسوب می‌شود؛ از این گزینه استفاده نکنید.

Netcat می‌تواند بدون هیچ‌گونه محدودیتی به هر پورتی (حتی اگر آن پورت در حال استفاده باشد) مقید[1] شود. حتی امکان استفاده از هر آدرس مبدا محلی را نیز دارد (از بین آدرس‌هایی که در کارت شبکه ثبت شده باشد). از پارامتر p– برای مشخص کردن پورت مورد نظر استفاده کنید. همچنین از s– نیز برای مشخص کردن آدرس مبدا خود می‌توانید استفاده کنید. به این عمل اصطلاحاً تعیین سوکت گفته می‌شود. کاربران با دسترسی ریشه می‌توانند هر پورت استفاده نشده‌ای را حتی زیر ۱۰۲۴ به عنوان پورت مبدا انتخاب کنند. عدم استفاده از p– باعث می‌شود که پورت مبدا توسط سیستم‌عامل و از بین پورتهای استفاده نشده انتخاب گردد –درست مانند سایر برنامه‌ها- مگر اینکه از گزینه r– استفاده شود(در ادامه توضیح داده می‌شود).

در وضعیت گوش دادن (Listening)، Netcat منتظر یک اتصال ورودی مانده و سپس تبادل داده را آغاز می‌کند. بنابراین، با استفاده از nc –l –p 1234 <filename> ، وقتی کسی به پورت ۱۲۳۴ وصل شود، فایل مزبور برایش ارسال می‌شود –خواه این فایل مورد تقاضا باشد خواه نه. وضعیت گوش دادن عموماً همراه با پارامتر پورت محلی استفاده می‌شود. اگر در وضعیت گوش دادن، میزبان مقصد و پورت

[1] Bind

محلی نیز تعیین شود، Netcat تنها از میزبان و پورت مشخص شده اتصالات را قبول کرده و بقیه اتصالات را رد خواهد کرد. اگر اعلان جزییات توسط سوییچ v- انتخاب شده باشد، Netcat با دریافت هر تقاضای اتصال، آدرس و شماره پورت متقاضی را ثبت خواهد کرد. استفاده از سوییچ d- باعث می‌شود که کنترل اجرای برنامه از کنسول جدا شده و برنامه در پشت‌زمینه اجرا شود.

درحالت معمول وقتی یک اتصال توسط Netcat دریافت شد و به پایان رسید، کار Netcat خاتمه پیدا می‌کند و اتصال دیگری توسط آن پذیرفته نمی‌شود. برای اینکه بتوان اتصالات متعددی به Netcat بر قرار کرد، از سوییچ L- استفاده می‌شود. در صورت استفاده از این سوییچ، با به پایان رسیدن هر اتصال، Netcat به طور اتوماتیک با پارامترهای قبلی دوباره اجرا می‌شود و آماده‌ی دریافت اتصال جدیدی می‌ماند.

اگر Netcat با DGAPING_SECURITY_HOLE- کامپایل شده باشد، سوییچ e- برنامه‌ای را مشخص می‌کند که بایستی بعد از دریافت یک اتصال موفق اجرا شود. کارکرد این گزینه در وضعیت گوش دادن، همانند inetd می‌باشد با این تفاوت که بدین طریق تنها یک برنامه می‌تواند اجرا شود. بنابراین بایستی بسیار مواظب این قسمت بود. این قطعه از کد در حالت عادی فعال نیست و شما نیز اگر دقیقاً از کاری که می‌کنید آگاهی دارید؛ پس از آن لذت ببرید☺. دقت کنید که تنها اجرای برنامه مورد نظر بدون امکان ارسال هیچ‌گونه پارامتری ممکن می‌باشد. بنابراین اگر قصد اجرای برنامه‌ای همراه با پارامتر را دارید، از یک فایل دسته‌ای یا اسکریپت استفاده کنید. استفاده از اسکریپتی که دوباره از Netcat استفاده شده، بسیار معمول می‌باشد.

اگر Netcat با DTELNET- کامپایل شده باشد، سوییچ t- امکان پاسخگویی به گزینه‌های مذاکره‌ی Telnet را فراهم می‌کند. بدین‌ترتیب Netcat می‌تواند به telnetd (سرویس دهنده‌ی telnet)

وصل شده و مذاکرات اولیه را تا رسیدن به اعلان login پشت سر بگذارد. به دلیل اینکه این خصیصه توان تغییر جریان داده را دارد، به طور پیش‌فرض فعال نیست. کاربر بایستی درک کاملی از نحوه و موارد استفاده آن داشته باشد و سپس شخصاً این گزینه را فعال کند.

داده‌هایی که از شبکه دریافت می‌شوند، همیشه با بیشترین کارایی ممکن در بسته‌های 8k به خروجی استاندارد تحویل داده می‌شوند. ورودی استاندارد نیز به همان طریق به شبکه ارسال می‌شود ولی سوییچ i– فرجه زمانی‌ای را تعیین می‌کند که سرعت ارسال را به میزان قابل توجهی کاهش می‌دهد. هنوز ورودی استاندارد در بسته‌های بزرگ خوانده می‌شوند ولی در ادامه Netcat سعی می‌کند تا سطرهای مختلف را کشف کرده و هر سطر را در یک فرجه زمانی ارسال کند. دقت کنید که اگر ورودی استاندارد، صفحه کلید باشد؛ داده‌ها به صورت خط به خط خوانده می‌شوند و سوییچ i– نیز تاثیر چندانی در ارسال این سطور نخواهد داشت. استفاده از سوییچ i– زمانی مفید است که ورودی استاندارد چیزی به جز صفحه کلید باشد مثلا از یک فایل یا خروجی برنامه‌ی دیگری استفاده شده باشد.

پویش پورت روش بسیار متداولی برای کشف دنیای خارج است! Netcat پارامترهای خود را اینچنین قبول می‌کند: ابتدا سوییچ‌ها، سپس میزبان مقصد، و هر چه که بعد از آن بیاید به عنوان نام یا شماره پورت‌ها تعبیر می‌شود که می‌تواند به شکل محدوده‌ای از پورت‌ها به فرم M-N نیز درج شود. اگر بیش از یک پورت مشخص شده باشد، Netcat به تمام آنها یکی‌یکی وصل شده و به همه آنها داده‌ای یکسان –که از ورودی استاندارد گرفته است– می‌فرستد. ذکر بیش از یک پورت مقصد، موجب توقف ارسال پیغام‌های تشخیص خطا می‌گردد مگر آنکه سوییچ v– دوبار درج شود.

برای انجام پویش پورت بدون ارسال داده بایستی از سوییچ z– استفاده شود. به عنوان مثال:

nc –v –w 2 –z target 20-30 سعی دارد پورتهای ۲۰ الی ۳۰ میزبان مقصد را پویش کند و چون سرویس‌هایی مانند ftp و telnet و mail در این محدوده هستند؛ با این دستور باز یا بسته بودن آنها مشخص می‌شود. سوییچ z– مانع از ارسال داده به اتصال tcp می‌شود و اطلاعات کاوشی بسیار کمی نیز به اتصال udp ارسال می‌کند و بنابراین در پویش سریع پورت بسیار مفید است. در صورت تمایل می‌توانید با i– بین دو ارسال متوالی وقفه‌ای ایجاد کنید تا سرعت اتصال کم شود.

برای هر محدوده پورت مشخص شده، عمل پویش به طور پیش‌فرض به ترتیب از بالا به پایین (از بیشترین شماره پورت به کمترین شماره پورت) انجام می‌پذیرد. اگر سوییچ r– مورد استفاده قرار گیرد، پورت‌ها به صورت تصادفی انتخاب می‌شوند. اگر برای پورت مبدا نیز از سوییچ r– استفاده شود، انتخاب پورت مبدا نیز به صورت تصادفی انتخاب می‌شود و بنابراین الگوی پویش نامنظمی به‌وجود خواهد آمد که تشخیص آن تا حدودی سخت‌تر می‌شود. اگر برای تنها یک اتصال از این گزینه استفاده شود، اختصاص پورت مبدا به جای اینکه از مقداری قبلی استفاده شده یکی بالاتر باشد (طبق فرایند معمول)، پورت مبدا مقداری تصادفی و بالاتر از ۸۱۹۲ به خود خواهد گرفت. توجه کنید که اختصاص یک پورت توسط سوییچ p– ، تاثیر سوییچ r– را خنثی می‌کند.

خلاصه‌ای از سوییچ‌های Netcat در زیر ارایه شده است.

-d	بعد از اجرای برنامه‌ی Netcat کنسول آزاد می‌شود.
-e	برنامه‌ی ثانویه‌ای را اجرا می‌کند (اگر Netcat با -DGAPING_SECURITY_HOLE کامپایل شده باشد).
-i	زمان فاصل را تنظیم می‌کند. معمولا وقتی یک فایل به عنوان ورودی استاندارد مورد استفاده قرار می‌گیرد از این گزینه برای ایجاد فاصله زمانی بین دو سطر متوالی استفاده می‌شود.
-l	Netcat را وادار می‌کند که به یک اتصال ورودی گوش دهد.

-L	Netcat را با همان پارامترهایی که برای ایجاد اتصال به کار رفته بود، دوباره اجرا می‌کند. بدین ترتیب امکان برقراری اتصال بیش از یک‌بار به یک پردازه Netcat امکان‌پذیر است.
-n	Netcat تنها آدرس‌های IP عددی را قبول می‌کند و هیچ‌گونه DNS Lookup انجام نمی‌دهد.
-o	یک فایل Hex از داده‌های مبادله شده در هر دو جهت تهیه می‌کند. سطرهای فایل با دو علامت < و > به‌ترتیب به نشانه به سمت شبکه، و از شبکه نشانه‌گذاری می‌شوند.
-p	برای برقراری اتصال خارجی نیاز است و پورت مورد نظر را مشخص می‌کند. به هنگام گوش دادن به اتصال ورودی نیز پورت مورد نظر را مشخص می‌کند.
-r	موجب می‌شود پویش پورت‌ها به صورت تصادفی انجام گیرد. همچنین برای انتخاب پورت مبدا به صورت تصادفی نیز به کار می‌رود.
-s	برای تعیین آدرس IP مبدا استفاده می‌شود.
-t	اگر Netcat با گزینه DTELNET– کامپایل شده باشد، انتخاب این پارامتر به هنگام اتصال به سرویس‌دهنده Telnet باعث می‌شود تا مذاکرات اولیه به صورت اتوماتیک انجام شود.
-u	برای برقراری اتصال UDP به جای TCP مورد استفاده قرار می‌گیرد.
-v	میزان اعلان جزئیات اتصال را افزایش می‌دهد.
-w	تعداد تلاش برای برقراری اتصال را محدود می‌کند.
-z	مانع از ارسال هر گونه داده به اتصال TCP می‌شود. در اتصال UDP نیز داده‌های بسیار محدود، تنها برای کاوش ارسال می‌شود. اصولا در پویش پورت و تنها به منظور تعیین پورتهای باز از این پارامتر استفاده می‌شود.

چند مثال عملی از کاربرد Netcat

در زیر چند مثال از نحوه کاربرد Netcat ارایه شده است تا قابلیت‌های بلقوه این نرم‌افزار ساده و در عین حال همه کاره بهتر روشن شود. بدیهی است Netcat با توجه به خلاقیت کاربران آن کاربردهای فراوانی پیدا کرده است که در اینجا تنها به چند نمونه بسیار محدود اشاره می‌شود. استفاده از Netcat به

صورت اسکریپت، همانند ابزاری قابل برنامه‌ریزی خواهد بود که حتی تصور کاربردهای آن نیز به خلاقیت زیادی نیاز دارد!

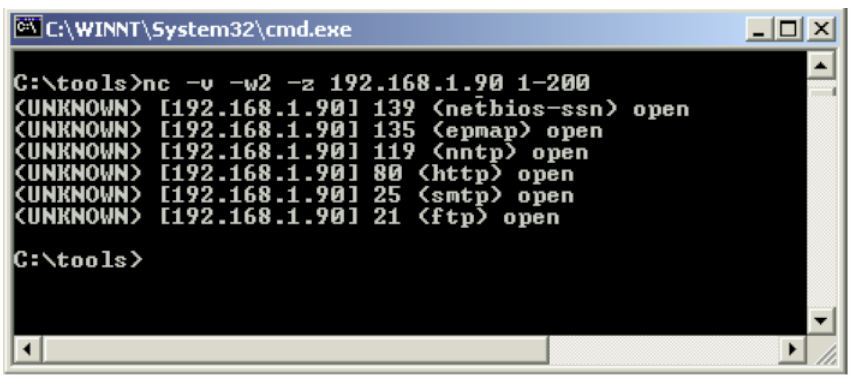
۱- استفاده به جای Telnet

Netcat به عنوان ابزاری که قادر است با daemon های مختلف مکالمه کند، جانشین کاملی برای telnet محسوب می‌شود. به عنوان مثال استفاده از آن به شکل "nc host 25" برای اتصال و مکالمه با یک سرویس‌دهنده‌ی mail (smtp) نه تنها بسیار ساده‌تر بلکه بسیار کارآمدتر نیز می‌باشد. تنها با فشار ctrl+c مکالمه پایان می‌یابد و به دستور QUIT یا c]^ که مورد نیاز telnet می‌باشد، نیازی نیست. احتمالا مایل باشید که پیغام‌های Netcat را در فایلی ذخیره کنید، بدین منظور با تغییر مسیر خروجی با استفاده از "file >&" در پوسته csh* و یا "file 2>&1 >" در پوسته bourne ، خطای استاندارد را به خروجی بفرستید.

۲- پویش پورت

دستور زیر پورتهای مابین ۱ الی ۲۰۰ از ماشین 192.168.1.90 را پویش کرده و پورت‌های باز در این بازه را همراه با اسم کاربردی آن پورت (مثلا اسم http برای ۸۰) اعلام می‌کند.

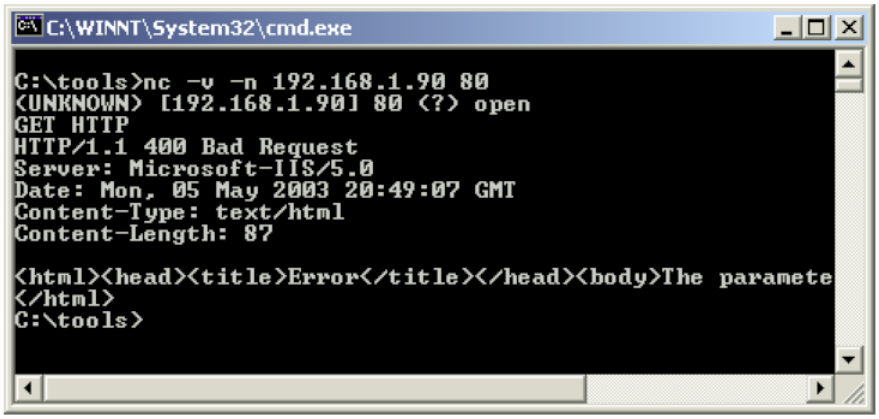
nc –v –w2 –z 192.168.1.90 1-200

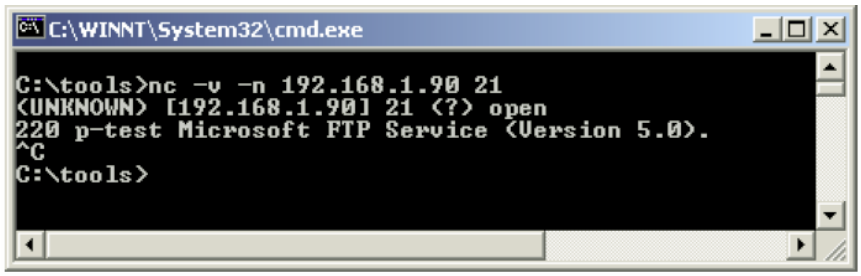


نتیجه این پویش (شکل بالا) نشان می‌دهد که پورتهای ۲۱، ۲۵، ۸۰، ۱۱۹، ۱۳۵و ۱۳۹ باز می‌باشند. دقت کنید که Netcat عمل پویش پورت را به روش مودبانه یا همان دست‌تکانی کامل سه مرحله‌ای انجام می‌دهد. بهتر است پویش‌های دقیق‌تر و حرفه‌ای‌تر توسط نرم‌افزارهای دیگری همچون nmap صورت گیرد.

۳- جمع‌آوری Banner های سرویس‌دهنده‌ها با Netcat

حال اگر مایل باشیم بدانیم که پشت پورت‌های ۸۰ یا ۲۱ چه سرویس‌دهنده‌ای فعال است، به شکل زیر از Netcat استفاده می‌کنیم.





بدین‌ترتیب به احتمال زیاد مشخص است که از ماشین ویندوز ۲۰۰۰ به همراه سرویس‌دهنده وب IIS 5.0 و Microsoft FTP Service استفاده شده است.

استفاده از دستور ترکیبی " echo QUIT | nc –v –w3 192.168.1.90 20-250 " موجب می‌شود تا Banner بسیاری از پورت‌های باز در محدوده ۲۰ الی ۲۵۰ جمع‌آوری شود. این دستور برای برخی از سرویس‌دهنده‌ها که بدون دریافت تقاضایی صریح، داده‌ای ارسال نمی‌کنند، کارایی ندارد (مانند سرویس‌دهنده‌ی وب) ولی در مورد بسیاری از سرویس‌های دیگر می‌تواند اعمال شود.

٤- سرویس‌دهنده‌ی ساده وب

از Netcat می‌توان به جای یک سرویس‌دهنده‌ی ساده‌ی وب استفاده نمود. کافی است تا پاسخ کامل HTTP را در ابتدای فایل html خود قرار دهیم. بنابراین فایلی همانند زیر تولید می‌کنیم:

```
HTTP/1.0 200 OK
Centent-type: text/plain
Content-length: 724

<HTML>
<BODY>
…
</BODY>
</HTML>
```

این فایل را با نام index.htm ذخیره می‌کنیم. در ادامه Netcat را به شکل زیر راه‌اندازی می‌کنیم:

```
nc –l –p 80 < index.htm
```

طبق این دستور، Netcat به پورت ۸۰ گوش داده و برای هر اتصالی فایل index.htm را ارسال می‌کند. پورت ۸۰ تنها به منظور همخوانی با استاندارد www انتخاب شده است وگرنه هر پورتی را می‌توان بدین منظور استفاده کرد.

این مثال بسیار ساده‌ای است که تنها از یک صفحه تشکیل شده است. می‌توان با استفاده از اسکریپت، Netcat را به ارسال صفحات مختلف و یا فایل‌های باینری مختلف نیز واداشت. مثالی در این

رابطه در مسیر scripts/web/ از بسته نرم‌افزاری Netcat قرار دارد که از Netcat به منظور ارسال صفحات مختلف استفاده کرده است. جهت مطالعه بیشتر می‌توانید به این اسکریپت مراجعه کنید.

۵- مبادله فایل توسط Netcat – جانشینی برای ftp

از Netcat می‌توان به‌عنوان ابزار ساده‌ای برای انتقال فایل استفاده کرد. مزیت عمده‌ی این کاربرد در این است که هیچ مهم نیست که کدام طرف سرویس‌دهنده و کدام طرف سرویس‌گیرنده است! ورودی در یک سمت به خروجی در سمت دیگر منتهی می‌شود. توصیه می‌شود که سرویس‌دهنده برای دریافت فایل و بدون timeout تنظیم شود در حالیکه در سمت سرویس‌گیرنده (ارسال کننده‌ی فایل) از timeout کوچکی (-w 2) استفاده شود. در اینصورت دریافت کننده فایل منتظر برقراری ارتباط خواهد ماند، بعد از برقرار ارتباط و خاتمه جریان داده، فرستنده timeout شده و اتصال را خاتمه می‌دهد که منجر به خاتمه اتصال از طرف دریافت کننده نیز می‌گردد. به عنوان مثال برای ارسال فایل myfile ، در سمت دریافت کننده چنین عمل می‌شود:

```
nc –l –p 4567 > myfile
```

و در سمت ارسال کننده نیز چنین عمل می‌شود:

```
nc  -w 3 destination  4567 < myfile
```

انتقال فایل بدون هیچ‌گونه تصدیق هویتی صورت می‌گیرد و می‌توان آن را در بخشی از cron تعریف کرد تا در زمان‌های خاصی این عمل را به منظورهای مختلفی مانند پشتیبان‌گیری انجام دهد. مثال دیگری در زیر ارایه می‌شود که از مثال قبلی کمی پیچیده‌تر است. این مثال برای ارسال یک دایرکتوری کامل استفاده می‌شود. در یک سمت (گیرنده) :

```
nc –l –p 1234 | uncompress –c | tar xvfp –
```

و در سمت دیگر (فرستنده):

```
tar cfp - /some/dir | compress –c | nc –w 3 othermachine 1234
```

این مثال محتویات یک دایرکتوری را از یک ماشین به ماشین دیگر انتقال می‌دهد بدون اینکه در مورد فایل‌های rhosts. ، حسابهای کاربری، یا پیکربندی inetd ، نگرانی‌ای در هر دو طرف وجود داشته باشد. با محدود کردن listener به آدرس و پورت خاص، می‌توان مانع از دسترسی سایرین شد. اگر Netcat دسترسی از آدرسی را قبول نکند، با حالت غیر صفر خارج خواهد شد لذا اسکریپت‌هایی که برای انجام این کار مورد استفاده قرار می‌گیرند می‌توانند به سادگی دسترسی‌های غیرمجاز را log کنند.

٦- Flooding شبکه

شاید پرکردن شبکه از حجم عظیمی از داده‌های بی‌مصرف، بسیار نامعقول به نظر برسد. ولی در بسیاری از موارد برای تست اجزای مختلف شبکه، مانند مسیریاب، سوییچ، فایروال، سیستم تشخیص نفوذ، log server و حتی برخی سرویس‌دهنده‌ها، بسیار مفید است. مسلماً Netcat برای این کار نیز راه‌حلی دارد. به عنوان مثال:

```
yes AAAAAAAAAAAAAA | nc –v –v –l –p 2222 > /dev/null
```

و در طرف دیگر:

```
yes BBBBBBBBBBBBBB | nc othermachine 2222 > /dev/null
```

باعث ایجاد ترافیک زیادی از Aها و Bها بر روی خط ارتباطی خواهد شد. استفاده از سوییچ v– v– باعث می‌شود میزان ارسال و دریافت داده، بعد از اتمام این کار (توسط کاربر)، ثبت شود. استفاده از udp حجم داده‌های تولید شده در واحد زمان را بیشتر از قبل افزایش می‌دهد و آزمایش بهتری برای سنجش کارایی محسوب می‌شود. حتی گاهی تولید داده‌های تصادفی و استفاده از آن به عنوان ورودی

سرویس‌های مختلف شبکه، باعث آشکار شدن باگ‌های برنامه می‌شود -که این‌روزها به صورت فراگیر از آن استفاده می‌شود. مثال ساده‌ای برای تولید داده‌های تصادفی در آدرس /data/data.c در بسته نرم‌افزاری Netcat وجود دارد (به همراه مجموعه کوچکی از داده‌های ورودی مختلف) که به منظور ایجاد بسته‌هایی با محتویات مختلف مورد استفاده قرار می‌گیرد. اگر شما توانستید daemon خود را با این ابزار crash کنید، به احتمال زیاد مشکل امنیتی خواهید داشت!

۷- Backdoor

Netcat قادر است هر برنامه‌ی ثانوی را با استفاده از e– اجرا کند. در Windows NT دستور زیر را صادر کنید تا از راه دور بتوانید به command prompt آن ماشین دسترسی داشته باشید:

```
nc –l –p 1234 –d –e cmd.exe –L
```

برای بهره‌برداری از آن،در طرف دیگر چنین عمل می‌کنیم:

```
nc destination 1234
```

به محض برقراری اتصال، صاحب اختیار یک کنسول از ماشین مقصد خواهید بود بدون اینکه نیاز به هیچ‌گونه تصدیق هویتی باشد. وجود پارامتر d– در سمت listener باعث می‌شود که بعد از اجرای دستور nc، کنسول آزاد شده و Netcat به‌صورت پنهانی اجرا شود. پارامتر L– نیز بدین منظور به کار رفته است که بعد از اتمام هر اتصال، Netcat دوباره در حالت listener قرار بگیرد و بعدا نیز بتوان دوباره اتصالی به آن برقرار کرد.

بدیهی است که اگر بتوانیم از ماشینی یک کنسول به صورت remote در اختیار داشته باشیم، دیگر امنیت برای آن هیچ معنی نخواهد داشت. هر گونه انتقال داده، تغییر و حذف اسناد (مانند Web

Defacing)، و تقریباً هر کار دیگری ممکن خواهد بود. با توجه به حجم کم Netcat، نفوذگرها مایلند آن را از راه‌های مختلفی به ماشین مقصد انتقال داده و از آن ماشین یک کنسول در اختیار بگیرند. با تعیین پورت‌های مبدا و مقصد به صورتی که ترافیک ایجاد شده از نظر دیوارآتش، ترافیک مجاز شناخته شود، از دیگر شگردهایی است که به سادگی توسط Netcat پیاده‌سازی می‌شود.

۸- سایر کاربردها

مطمئن باشید که برای Netcat کاربردهای بسیار بیشتر از آنچه در بالا اشاره شد وجود دارد. به عنوان مثال برای کاربردهای که قبلا مورد استفاده قرار گرفته‌اند می‌توان به موارد زیر اشاره کرد: پویشگر متنی وب، انتقال فایل به دستگاهی که تنها پویشگر وب دارد، استفاده همانند inetd، رله ترافیک، port redirect، استراق سمع برنامه کاربردی، تست remote syslogger ، تست packet filter، محافظت از سرویس‌دهنده‌ی X (رابط گرافیکی لینوکس) در برابر دسترسی خارجی، ایجاد سرویس‌دهنده‌های خاص منظوره، و بسیاری دیگر که مطمئناً مستندسازی نشده‌اند. هر روز کاربردهای جدیدی با توجه به خلاقیت کاربران (و یا شاید نیاز کاربران) از این نرم‌افزار به عمل می‌آید. ویژگی‌ها و قابلیت‌های Netcat، زمانی که با قدرت برنامه‌نویسی و در حقیقت زبان اسکریپت تلفیق شود، امکانات بی‌نظیری به این برنامه خواهد داد. یک مجموعه از اسکریپتهایی که از Netcat استفاده نموده‌اند در داخل پوشه scripts/ در بسته نرم‌افزاری Netcat وجود دارد. مطالعه بیشتر این اسکریپت‌ها توصیه می‌شود.