Erlangen von Administrator-Privilegien unter Microsoft Windows NT 4.0 durch Ausnutzung einer Sicherheitslücke im Systemcache

Ein Bericht aus der Projektarbeit im Rahmen der
Vorlesung Informationssicherheit
von Prof. Dr. Hartmut Pohl
im Sommersemester 1999 an der Fachhochschule Rhein-Sieg

stud. inf. Thomas Hungenberg

08. Juni 1999

Hintergrund

Microsoft Windows NT 4.0 benutzt einen systemweiten Cache sogenannter "Datei-Zuordnungsobjekte" (*file-mapping objects*), um dynamisch gelinkte Bibliotheken[1] (*DLLs*) möglichst schnell laden zu können. Diese Objekte werden mit Zugriffsrechten (*Permission*) der Art angelegt, daß die *Gruppe* "*Jeder*" uneingeschränkten Zugriff auf sie hat. Dadurch ist es jedem Benutzer des Systems möglich, einzelne Cache-Objekte zu löschen und durch andere zu ersetzen, die auf eigene DLLs verweisen.

Wenn ein Programm geladen und somit ein neuer *Prozeß* erzeugt wird, werden die dazu benötigten (dynamisch gelinkten) DLLs in den Adreßraum des Prozesses geladen. Wenn dabei ein passendes DLL Cache-Objekt existiert, so wird dieses einfach in den Adreßraum des Prozesses abgebildet. Die DLL wird also nicht erneut von einem Datenträger geladen. Dieses Verhalten kann für einen Angriff mittels eines *Trojanischen Pferdes* wie folgt ausgenutzt werden:

Ein *gewöhnlicher*[2] Benutzer kann eine bestimmte DLL (im folgenden als `abc.dll` bezeichnet) im Cache durch seine eigene *trojanische* DLL ersetzen. Wenn danach ein Prozeß mit hohen Privilegien gestartet wird, der dynamisch mit `abc.dll` gelinkt ist, so wird diesem die trojanische DLL zugewiesen und der Prozeß führt den Code eines niedrig-privilegierten Benutzers mit hohen Privilegien aus.

Betroffene Systeme

Betroffen sind sowohl Windows NT 4.0 Server als auch Windows NT Workstation, jeweils mit dem für die deutsche Version zur Zeit aktuellen Service Pack 4. Systeme mit früheren Service Packs sind wahrscheinlich auch betroffen; dies wurde jedoch hier nicht getestet.

Details

Ein DLL Cache-Objekt, welches von einem Benutzer (durch den Aufruf eines Programms) angelegt wird, ist nicht *permanent*. Das heißt, daß es beim Abmelden (*ausloggen*) des Users vom System aus dem Cache gelöscht wird.

Wie kann man nun erreichen, daß ein hoch-privilegierter Prozeß gestartet wird, während man als Benutzer angemeldet (*eingeloggt*) ist?

[1] *Dynamic Link Libraries* (*DLLs*) sind Objekt-Bibliotheken, die dynamisch, also bei Bedarf, geladen werden und von verschiednenen Programmen genutzt werden können.

[2] niedrig-privilegierter, also ohne *Administrator-Rechte*

Es gibt dazu viele Möglichkeiten. Man kann zum Beispiel warten, bis ein *Systemdienst* gestartet wird. Ein einfacher und verläßlicher Weg ist aber folgender:

Wenn ein neues *Windows NT Subsystem*[3] gestartet wird, wird ein *Subsystem Prozeß* erzeugt, der verschiedene *System-Details* verarbeitet. Beispiele für diese Prozesse sind `LSASS.EXE` und `PSXSS.EXE`, wobei `PSXSS.EXE` das POSIX[4] Subsystem ist, welches niemand wirklich nutzt. Daher sind die Chancen, daß dieses noch nicht in den Speicher geladen worden ist, recht groß. Wenn es allerdings einmal geladen worden ist, bleibt es bis zum Neustart des Systems im Speicher.

Um die oben beschriebene Sicherheitslücke auszunutzen und sich als gewöhnlicher Benutzer Administrator-Rechte zu verschaffen, tauscht man also eine DLL, die so gut wie von jeder gewöhnlichen Applikation geladen wird, im DLL Cache aus. Hierfür bietet sich zum Beispiel die `KERNEL32.DLL` an. Danach ruft man einen POSIX Subsystem Kommando auf, welches `PSXSS.EXE` startet. Dieses läuft mit den Zugriffsrechten von "NT AUTHORITY\SYSTEM", dem *System Account* und führt den Code der ausgetauschten DLL mit Administrator-Rechten aus.

Beispiel

Im folgenden wird ein Trojaner[5] beschrieben, der die oben beschriebene Sicherheitslücke ausnutzt und es einem gewöhnlichen Benutzer erlaubt, Administrator-Rechte zu erlangen. Dieser Trojaner wurde von *Dildog*, einem Mitglied des Projekts *L0pht Heavy Industries*[6] geschrieben.

Der Trojaner stellt eine einfache *Forwarder-DLL* dar, die die gleichen Funktionen wie die `KERNEL32.DLL` zur Verfügung stellt, aber eine andere `DllMain()`-Funktion enthält, die beim Laden der DLL aufgerufen wird. Die Funktionsaufrufe werden von dem Trojaner einfach an die echte `KERNEL32.DLL`, von der eine Kopie unter dem Namen `REALKERN.DLL` im Verzeichnis `C:\TEMP` angelegt werden muß, weitergeleitet ("*geforwarde*t").

Die folgenden Schritte beschreiben, wie man sich mit Hilfe dieses Trojaners als gewöhnlicher Benutzer - im folgenden "*Peter*" genannt - eines Windows NT-Systems Administrator-Rechte verschaffen kann. Die dazu benötigten Dateien `HACKDLL.EXE` und `EGGDLL.DLL` liegen dem Archiv mit der Beschreibung der

[3] Windows NT kann als Plattform für verschiedene Applikationstypen dienen. Es ist möglich, sowohl MS-DOS- und 16-Bit- und 32-Bit Windows-, als auch zeichenorientierte OS/2- und POSIX 1003.1-Anwendungen zu nutzen.

[4] *Portable Operating System Interface*, ein Satz von IEEE-Standards zur Gewährleistung der Portabilität von Applikationen zwischen verschiedenen UNIX-Varianten.

[5] Hier Abkürzung für "Trojanisches Pferd"

[6] Homepage auf http://www.l0pht.com

Sicherheitslücke, welches auf der Homepage von *L0pht Heavy Industries* zu finden ist, bei.

1. Einloggen als Benutzer *Peter* auf der Konsole.
2. Starten zweier “MS-DOS Eingabeaufforderungen” (`cmd.exe`) und in einem der beiden Eingabefenster folgende Aktionen ausführen:
3. Kopieren von `C:\WINNT\SYSTEM32\KERNEL32.DLL` nach `C:\TEMP\REALKERN.DLL`
4. Die Dateien `HACKDLL.EXE` und `EGGDLL.DLL` nach `C:\TEMP` kopieren.
5. Sicherstellen, daß keine Datei `C:\LOCKOUT` existiert. Wenn doch, diese Löschen, da der Trojaner diese als *Lock-Datei* benutzt.
6. Das Datei-Zuordnungsobjekt der `KERNEL32.DLL` aus dem Systemcache löschen:
   ```
   C:\> cd \temp
   C:\TEMP> hackdll -d kernel32.dll
   ```
7. Das trojanische Datei-Zuordnungsobjekt einfügen:
   ```
   C:\TEMP> hackdll -a kernel32.dll c:\temp\eggdll.dll
   ```
   Danach keine weitere Taste in diesem Fenster drücken.
8. Nun in das zweite Eingabefenster wechseln und dort ein POSIX-Subsystem Kommando starten. Als Aufruf bietet sich hier zum Beispiel
   ```
   C:\> posix /c calc
   ```
   an, wenn der *Taschenrechner* installiert ist. Ansonsten muß ein anderes Programm gewählt werden.
9. Es erscheinen nun einige Dialogboxen der `EGGDLL.DLL`, die nacheinander mit
   ’Nein’, ’Nein’, ’Ja’, ’Ja’, ’Ja’, ’Nein’ beantwortet werden müssen.
   Nach dem Beantworten der vorletzen Dialogbox wurde ein Eingabefenster mit einer *System Console* geöffnet, welche unter Administrator-Rechten läuft.
10. In das erste Eingabefenster wechseln und dort eine Taste drücken, um die trojanische DLL wieder aus dem DLL-Cache zu entfernen.

11. Auf der *System Console* den Benutzer-Manager aufrufen durch:
    - unter NT-Server: `C:\WINNT\SYSTEM32> usrmgr`
    - unter NT-Workstation: `C:\WINNT\SYSTEM32> musrmgr`
    und das Benutzer-Konto (*Account*) von *Peter* der Gruppe *Administratoren* hinzufügen.
    Peter verfügt ab jetzt über Administrator-Rechte.

Literatur

[1] L0pht Security Advisory vom 18. Februar 1999,
http://www.l0pht.com/advisories.html