CITRIX UNDER ATTACK

Wie hacke ich einen Citrix-Zugang?

Zentralisierte Lösungen wie Citrix erfreuen sich aufgrund sicherheitstechnischer und wirtschaftlicher Vorteile vor allem im professionellen Umfeld immer grösserer Beliebtheit. Administratoren entsprechender Systeme sind sich aber nur selten den Risiken bewusst, die eine derartig umfassende Multiuser-Umgebung mit sich bringt. Durch einfache Tricks können legitime Citrix-Nutzer ihre Rechte erweitern und im schlimmsten Fall mit wenigen Mausklicks die Kontrolle des Hosts erlangen.

Marc Ruef arbeitet als Security Consultant bei der schweizer Firma scip AG.

In der klassischen Informatik sind zentralisierte Host-Systeme die vorzugsweise angestrebte Lösung gewesen. Aufgrund der Tatsache, dass Hardware (und Software) sehr viel Geld gekostet haben, hat man dieses lieber in einen Host investiert, auf den die einzelnen Clients, die mit weniger Möglichkeiten ausgestattet waren, zugreifen konnten. Der Fall der Preise für Computer-Hardware hat dazu beigetragen, dass in diesem Belang eine Dezentralisierung stattfand. Da Systeme mit viel Leistung nur mehr wenig Geld kosteten und die Ausfallsicherheit verteilter Systeme grösser war, entschied man sich zunehmends für die Umsetzung leistungsfähiger Client-Lösungen.

In den letzten Jahren ist man aber, wenigstens in einigen Bereichen, wieder davon weggekommen. Der Grund liegt einmal mehr in der Performance, die durch zentralisierte Systeme viel besser und gebündelt bereitgestellt werden kann – Vor allem in Highend-Bereich ein Killer-Kriterium. Aber auch die Sicherheit, die sich bei solchen Umgebungen ebenfalls zentralisiert applizieren und administrieren lässt, ist ein Plus für host-basierte Lösungen.

In diesem Bereich wurde in den letzten Jahren vor allem ein Produkt zum Markt-Leader schlechthin. Mit Citrix MetaFrame (http://www.citrix.com) wird ein Zusatz für windows-basierte Systeme geboten, der ein System multiuser-fähig und remote-nutzbar macht. Durch einen speziellen Client kann sich ein jeder authentisierte Benutzer auf ein Citrix-System verbinden und dort in seiner eigenen Umgebung arbeiten. Die zur Verfügung gestellten Applikationen werden dabei entweder dediziert freigegeben oder der Anwender erhält gleich kompletten Zugriff auf den Desktop. In letzterem Fall kann er in der Regel auf dem System so arbeiten, als sässe er direkt davor.

Gerade weil die Sicherheit zentralisierter Lösungen ihr grosser Vorteil und damit auch der heiss diskutierte Aspekt ist, wird darauf sehr viel Wert gelegt. Grösstes Problem dabei ist nämlich, dass bei einem zentralisierten Ansatz auch ein zentralisierter Angriffspunkt gegeben ist. Will ein Angreifer ein dezentralisiertes Netzwerk übernehmen, muss er in der Regel jeden Host separat angreifen. Bei einer Lösung, wie sie durch Citrix bereitgestellt wird, muss theoretisch nur ein System fallen, um an die Daten sämtlicher Benutzer und ihrer Umgebungen zu kommen.

„Eine zentralisierte Lösung wie Citrix ist damit auch ein zentraler Angriffspunkt.“

Aus diesem Grund sind Citrix-Administratoren sehr darum bemüht, die Sicherheit ihrer Installationen so hoch wie möglich anzusetzen. Die Freigabe eines kompletten Desktops wird aus Sicherheitsgründen verweigert und deshalb lediglich dedizierte Anwendungen freigeschaltet. Ganz im Sinne der Empfehlungen im Firewalling wird also quasi eine „Whitelist“ der erlaubten Programme definiert [Ruef et al. 2002]. Damit will man verhindern, dass die normalen Anwender nach Belieben Software nutzen können, die sie für ihre Tätigkeit gar nicht brauchen. Diese Einschränkung soll die Chance auf Übergriffe durch den Missbrauch von Funktionen und ihren Schwachstellen minimieren.

In einem solchen Umfeld wird ein Angreifer jedoch darum bemüht sein, seine Rechte auf einem System auszuweiten. Obschon er legitimer Benutzer mit einer dedizierten Umgebung ist, will er zusätzliche Zugriffe umsetzen, eventuell nach Belieben Software starten oder gar auf die Daten der anderen Citrix-Anwender zugreifen können. Diese Abschrift behandelt explizit das Thema des Penetration Testings von Citrix-Umgebungen als authentisierter Benutzer. Es werden die einzelnen Angriffstechniken zur Erlangung der Rechteausweitung exemplarisch vorgestellt, ohne auf sämtliche Varianten und Ausnahmen einzugehen.

Auto-Complete als freizügiger Suchdienst

Hat ein Angreifer Zugriff auf ein System, möchte er sich normalerweise als erstes einen Überblick verschaffen [Ruef et al. 2002]. Bei lokalen Attacken und Kompromittierungen komplexerer Natur schliesst dies die Analyse des Dateisystems mit ein. Der Angreifer versucht sich in diesem Fall über den Aufbau der Verzeichnisstruktur im Klaren zu werden, um spezifische Dateizugriffe und Manipulationen effizient vorbereiten und umsetzen zu können.

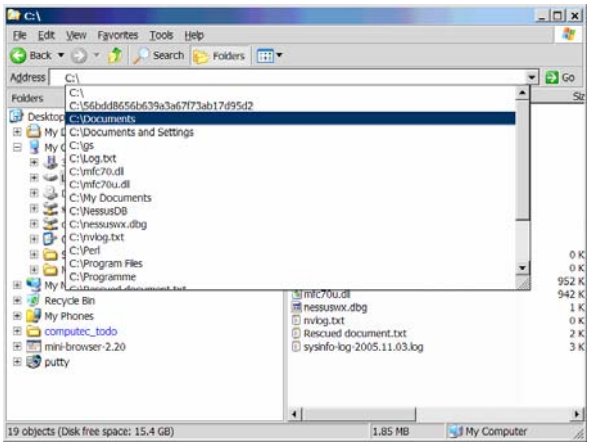

Abbildung 1: Die komfortable Auto-Complete-Funktion macht Vorschläge für die Dateizugriffe.

Um das Risiko einer Entdeckung durch fehlerhafte Zugriffe zu verringern, kann er auf das Auto-Complete Feature so mancher Software zurückgreifen. Diese Funktionalität verspricht ein Mehr an Komfort für die Benutzer, denn so komplettiert es halbfertige Pfad- und Dateiangaben. Sind mehrdeutige Lösungen vorhaben, wird eine Auswahlliste angezeigt. Durch das einfache Eingeben von C:\ lassen sich somit alle Unterverzeichnisse und Dateien des Wurzelverzeichnisses mit dem Laufwerksbuchstaben C anzeigen. Selbiges wiederum mit C:\WINDOWS\ für das Windows-Verzeichnis, usw.

Das Auto-Complete Feature wird vor allem beim Microsoft Internet Explorer und explorer.exe angeboten. Aber auch andere Teile des Windows-Systems trumpfen damit auf. So ist es ebenfalls ein Teil der über System-Bibliotheken wie mit comdlg32.ocx generierte Dialogboxen für das Öffnen und Speichern von Dateien. Dort kann in der Textbox für die Angabe der zu bearbeitenden Datei eine nicht-eindeutige Pfadangabe umgesetzt und damit die Auswahlliste provoziert werden. Da dabei keine direkten Dateizugriffe stattfinden, werden auch keine Fehlermeldungen und Protokoll-Einträge auf dem System generiert. Der Angreifer kann somit unbehelligt und in aller Ruhe das Mapping des Laufwerks umsetzen.

Erweiterte Leserechte dank Dialogboxen

In sicheren Citrix-Umgebungen werden lediglich dedizierte Anwendungen freigegeben. Greift ein Benutzer mit seinem ICA-Client auf eine solche zu, wird diese auf dem MetaFrame-Server gestartet. Sämtliche Ausgaben des Programms werden auf den Bildschirm des Clients geschickt. Dieser wiederum schickt die Eingaben des Benutzers - vorwiegend die Befehle über Tastatur und Maus - an die Anwendung auf dem Server. In diesem Fall hat der Benutzer keinen direkten Zugriff auf die Desktop-Umgebung. Die Start-Leiste bleibt dabei genauso verborgen, wie der Desktop-Hintergrund oder der Zugriff auf andere Prozesse (z.B. Taskmanager).

Ein Angreifer wird nun mittelfristig versuchen, aus dem Kontext seiner Applikation(en) auszubrechen. Dies kann mit verschiedenen Techniken umgesetzt werden. Betrachten wir als erstes die schlichte Möglichkeit, auf untypische Dateien mit Leserechten zuzugreifen. Dies geschieht in der Regel durch das Miteinbeziehen eines Öffnen-Dialogs, wie sie so manche interaktive Applikation (z.B. Microsoft Word oder Outlook) zur Verfügung stellt. Nachdem ein Benutzer diese

„Dialogboxen erscheinen ungefährlich – Es erstaunt deshalb, dass sie ein System zu Fall bringen können."

Dialogbox vorgesetzt bekommt, zeigt ihm diese gewisse Dateien an. Oftmals wird von der Anwendung, die diese Dialogbox generiert, ein Filter für die Dateien gesetzt. Bei notepad.exe ist dies Beispielsweise der Filter für die Dateiendung txt. Es wird also mittels *.txt angegeben, dass nur Dateien, die auf .txt enden, angezeigt werden sollen. Eine erste Möglichkeit, zusätzli-

che Einsichten zu erhalten, ist in der Änderung des Filters auf * gegeben. Mit dieser absoluten Wildcard-Angabe wird die Dialogbox angewiesen, sämtliche Dateien im aktuellen Verzeichnis anzuzeigen. Durch diese erste Massnahme ist es möglich über den ersten Tellerrand hinaus zu sehen.

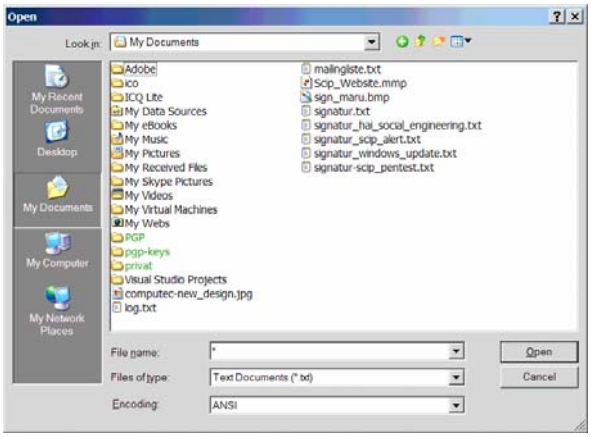

Abbildung 2: Durch das manuelle Setzen von Filtern können zusätzliche Dateien angezeigt werden.

Ein ähnlicher Trick, der sich ebenfalls die Menü-Funktionalität einer Software zunutze macht, ist in der Anzeige des Quelltextes einer Webseite über den Microsoft Internet Explorer gegeben. Zeigt dieser eine Webseite an, kann über den Menü-Punkt View/Source der Quelltext des geladenen Dokuments angezeigt werden. Standardmässig wird dabei auf Notepad zurückgegriffen, das sodann wiederum weitere Zugriffsmöglichkeiten und Manipulationsoptionen gewährt. Die Definition der standardmässig in diesem Zusammenhang zu nutzende Applikation kann in den Internet Optionen des Explorers umgesetzt werden. Von diesem Problem betroffen ist grundsätzliche jede Software, die auf den Internet Explorer zur Anzeige von Dokumenten zurückgreift. Bekanntestes Beispiel ist hierbei Microsoft Outlook.

Weitere Leserechte dank Directory Traversal

Eine Dialogbox lässt standardmässig den Wechsel in andere Verzeichnisse zu. Dies ist komfortabel durch eine Dropdown-Liste möglich, mit der man sich durch die Hierarchie der Verzeichnisse bewegen kann. Aber gerade in sicheren Citrix-Umgebungen wird diese Funktionalität gänzlich oder wenigstens teilweise abgeschaltet. In solchen Fällen erhält man dann eine Fehlermeldung, will man auf ein Verzeichnis zugreifen, für das die erforderlichen Rechte nicht existieren.

Ein gutes Beispiel ist mit dedizierten Benutzerverzeichnissen gegeben. So finden sich dann im Verzeichnis C:\Documents and Settings die Heimverzeichnisse der jeweiligen Benutzer. Beispielsweise lautet dann für den Anwender mit dem Benutzernamen maru das Heimverzeichnis C:\Documents and Settings\maru. Wird nun standardmässig in einer Dialogbox dieses Verzeichnis geladen, kann der Benutzer versuchen, eine Hierarchie-Stufe höher im Verzeichnisbaum zu steigen. Ein sicherheitsbewusster Administrator wird jedoch die Rechte des Verzeichnisses C:\Documents and Settings so eingeschränkt haben, dass nicht jeder Benutzer Nutzungsrechte für dieses besitzt. Eine Fehlermeldung mit Hinweis auf das Fehlen der entsprechenden Privilegien wird der Fall sein.

Oftmals vergessen Administratoren jedoch, dass auch das noch höhere Verzeichnis, in diesem Beispiel das Wurzelverzeichnis C:\ an sich, ebenfalls mit einer Zugriffslimitierung ausgestattet werden sollte. Der Benutzer kann sich nun also bemühen, nicht nur eine, sondern gleich zwei oder noch mehr Stufen höher im Verzeichnisbaum zu kommen. Dass bestimmte Unterverzeichnisse eine Rechtelimitierung aufgewiesen hätten, ist bei diesem nicht direkt darauf umgesetzten Zugriff irrelevant. Man ist quasi aus dem vorgegebenen Verzeichnis-Kontext gesprungen.

„Aus vielen Webbrowsern heraus lässt sich ein Texteditor starten, der weitere Zugriffe ermöglicht."

Ist die Dropdown-Liste mit den Verzeichnisnamen nicht vorhanden oder kann nicht auf den "Up One Level"-Button zurückgegriffen werden, muss man die Verzeichnisnamen in der Textbox für die Dateinamen selber eingeben. Dabei bedient man sich der klassischen Directory Traversal, wie man sie auch von Angriffen auf Webserver her kennt. Gibt man in der besagten Textbox den Dateinamen test.txt an, wird nun eben diese Datei im aktuellen Verzeichnis der Dialogbox geöffnet. Ändert man nun jedoch dies zum relativen Pfad ..\..\config.sys, kann man aus dem ursprünglichen Verzeichnis springen und direkt auf eine andere Datei, in diesem Fall config.sys zwei Stufen höher in der Verzeichnis-Hierarchie, zugreifen.

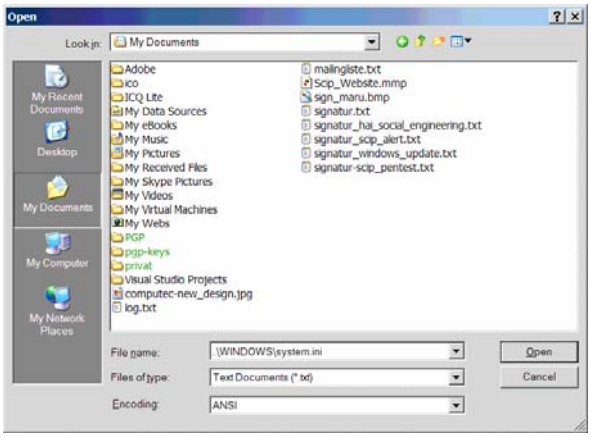

Abbildung 3: In vielen Anwendungen lassen sich aus dem Web-Bereich bekannte Directory Traversal-Attacken umsetzen.

Dies funktioniert natürlich auch mit komplexeren Verzeichnisangaben, bei denen Hierarchie-Stufen verlassen und wieder betreten werden können. Nehmen wir als Beispiel den Zugriff ..\..\WINDOWS\system.ini, bei dem zuerst mit ..\..\ zwei Stufen höher gesprungen wird. Danach wird das Verzeichnis WINDOWS betreten und auf die Datei mit dem Namen system.ini zugegriffen. Ist nun dieses spezifische Zielverzeichnis und/oder die Zieldatei nicht zusätzlich geschützt, wird dieser Zugriff funktionieren.

Alternative Dialogboxen und mehr

In einigen Fällen werden Schutzmassnahmen umgesetzt, so dass die Öffnen-Dialogbox keine Änderung des Filters, des Wechselns von Verzeichnissen oder gar des Öffnen alternativer Quellen zulässt. In diesem Fall lohnt es sich auszuprobieren, ob diese Schutzvorkehrungen ebenfalls für den Speichern-Dialog angesetzt wurden. In den meisten Fällen übersehen die Entwickler und Administratoren nämlich, dass sich die Öffnen- und Speichern-Dialogboxen technisch nicht voneinander unterscheiden (es

„Es gibt verschiedene Varianten von Directory Traversal-Zugriffen, die nicht einfach zu erkennen sind."

werden bei der Anzeige dieser Standard-Elemente exakt die gleichen DLL-Aufrufe umgesetzt; erst nach dem Klicken auf den Bestätigungs-Button wird eine andere Aktion, die zudem vom Entwickler spezifiziert werden muss, ausgelöst). Folgend ein simples Beispiel mit Microsoft Visual Basic 6.0, bei dem unterschiedliche primitive Menu-Zugriffe für Open und Save As vorgetragen werden.

```
Private Sub mnuFileOpen_Click()
   'Zeige die Öffnen-Dialogbox
   cdgOpen.ShowOpen

   'Lade die angegebene Datei
   LoadFile (cdgOpen.Filename)
End Sub

Private Sub mnuFileSaveAs_Click()
   'Öffne die Save As-Dialogbox
   cdgSaveAs.ShowSave

   'Speichere die angegebene Datei
   SaveFile (cdgSaveAs.Filename)
End Sub
```

Abbildung 4: Viele Dialogboxen verarbeiten die Eingaben ohne zusätzliche Überprüfungen.

Einige Entwickler wissen um die Möglichkeiten solcher Directory Traversal-Attacken [Hoglund et al. 2004]. Sodann verhindern sie eine Eingabe der dafür typischen ..\ Zeichenkette, die das Herausspringen aus Verzeichnissen ermöglicht. Wiederum ein einfaches Beispiel einer Öffnen-Dialogbox mit Visual Basic, bei der die Funktion Replace() die Schutzmassnahme übernimmt:

```
Private Sub mnuFileOpen_Click()
   'Zeige die Öffnen-Dialogbox
   cdgOpen.ShowOpen

   'Lade ohne Dir. Traversal
   LoadFile (Re-
  place(cdgOpen.Filename, "..\", "")
End Sub
```

Abbildung 5: Durch Suchen-Ersetzen können korrupte Eingaben erkannt und verhindert werden.

In den besagten Codezeilen findet einfach eine Ersetzung der verdächtigen Zeichenketten ..\ statt. Ist dieses Muster in der Dateiangabe der Dialogbox enthalten, wird es somit schlichtweg gelöscht. Der Entwickler hat aber hierbei übersehen, dass Windows-Systeme ebenfalls den alternativen Slash / zur Spezifizierung eines Pfads zulassen. Der Angreifer könnte nun also anstatt ..\..\ einfach das equivalente Muster ../../ nutzen. Die logische Erweiterung der Suchen-Ersetzen-Funktion zur Erreichung eines adäquaten Schutzes ist absehbar.

In solchen Fällen kann man sich zudem einem anderen Trick bedienen, der trotzdem direkte Zugriffe auf andere Verzeichnisse und Dateien erlaubt. Und zwar kann man absolute Pfadangaben zur Referenzierung heranziehen. Anstatt des

Zugriffs mittels ..\..\WINDOWS\system.ini kann einfach der direkte Pfad mit C:\WINDOWS\system.ini angegeben werden. Nur die wenigstens Entwickler, falls überhaupt, schützen sich vor solchen direkten Zugriffen. Diese sind natürlich auch für andere Laufwerksbuchstaben und gar Netzwerkverzeichnisse (z.B. \\192.168.0.1\test) möglich.

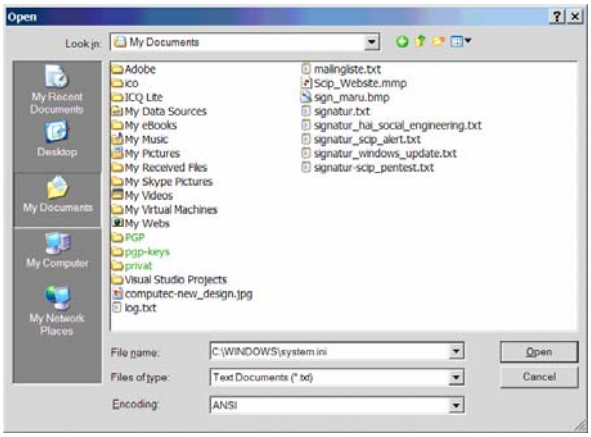

Abbildung 6: Absolute Pfad-Angaben ermöglichen direkte Zugriffe auf Bereiche, die sich sonst nicht erreichen lassen.

Entwickler, die nicht auf die API-Funktionen und Bibliotheken anderer Hersteller (z.B. Microsoft) zurückgreifen wollen, schreiben sich entsprechend ihre Masken selbst. Ein beliebter Trick, um dort unerwünschten Eingaben zuvor zu kommen, ist das Deaktivieren gewisser Tastatur-Eingaben. Dabei werden die jeweiligen Events in KeyDown(), KeyUp() und KeyPress() abgefangen und anhand einer Blacklist der KeyCodes aussortiert. So kann es durchaus sein, dass in einem Feld, in dem lediglich numerische Werte erwartet werden, sämtlichen Buchstaben und Sonderzeichen gar nicht erst angenommen werden. In solchen Ausnahmefällen kann manchmal mittels Copy & Paste trotzdem noch der unerwartete Inhalt erstellt werden. Ob und inwiefern diese Möglichkeit für einen Angreifer von Vorteil sein kann, ist danach stark von der programminternen Verarbeitung der Daten abhängig.

Diese Angriffstechniken funktionieren jedoch nur, wenn die Schutzmassnahmen innerhalb der missbrauchten Applikation ihren Einsatz finden. Also dann, wenn die Entwickler unerwünschte Eingaben direkt im Programm selbst abfangen. Werden die Schutzmassnahmen umfassend auf der Betriebssystem-Ebene angesetzt (z.B. NTFS-Rechte oder Group Policies), versagen diese Methoden.

Applikationen mittels Dialogboxen offnen

Über Dialogboxen können jedoch nicht nur auf untypische Dateitypen oder mittels Directory Traversal auf vermeintlich geschützte Bereiche zugegriffen werden. Ebenso lassen sich ausführbare Dateien starten. Als erstes öffnet man eine Dialogbox, die von sich aus möglichst viele Rechte mitbringt. Danach steuert man, egal ob mit den normalen Buttons oder mit Directory Traversal, ein Verzeichnis an, in dem sich ausführbare Dateien befinden. Nachdem der Filter auf * oder *.exe gesetzt wurde, sollten die entsprechenden Programme angezeigt werden. Sodann klickt man mit der rechten Maustaste auf jene Applikation, die man starten möchte. Daraufhin öffnet sich das Kontext-Menü, das zusätzliche Manipulationen der besagten Datei erlaubt. Einer der Punkte lautet Open und lässt das Öffnen der Applikation zu.

Dateiendung	Dateityp	Beschreibung	Gefahr
EXE	Reguläre ausführbare Dateien	Die meisten ausführbaren Dateien in Microsoft-Umgebungen werden zu EXE-Dateien kompiliert.	Sehr hoch
COM	Klassische Kompilierte Dateien	Als Relikt aus MS DOS-Zeiten werden kleinere Applikationen in COM-Dateien kompiliert.	Hoch
BAT	Stapelverarbeitungs-Dateien	Mittels Batch-Dateien lassen sich zeilenbasiertes Skripting-Umsetzen und teilweise auf systemnahe Komponenten und Variablen zugreifen.	Mittel
CMD	Stapelverarbeitungs-Dateien von NT	Dies sind spezifische Stapelverarbeitungs-Dateien aus dem Windows NT-Umfeld (ähnlich BAT).	Mittel
SCR	Bildschirmschoner	Dieses klassische Dateiformat für Bildschirmschoner stellt eigentlich eine üblich kompilierte Datei dar, die jedoch gewisse API-Schnittstellen zum regulären Einsatz als Screensaver zulässt.	Hoch
PIF	Program Informations Datei	Das Relikt aus sehr frühen Windows-Zeiten erlaubt den Aufruf von Programmen mit zusätzlicher Initiierung bestimmter Prozeduren.	Mittel
SHS	Shell Scrap Object Datei	Im Rahmen der Embedded-Ideologie von Microsoft wird dieses Dateiformat verwendet, um ausführbare Inhalte bereitzustellen. Derlei Mechanismen lassen sich nur schwer erkennen.	Mittel

Abbildung 7: Es gibt eine Vielzahl an Datei-Erweiterungen, die sich für aktive Inhalte und damit für erweiterte Übergriffe nutzen lassen. Viele davon sind schon im Bereich der Computerviren ein bekanntes Risiko.

Kann dies erfolgreich umgesetzt werden, startet auf dem MetaFrame-Server eben die initialisierte Anwendung. Diese wird im Kontext des Benutzers bzw. seiner Dialogbox geladen und kann nun wie jede andere über Citrix freigegebene Anwendung bedient werden. Sehr einfach lassen sich so weitere Programme öffnen. Ein beliebtes Ziel bleibt dabei notepad.exe, das sich standardmässig im Windows-Verzeichnis %windir% (meist C:\WINDOWS oder C:\WINNT) befindet. Über diesen einfachen Texteditor können sodann Manipulationen an Dateien vorgenommen oder anderweitig erweiterte Rechte erlangt werden. Ebenso beliebt sind cmd.exe bzw. command.com, über die die MS DOS-Eingabeaufforderung erlangt werden kann. Ist dieser Schritt geglückt, können sehr einfach und effizient Manipulationen auf Dateiebene umgesetzt werden. Wie dies im Detail aussieht und mit welchen Problemen sich ein Angreifer konfrontiert sehen könnte, werden wir weiter unten besprechen.

„Assoziationen von Datei-Erweiterungen können dazu missbraucht werden, Programme zu starten."

Es kann aber durchaus vorkommen, dass eine Dialogbox die erforderlichen Rechte nicht mitbringt, um ausführbare Dateien anzuzeigen oder sie gar über das Kontext-Menü zu starten. Sodann kann ein indirekter Angriff umgesetzt werden, mit dem sich trotzdem externe Applikationen - wenigstens einige - öffnen lassen. Der Trick besteht darin, dass man gewisse Dateitypen öffnet, die mit einer Applikation assoziiert werden. Setzen wir als Beispiel den Filter der Dialogbox von notepad.exe auf *.xls, um sämtliche Excel-Sheets des gegenwärtigen Verzeichnisses anzuzeigen. Anstatt die neu angezeigten XLS-Dateien direkt mit der Dialogbox zu öffnen, wird eine solche mit der rechten Maustaste angeklickt. Im sodann geöffneten Kontextmenü sind einige Punkte für den Angriff relevant. In manchen Fällen taucht nämlich der Eintrag "Edit" auf, bei dem eine Datei sofort mit dem verknüpften Editor verändert werden kann (z.B. Bilddateien wie BMP und JPEG mit Microsoft Paint oder XLS-Sheets mit Microsoft Excel). Damit ist schnell eine weitere Applikation gestartet.

Zu Problemen kann es kommen, wenn halt eben gerade keine spezifischen Dateitypen einer Applikation vorhanden sind, die man gerne ansteuern würde. In solchen Fällen kann man sich mit einfachen Mitteln Abhilfe schaffen. Ist es einem Angreifer möglich eigene Dateien zu schreiben oder bestehende umzubenennen, kann man einen Dateinamen generieren lassen, der eine vermeintliche Assoziation mit der Ziel-Applikation vortäuscht. Zum Beispiel erstellt man (mit Notepad) eine leere Datei mit dem Namen pseudo.xls. Wird diese nun über das Kontextmenü von Notepad editiert, öffnet sich Excel, da dies die der Dateiendung xls zugewiesene Anwendung ist. So lassen sich teilweise auch Programme starten, die weder freigegeben noch anderweitig ausgewiesen, aber dennoch auf dem Citrix-Server installiert sind. Die Information dieser Zuweisungen wird simpel in der Registry in HKEY_CLASSES_ROOT gespeichert.

Externe Applikationen mittels URIs

Eine klassische Angriffstechnik zur Ausweitung von Rechten über eine Anwendung ist mittels des URI-Aufrufs gegeben. Eine URI ist eine registrierte Komponente, die die Ansteuerung einer Ressource (URL) mittels spezifischer Software ermöglicht. Beispielsweise ist im Link http://www.scip.ch die Zeichenkette http:// die URI für den HTTP-Aufruf über den Webbrowser. Wird ein solcher Link geöffnet, startet sich normalerweise der Browser und dieser wiederum lädt die eigentliche Ressource, in diesem Fall die Webseite mit dem Namen www.scip.ch. So gibt es nun verschiedene URIs für verschiedene Klassen. Die populärsten Schemata, da zu grossen Teilen standardisiert, sind:

- HTTP (Hyper Text Transfer Protocol, tcp/80 und tcp/443); Öffnet normalerweise den Standard-Webbrowser (z.B. Microsoft Internet Explorer) und darin die angegebene URL, Standardports.
- FTP (File Transfer Protocol, tcp/21); FTP-Client, falls kein solcher Installiert ist, wird auf Windows-Systemen auf den Explorer zurückgegriffen.
- MAILTO (Mail); öffnet ein Mail-Client zum Versand von Email (normalerweise SMTP, tcp/25).
- NNTP (News Network Transfer Protocol, tcp/119); Startet den News-Client (USENET), auf Windows-Systemen ist standardmässig keiner Vorhanden.
- Gopher (tcp/70), ein schon etwas angestaubter Dienst im Stil von HTTP, als Client wird in Windows-Umgebungen ebenfalls der Microsoft Internet Explorer genutzt.

Ist es nun möglich aus einer Applikation heraus eine URI anzuwählen, wird mit eben dieser die

damit verknüpfte Anwendung gestartet. Sehr einfach lässt sich so etwas mit einem HTML-Dokument automatisieren, in dem sich die Links für die jeweiligen Zugriffe finden. Dies kann mit FTP-Client starten sehr einfach umgesetzt werden. Aber gerade im Zusammenhang mit dem Internet Explorer ist dies eher uninteressant, da die meisten URIs sowieso lediglich auf Komponenten dessen zurückgreifen (z.B. http://, ftp://, gopher://).

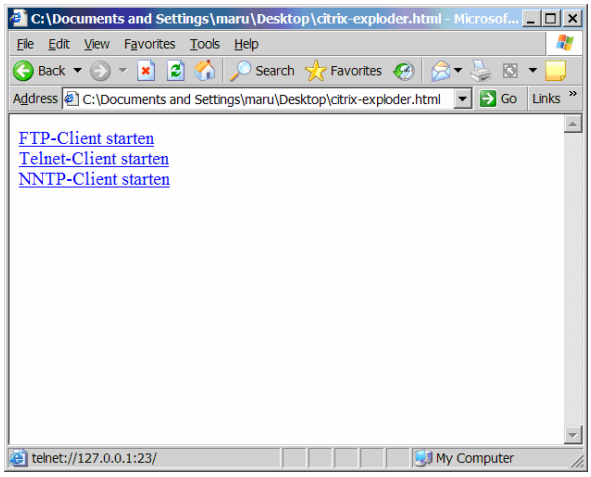

Abbildung 8: Mit HTML-Dateien lassen sich mittels traditionellen Links externe Applikationen aufrufen.

Interessant bleibt aber zu bemerken, dass viele andere Anwendungen aus dem Hause Microsoft eine (automatische) URI-Interpretation zulassen. Allen voran die Applikationen aus dem Office-Paket (http://office.microsoft.com). Wird Beispielsweise in einer Excel-Zelle der Begriff http://www.scip.ch eingetragen und einmal Enter gedrückt, verwandelt sich die Zeichenkette in einen Hyperlink. Durch einmaliges Anklicken dieses wird die URI samt Anwendung (im Normalfall der Internet Explorer als Standardbrowser) geöffnet. Auch Word-Dokumente sind in der Hinsicht sehr nützlich, verwandeln sich viele Zeichenketten mit vorangestellter URI automatisch in einen entsprechenden Hyperlink.

Besonderes Interesse geniesst die in Windows-Umgebungen zur Verfügung gestellte URI file://. Mit dieser können lokale Dateizugriffe definiert werden. Nehmen wir einmal mehr die Tabellenkalkulation Excel als Beispiel. Geben wir dort in einer Zelle den Wert file://C:\WINDOWS\system.ini ein, verwandelt sich dieser wiederum in einen Hyperlink. Dieser verweist nun aber nicht mehr auf eine Netzwerkressource, sondern auf die lokale Datei system.ini im Windows-Verzeichnis C:\WINDOWS. Ein Anklicken dieses Links öffnet sodann den Explorer, mit dem unverzüglich erweiterte Rechte genossen werden können. Eine komfortable Einsicht und Manipulation des Systems wird möglich. Um den Datei-Explorer samt Anzeige eines Verzeichnis-Inhalts zu erzwingen, kann einfach ein file-Zugriff auf einen Verzeichnis-Namen, beispielsweise file://C:\WINDOWS für das Verzeichnis C:\WINDOWS initiiert werden.

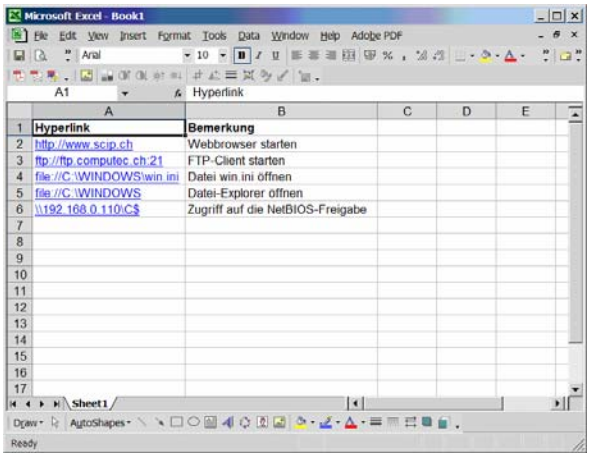

Abbildung 9: Ebenfalls viele Anwendungen erlauben eine automatische Interpretation von Links und URIs.

Auch der direkte Dateiaufruf wird dadurch möglich, indem einfach der absolute Pfad und Dateiname der zu startenden Anwendung angegeben wird. Um das Notepad zu starten, kann simpel auf file://C:\WINDOWS\system32\notepad.exe verwiesen bzw. zugegriffen werden, sofern es sich um eine Standard-Installation jüngeren Datums handelt.

Eine jede Applikation sieht sich theoretisch in der Lage, eigene URIs zu registrieren, sofern natürlich beim Installations-Prozess die dazu erforderlichen Rechte vorhanden waren. Beispielsweise registriert sich die populäre Voice-over-IP Anwendung Skype (http://www.skype.com) die URIs skype:// und callto://. Eine Liste der gängigsten URIs kann nützlich sein, um die potentiell dadurch aufrufbaren Anwendungen durchprobieren zu können.

Zugriffsschutz auf cmd.exe umgehen

In vielen Umgebungen verhindern Administratoren – und dies zu Recht -, dass Benutzer auf die MS DOS-Eingabeaufforderung zugreifen können [Microsoft 2004]. Dies geschieht, abgesehen von etwaigen NTFS-Zugriffslimitierungen, im System selbst. Wird über cmd.exe die Kommandozeile zur interaktiven Verarbeitung initiiert, generiert dies eine entsprechende Fehlermeldung. Der Benutzer wird sodann auf die Restriktion hingewiesen und muss mit einem Abbruch des gewünschten Zugriffs vorlieb nehmen.

Dabei gilt es jedoch zu bedenken, dass die Datei cmd.exe historisch mit Microsoft Windows NT gewachsen ist. Sie ist also auf allen NT-Systemen und –Derivaten (dazu zählen auch Windows 2000, XP und Server 2003) für die Kommandozeile zuständig. Auf älteren Windows-Systemen wie 95, 98 und ME kam jedoch command.com als Eingabeaufforderung zum Einsatz. Dieses Relikt vergangener MS DOS-Tage wird jedoch zwecks Rückwärtskompatibelität ebenfalls auf neueren Windows-Systemen angeboten. Interessant bleibt dabei zu bemerken, dass die über appsec umgesetzten Restriktionen, die für cmd.exe greifen, nicht ebenfalls automatisch auf command.com appliziert werden.

Das Nutzen von command.com hat für einen Angreifer nur marginale Nachteile. So muss er auf eine komfortable History-Funktion verzichten, kann nicht umfänglich mit langen Datei-/Verzeichnisnamen von VFAT/FAT32 und NTFS arbeiten und muss mit einer schlechteren Performance (nur 16-bit) umgehen können [Microsoft 2005]. Dies sind aber sowieso Dinge, die einen Angreifer eher weniger interessieren.

Abbildung 10: Zugriffe auf Dateiebene lassen sich genauso gut mit dem klassischen command.com-Interpreter (MS-DOS-Eingabeaufforderung) durchsetzen.

In manchen Fällen sind die Zugriffslimitierungen für die Kommandozeilen cmd.exe und command.com explizit auf ihre absoluten Pfade angewandt worden. Dies bedeutet, dass auf dem Windows-System der absolute Pfad, zum Beispiel C:\WINDOWS\system32\cmd.exe, vermerkt wird. Dies eröffnet die Möglichkeiten, eine Kopie der ursprünglichen Datei anzulegen. Speichert man diese zum Beispiel neu unter C:\shell.exe ab, kann die Blacklist-Funktionalität nicht mehr greifen. Sodann sieht man sich in der Lage, eine zuvor gesperrte Datei auszuführen.

Aufgrund strenger Limitierungen im Kopieren und Verschieben von Dateien kann es sehr schwierig sein, eine Adaption einer solchen anzulegen. In derartigen Situationen ist es nützlich, wenn man sich die gewünschten Dateien anderweitig auf den Server laden kann. Oftmals ist dies durch den Microsoft Internet Explorer möglich, durch den die entsprechenden Daten auf die Festplatte des Citrix-Systems gezogen werden können. Von Vorteil in solchen Situationen ist es, wenn entsprechend irgendwo im Internet ein Server steht, der die jeweiligen Versionen von cmd.exe oder explorer.exe zur Verfügung stellt. Eine mit einem spezifischen Windows-System mitgelieferte Datei kann nämlich nicht so ohne weiteres in einer anderen Windows-Umgebung genutzt werden.

Gleiches ist natürlich auch dann gegeben, wenn Whitelists zum Einsatz kommen, bei denen einzelne Programme freigeschaltet sind. Ausnahmen werden nämlich gerne für notepad.exe gemacht, das sich auf den meisten interaktiven Systemen früher oder später als Nützlich erweist. Ein Angreifer könnte in diesem Fall eine beliebige EXE-Datei auf den Citrix-Server laden (z.B. einen Exploit oder ein Debugging-Utility), diese dort als notepad.exe abspeichern und damit ausführen lassen.

Vererbung von Rechten

Obschon viele Administratoren darum bemüht sind, dass die einzelnen freigegebenen Anwendungen nur so wenig Rechte wie möglich für sich in Anspruch nehmen können, erfordert das eine oder andere Programm von sich aus erweiterte Zugriffsrechte auf dem System. So ist es dann nicht verwunderlich, wenn dedizierte DLL-Dateien gelesen werden können, obschon andere Bereiche in keinster Weise sichtbar sind. Dies kann dazu führen, dass der Angreifer quasi die neuen Rechte einer Applikation erbt. Dank dieser neuen Rechte könnte er weitere Datei- oder Programmzugriffe umsetzen und so weitere Privilegien erlangen.

„Besonders systemnahe Anwendungen pflegen erweiterte Rechte zu nutzen und anzubieten."

Besonders davon betroffen sind systemnahe Anwendungen. Dazu zählen vor allem Server-Dienste, die sich oftmals tief im System einnisten und zur Erledigung ihrer Arbeiten die zusätzlichen Rechte (oftmals Administrator oder gar SYSTEM) erfordern. Hat ein Angreifer einmal eine solche Applikation initiiert und erlaubt diese zusätzlichen Zugriffe, ist es nur noch eine Frage der Zeit, bis der Host unter totaler Kontrolle ist.

[Windows About 2006]

Aber auch die hauseigenen Anwendungen von Microsoft sind ein enormer Risikofaktor in diesem Belang. Dies liegt vor allem daran, weil die vermeintlich anwenderfreundlichen Applikationen eine Vielzahl an Hintergrund-Aufgaben erledigen, die wiederum zusätzliche Rechte erfordern. Dies reicht vom Mitlesen der Inhalte anderer Fenster bis hin zur dynamischen Manipulation sensitiver Registry-Einträge. Gerade wenn sich ein Angreifer den Microsoft Internet Explorer, Word oder Excel einverleiben konnte, ist es um das Citrix-System geschehen.

Ich möchte an dieser Stelle ein etwas komplexeres Beispiel eines solch verketteten Angriffs nennen, das diesen Umstand illustrieren können wird. Meine Aufgabe war es, den Citrix-Zugriff einer international tätigen Privatbank zu überprüfen. Den Nutzern wurde dabei lediglich der Internet Explorer angeboten, um auf „sichere“ Weise im Internet surfen zu können. Durch einen Directory Traversal-Zugriff war ich in der Lage, Notepad zu starten. Im Speichern-Dialog dessen konnte ich eine Kopie von command.com in meinem privaten Verzeichnis anlegen, in dem ich die Rechte zum Ausführen von Programmen hatte. Da diese Instanz der Kommandozeile mit administrativen Privilegien und ohne Einschränkungen initiiert wurde, konnte ich nun auf jede Datei auf dem System zugreifen und nach Belieben Applikationen starten. Der Citrix-Host war nun unter meiner kompletten Kontrolle.

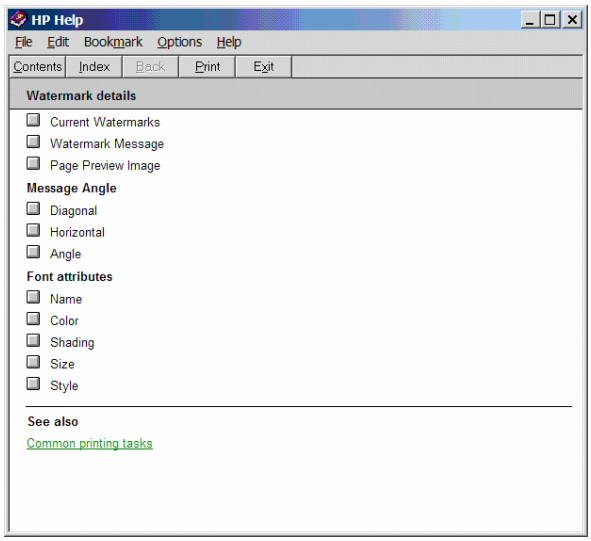

Abbildung 11: Erweiterte Rechte über Hilfe-Funktionen sind ein beliebter und altbekannter Weg.

Besonderes Interesse vererbter Rechte geniessen vor allem Hilfe-Funktionen, die bei der Rechtelimitierung gerne durch Administratoren vergessen oder übergangen werden. Unter Umständen werden eigene Hilfe-Utilities eingesetzt und diese wiederum erlauben das manuelle Öffnen von Dateien. Oftmals muss man Umwege über verschiedene Features machen, bis man eine Hilfe-Funktion findet, die auch wirklich von Nutzen ist – Dann springt man halt von Hilfe zu Hilfe. Vor allem die Hilfe-Anzeigen von Drucker-Einstellungen sind dafür bekannt, dass sie auf proprietäre und schlecht geschriebene Hilfe-Utilities zurückgreifen. Aber selbst die Microsoft Windows Help 5 erlaubt so manche Rechteausweitung.

Rechte dank Tastaturkombinationen

Wie weiter oben erwähnt, sind Administratoren darum bemüht, dass in einer Citrix-Umgebung möglichst nur jene Ressourcen zugänglich sind, die für den Benutzer von Wichtigkeit sind, damit dieser die ihm aufgetragenen Arbeiten erledigen kann. Dies führt dazu, dass oftmals lediglich einige wenige Applikationen, wie zum Beispiel Microsoft Excel oder eine Anwendung zur Buchhaltung, freigegeben sind. Das grösste Ziel für einen Angreifer in einer solchen Umgebung ist und bleibt dabei der gesamte Desktop-Zugriff. Hat er diesen erreicht, steht ihm die komplette interaktive Umgebung, wie man sie von einer lokalen Windows-Sitzung her kennt, zur Verfügung.

Oftmals kann der Weg zum Desktop über verschiedene Stationen führen. Aus Dialogboxen heraus können andere Anwendungen geöffnet werden, bis sich irgendwann der Taskmanager oder gar explorer.exe initialisieren lässt. Oftmals geht es jedoch viel einfacher und der Angreifer kann sich diesen Weg sparen. Stattdessen greift er auf altbekannte Tasten-Kombinationen zurück, die das Aufstarten weiterer Applikationen erlauben. Bestes Beispiel ist durch die Taste F1 gegeben, die in den meisten Anwendungen die Hilfe startet. Oftmals erlaubt der Start dieser zusätzlichen Anwendung das Erweitern der Rechte enorm. Vor allem dann, wenn sich damit nach Belieben andere Dateien (z.B. HTML-Dokumente), die sich eventuell gar noch vorgängig selber erstellen lassen, öffnen lassen.

Eine weitere Tasten-Kombination, die in dieser Hinsicht immerwieder für Angreifer von Interesse ist, ist im Drücken der Windows-Taste und der Taste E gegeben. Dies öffnet unverzüglich und

„Oftmals muss man Umwege über verschiedene Features machen, bis man ein Türchen findet.“

praktisch aus jeder Umgebung heraus, den Explorer, mit dem sich fortan sehr einfach Dateizugriffe umsetzen lassen. Ebenfalls einen Schritt weiter geht die Tasten-Kombination Shift+Ctrl+Esc, mit der sich nach Belieben eine Instanz des Taskmanagers ausführen lässt. Aus diesem Heraus lassen sich wiederum ganz einfach Prozesse beeinflussen oder neue mittels File/New Task starten.

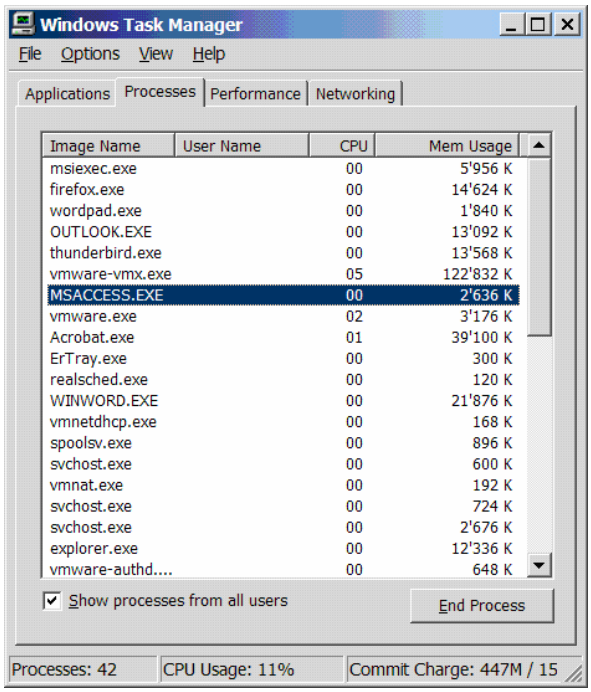

Abbildung 12: Der Windows-Taskmanager ist hervorragend für System-Auswertungen und das Starten von Programmen geeignet.

Skripting als Möglichkeit oder Sprungbrett

In manchen Fällen kann es gegeben sein, dass den Benutzern über Citrix eine gewisse Entwicklungs-Umgebung zur Verfügung steht. Im Idealfall ist dies natürlich ein Paket wie Microsoft Visual .NET, bei dem auf sämtliche wichtigen Komponenten der Windows-Programmierung zurückgegriffen werden kann. In einem solchen Fall könnte ein Angreifer nun eine Anwendung schreiben, die ihm gewisse System-Zugriffe umsetzt. Wird diese nun nicht oder nicht korrekt von ihrem Start (es muss sich dabei ja noch nicht einmal um eine kompilierte Fassung handeln) abgehalten, kann eventuell durch die Vererbung von Rechten ein erweiterter Zugriff umgesetzt werden. Einfaches Beispiel ist eine Umgebung, in der lediglich ein paar ausgewählte Programme gestartet werden können sollen. So darf Beispielsweise alles gestartet werden, was sich notepad.exe nennt. Ein Angreifer könnte nun seine Applikation als solche Kompilieren lassen, um von der Fehlenden Restriktion profitieren zu können. Ein solches Szenario ist aber nur in speziellen Einzelfällen von Erfolg gekrönt.

Die Entwicklung eigener Skripte oder Programme lässt sich aber auch auf einer primitiveren Ebene umsetzen. So kommt beispielsweise die Office-Produktreihe von Microsoft mit der Möglichkeit von aktivem Skripting daher. So lassen sich in Microsoft Word, Excel und Access eigene VBA-Skripte schreiben (Visual Basic for Applications), mit denen unter Umständen sicherheitskritische Zugriffe initiiert werden können. Ist die Mutter-Applikation nämlich einmal auf einem Citrix-System freigegeben, werden unter Umständen ihre erweiterten Zugriffe nicht mehr so genau überprüft. Solide Programmierkenntnisse für die Umsetzung eines solchen Angriffs sind natürlich Pflicht.

Noch eine Stufe primitiver, aber in einigen Situationen nicht minder erfolgreich, kann das Schreiben von Batch-Skripten sein. Durch einen herkömmlichen Editor lassen sich kleine Programme schreiben, mit denen Auswertungen oder gar als Angriff klassifizierbare Zugriffe automatisieren lassen.

Systemnahe Exploits von Anwendungen

Wir haben bisher sehr viele manuelle Überprüfungen, die sich grundsätzliche, konzeptionelle oder administrative Schwachstellen einer Umgebung zunutze machen, betrachtet. Dies sind mit Sicherheit jene Schwachstellen, die explizit in Multiuser-Umgebungen – wie Citrix nun mal auch eine ist – angetroffen wird. Alternative Herangehensweisen sind im Heranziehen spezifischer Exploits zu finden. Dabei versucht ein Angreifer, eine spezifische Schwachstelle in einer Anwendung auszunutzen, um erweiterte Rechte zu erlangen. Wir kennen dies vor allem in Netzwerkumgebungen, in denen ein Angreifer beispielsweise eine Pufferüberlauf-Schwachstelle in einem Apache-Webserver ausnutzt, um administrative Privilegien zu erschleichen. Das gleiche Prinzip kann auch auf Hostsicherheit angewandt werden.

„Die Entwicklung eigener Programme lässt sich auch mit primitiven Mitteln umsetzen."

Es würde nun den Rahmen dieser Arbeit sprengen, würde ich auf alle Angriffstechniken bis ins Detail eingehen und sämtliche Software, die von dieser betroffen ist, aufzählen. Stattdessen möchte ich lediglich einige exemplarische Bei-

spiele anfügen, wie derartige „lokale“ Exploitings aussehen können.

Von besonderem Interesse für einen Angreifer sind hierbei Bereiche des Betriebssystems. Können solche übernommen oder zu einer nützlichen Aktion bewegt werden, kann oftmals mit einer enormen Rechteausweitung gerechnet werden. Beispielsweise verspricht die Übernahme eines Treibers SYSTEM-Rechte, welche wiederum die totale Kontrolle des Hosts versprechen. Gerade weil die Betriebssystemreihe von Microsoft so populär ist, werden auch besonders viele Schwächen im Kern gefunden. Dabei liegt der Fokus der Hacker-Szene jedoch eher im Netzwerk-Bereich, da derartige Fehler eine netzübergreifende Übernahme eines Hosts versprechen. Wohingegen lokale Schwachstellen eher in Ausnahmefällen wirklich von Bedeutung sind (z.B. wenn der Hosts schon kontrolliert wird oder bei legitimen Remote-Zugriffen).

Viele Administratoren übersehen bei dieser Problematik, dass Netzwerkteile ebenfalls dem Risiko eines lokalen Angriffs ausgesetzt sind. Eine Remote-Verbindung lässt nämlich nicht selten den weiteren Zugriff auf localhost (127.0.0.1), also das eigene System, zu. Diese Loopback-Verbindungen können sodann über abgehörte Ports stattfinden, wie eventuell durch eine Firewall geschützt sind. Lokale Zugriffe werden aber durch dedizierte Firewall-Elemente gar nicht tangiert und hostbasierte Lösungen verhindern ebenfalls sehr selten Verbindungen über die Loopback-Schnittstelle. Nicht zwingend benötigte Standarddienste wie Webserver (HTTP, tcp/80), FTP-Server (tcp/21), Mailserver (SMTP, tcp/25; POP3, tcp/110) sind da gern gesehene Ziele. Aber ganz besonders sind dies auch die NetBIOS-/SMB-Dienste, wie sie überlicherweise in Windows-Umgebungen angeboten werden. Die klassischen TCP-/UDP-Portbereiche wie 135, 137 bis 139 und 445 bleiben also noch immer kritischer Natur.

Dieser Umstand bleibt brisant, auch wenn er sich wohl bisher nur für die wenigsten so abgezeichnet hat. Nehmen wir als Beispiel einen Citrix-Server, der in einer von allen Seiten mittels Segmentierung, Routing und Firewalling sehr gut abgeschotteten Zone befindet. Ein Nutzer, der von der Remote-Verbindung über Citrix legitimen Zugriff hat, kann diese netzwerkbasierten Schutzmassnahmen sehr wohl umgehen. Schafft er es nämlich, auf das Zielsystem einen Exploit für die SMB-Implementierung von Microsoft Windows einzuschleusen, kann er diesen Angriff lokal auf dem Host selbst und dabei fernab der Schutzmassnahmen der Firewalls initiieren. Ein fehlender Patch kann sich in dieser Situation plötzlich als verheerend herausstellen.

Kehren wir zurück zu den auf dem Host installierten Anwendungen, über die sich mittels dedizierten Exploits erweiterte Rechte erlangen lassen. Hierbei sind vor allem Applikationen im Fokus, welche für ihre Tätigkeiten möglichst viele Privilegien, am besten administrativer Natur, zunutze machen müssen. In dieser Hinsicht werden gerne Personal Firewalls und Antiviren-Lösungen genannt. Derlei Software nistet sich sehr tief im System ein und entsprechend kann eine Übernahme einer solchen Komponente sehr viel versprechend sein [scip AG 2005].

Gerade Antiviren-Lösungen sind besonders anfällig, wenn es um Attacken auf das Dateisystem geht. Da derartige Produkte in der Regel die Zugriffe auf Dateien überwachen, können manipulative Bestrebungen direkten Gewinn ermöglichen. Ein einfaches Beispiel ist das Scannen eines Verzeichnisses auf mögliche Infektion durch Computerviren. Bietet die eingesetzte Antiviren-Lösung nun die Möglichkeit an, dass man sich mit einem „Explorer“ durch das Dateisystem navigieren kann, würde ein Angreifer als erstes überprüfen, ob man mit diesen Explorer nun auch explizite Dateimanipulationen (z.B. Öffnen, Kopieren, Verschieben, Löschen) anstellen kann. Falls dem so ist, dann könnte es durchaus sein, dass er erweiterte Rechte der Applikation erbt, die beispielsweise Zugriffe auf Bereiche und Dateien ermöglicht, für die er sonst gar keine Privilegien gehabt hätte. In einem solchen Fall liesse sich sehr schnell das System durch gezielte Eingriffe unter Kontrolle bringen.

Portscanning mittels Netzwerkanwendungen

Citrix-Server stehen meistens in einer dedizierten Zone, die mittels Segmentierung und Firewalling vor unerlaubten Zugriffen geschützt wird. Oftmals ist dies aus Kostengründen die gängige DMZ des Unternehmens, in der sich auch noch andere Server-Systeme finden. Die Administratoren dieser Umgebung verhindern so mit dem gleichen Sicherheitsdispositiv, dass die sensitiven Systeme direkt angesprochen oder angegriffen werden können.

„Ein Remote-System, wie Citrix auch eines ist, kann als Hopping-Host missbraucht werden.“

Dabei wird jedoch gerne aus den Augen verlo-

ren, dass ein Remote-System, wie es halt eben durch Citrix gegeben ist, als Hopping-Host missbraucht werden kann. In diesem Fall werden etwaige Angriffe auf das Zielsegment direkt über das Citrix-System initiiert, welches sich in der gleichen Zone befindet und deshalb ein Mehr an Rechten geniesst.

Derlei Hopping-Angriffe können grundsätzlich mit jeder ausführbaren netzwerkfähigen Applikation, die die Angabe des Ziels ermöglicht, umgesetzt werden. Populär für diesen Angriff, da in den meisten Windows-Umgebungen vorhanden, sind die folgenden beiden Elemente [Ruef 2004]:

- C:\WINDOWS\system32\telnet.exe; Dies ist der Standard-Telnet-Client moderner Windows-System. Ist er einmal initiiert, können mit ihm beliebige TCP-Zugriffe umgesetzt werden. Ist der Zielport offen, wird die Ausgabe dessen angezeigt. Andernfalls weist der Client eine Fehlermeldung aus und informiert über den entsprechenden Port-Status des Ziels.
- C:\WINDOWS\explorer.exe; Dies ist sowohl der grafische Dateimanager als auch der Standard-Webbrowser (Microsoft Internet Explorer) von Windows-Systemen. Jenachdem ob eine lokale Ressource oder eine Netzwerk-URL zum Einsatz kommt, verändert er sein Aussehen und die Funktionalität. Durch das Ansteuern von URLs samt Heranziehen dedizierter Zielports (z.B. http://192.168.0.1:25) können die jeweiligen Port-Stati determiniert werden. Ist der Port offen und eine Verbindung möglich, wird diese hergestellt. Andernfalls wird eine Fehlermeldung ausgegeben, die den Port als geschlossen bzw. nicht erreichbar deklariert.

Damit sind aber nicht nur Angriffe auf andere Systeme denkbar. Ebenso liessen sich lokale Netzwerk-Attacken umsetzen, indem simpel auf die Loopback-Adresse (127.0.0.1) oder die eigene IP-Adresse des Citrix-Servers zugegriffen wird. Nur in den wenigsten Fällen verhindern Administratoren derlei Zugriffe mittels spezifischer ACLs auf transparenten Proxies oder über anderweitige Konfigurations-Einstellungen. Und selbst dann können altbekannte Angriffe auf Proxy-Elemente genutzt werden, um die Limitierungen zu umgehen [Ruef et al. 2002]. Derlei Angriffstechniken sind jedoch nicht Teil dieser Abhandlung.

Erweiterte Rechte: Was nun?

Das Ziel dieser Dokumentation ist das Festhalten der Angriffsmöglichkeiten, die ein legitimer Citrix-Nutzer, der nur auf einzelne freigegebene Applikationen zugreifen kann, hat. Einige der möglichen Techniken zur Rechteausweitung in diesem Kontext wurden besprochen. Doch nun stellt sich die Frage, was ein Angreifer macht, sobald er seine erweiterten Rechte erlangt hat.

Zuerst muss ich bemerken, dass die Ziele eine Angreifers unterschiedlicher nicht sein könnten. Es gibt solche, die brechen in Systeme ein, um Daten zu stehlen. Andere sind lediglich an der Funktionsweise der eingesetzten Technologien und ihrer Sicherheitsmerkmale interessiert. Wieder andere wollen von den brach liegenden Ressourcen grosser Konzerte profitieren. Es ist also grundsätzlich unmöglich, hier eine umfassende Darlegung der Wünsche und endgültigen Ziele eines Angreifers festzuhalten. [Ruef 2001]

Einzig die generische Aussage, dass sich ein Angreifer eines Citrix-Systems genau gleich verhalten wird, wie in einer anderen Umgebung, geniesst annehmbare Gültigkeit. Citrix ist im Grunde lediglich eine Möglichkeit, wie sich auf ein System verbunden und auf diesem gearbeitet werden kann. Da entsprechend keine für den Endanwender nach der Etablierung der Sitzung offensichtlichen Unterschiede bemerkt werden können, wird es auch keine Unterschiede in Bezug auf das Verhalten eines Angreifers geben.

Die Rechteausweitung geniesst in der Regel sehr hohe Priorität. Erst wenn das gesamte System unter Kontrolle ist, sind die meisten Angreifer zufrieden. Dies bedeutet, dass sie sich administrative Privilegien zuweisen können müssen. Sobald dieses Ziel erreicht ist, können die weiteren Phasen eines Angriffs initiiert werden. Diese sind im Allgemeinen die Folgenden:

- Daten entwenden
- Ressourcen nutzen
- Hintertüren einbringen
- Spuren verwischen

„Erst wenn das gesamte System unter Kontrolle ist, sind die meisten Angreifer zufrieden.“

Schutzmassnahmen für Administratoren

Grundsätzlich muss man sagen, dass das Absichern einer Citrix-Umgebung ein enorm schwieriges und umfangreiches Unterfangen darstellt [Dannbacher et al. 2002]. Einer der Gründe ist, dass die Windows-Betriebssystemreihe ur-

sprünglich nicht für Multiuser-Szenarien entwickelt wurde. Zwar wurden grobe Absätze mit einer dedizierten Anmeldung bei Windows 9x angefangen und autonome Benutzer-Umgebungen mit Windows NT angestrebt. Das gleichzeitige Arbeiten mehrerer Benutzer auf einem Windows-System ist aber nach wie vor eine Besonderheit. Diese historische Limitierung konzeptioneller und technischer Natur verhindert es, dass der gleichzeitige Nutzen sicher und effizient betrieben werden kann. Citrix ist und bleibt lediglich ein Aufsatz auf ein Pseudo-Multiuser-System – Mehr nicht.

Die andere grosse Hürde sicherer Citrix-Umgebungen ist der enorme Umfang einer Installation. Diese erhöhte Komplexität macht es administrativ ungemein schwer, eine Plattform sicher und effizient zu installieren und warten. Es gibt solch eine Vielzahl an potentiellen Angriffsflächen, denen sich ein Administrator bewusst sein muss. Diese alle zu minimieren und damit ein Höchstmass an Sicherheit zu erreichen ist schier unmöglich. Sehr solide Kenntnisse im Windows-, Citrix- und Netzwerk-Umfeld (inkl. Firewalling und IPsec) sind erforderlich, um dieses erstrebenswerte Ziel zu erreichen.

Eine sichere Citrix-Installation erfordert grundsätzlich, dass das genutzte Betriebssystem ebenfalls möglichst sicher ist. Dies bedeutet, dass das genutzte Windows so abgespeckt wie möglich installiert werden sollte. Unnötige Software-Pakete und Dienste stören nur. Sie können neben Performance-Einbussen die Sicherheit der gesamten Umgebung gefährden. Entsprechend installiert man sie am besten gar nicht, deinstalliert sie nachträglich oder deaktiviert sie wenigstens.

Des Weiteren gilt es den Host mit den jeweiligen Patches und Bugfixes auf dem neuesten Stand zu halten. Die zyklischen Patchdays von Microsoft sollte man möglichst zeitnah untersuchen, um die neuen Schwachstellen so schnell wie möglich zu beheben. Kritische Fehler wie Pufferüberlauf-Schwachstellen in System-Komponenten sollten bestmöglich – wenigstens aus Sicht der Sicherheit – sofort nach Erscheinen eines Patches installiert werden. Leider ist dies oftmals aufgrund der Höherstellung der Betriebsanforderungen nicht oder nur bedingt möglich. In solchen Fällen muss man mit grossen und damit eben kritischen Zeitfenstern für erfolgreiche Attacken leben. Administratoren tun in diesem Fall gut daran, ihre Vorgesetzten oder gar das Management schriftlich über den Umstand zu informieren, damit diese anhand einer Risiko-Analyse das weitere Vorgehen (Patch installieren, Komponente deaktivieren, System abstellen oder abwarten) absegnen können.

Die Sicherheit eines Citrix-Servers geht aber noch ein Stückchen weiter, weder man sich das sonst von einer Windows-Workstation oder –Server gewohnt ist. Da sich mehrere Benutzer (zeitgleich) auf dem Host bewegen werden und diese unter Umständen als nicht vertrauenswürdig eingestuft sind, müssen lokale Komponenten ebenfalls vor unerwünschten Zugriffen geschützt werden. In solchen Fällen lohnt es sich sodann überproportional Zugriffsberechtigungen ebenfalls für das Dateisystem (NTFS) und die Registry umzusetzen. Über Group Policies lassen sich die einzelnen Konten zusätzlich beschneiden, so dass bestimmte Funktionen und System-Aufrufe gar nicht zur Verfügung stehen. Ist dies strikt umgesetzt, wird es für einen Angreifer enorm schwierig, sich seinen Weg durch das System zu bahnen.

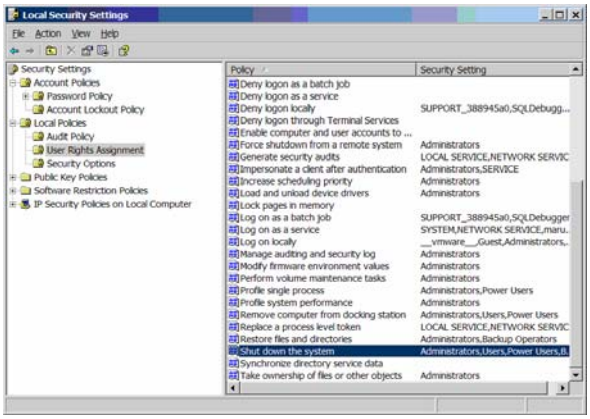

Abbildung 13: Policies sind eine der wichtigsten Waffen für Administratoren, um Angriffen einen Riegel vorzuschieben.

Zusätzliche hostbasierte Sicherheits-Installationen sind ebenfalls empfehlenswert [Ruef 2002]. Durch Personal Firewalls oder spezifische IPsec-Filter können unerwünschte Zugriffsversuche – auch über die Loopback-Schnittstelle – verhindert werden. Über das lokale Netzwerk initiierte Attacken lassen sich so abwehren (z.B. mit „telnet localhost 25“). Können die Anwender eigene Dateien herunterladen oder Programme installieren, lohnt sich die Installation einer Antiviren-Lösung. Diese sollte ebenfalls auf dem neuesten Stand gehalten werden und im Mindesten die ausführbaren Dateien nach korruptem Programmcode untersuchen.

Konnte ein annehmbares Mass an Grundschutz auf der Betriebssystem-Ebene erreicht werden, kann man sich um die Sicherheit von Citrix an sich kümmern. Es sollen stets so wenige Applikationen wie möglich freigegeben werden, um den Handlungsspielraum für Auswertungen und die Angriffsfläche für spezifische Attacken so

klein wie möglich zu halten.

Fazit

Citrix erlaubt es, ein Windows-System um wichtige Funktionalität zu bereichern. Sogleich wird es möglich, dass verschiedene Benutzer zeitgleich auf ihre Oberfläche auf einem System zugreifen können. Durch diesen zentralisierten Ansatz können Vorteile (aber auch Nachteile) für sich gewonnen werden, die im hart umkämpften Alltag eine entscheidende Rolle spielen können.

Wir haben aber gesehen, dass die Zugriffe der Benutzer nicht immer als sicher und vertrauenswürdig taxiert werden dürfen. Durch verschiedene Methoden ist es nämlich möglich, auf einem Citrix-System erweiterte Rechte zu erlangen. Einige Features erlauben es quasi eine komfortable Suche auf der Dateisystem-Ebene umzusetzen.

Wiederum andere Funktionen machen es möglich, dass andere Anwendungen gestartet und auf ansonsten verborgene Systembereiche zugegriffen werden kann. Vor allem der Missbrauch von Dialogboxen und das Heranziehen von (automatischen) URI-Links sind eine einfache und effiziente Methode, um seine Rechte auszuweiten. Aber auch spezifische Teile des Betriebssystems können direkt oder indirekt angesprochen werden, um damit weitere Auswertungen und Angriffe tätigen zu können. Insbesondere systemnahe Komponenten wie Dienste und Antiviren-Software sind vielversprechende Kandidaten, wenn es um die Suche nach dem Heiligen Gral geht.

Ein Angreifer auf einem Citrix-System verhält sich insgesamt nicht viel anders weder ein Angreifer aus dem Internet oder auf einem UNIX-Host auch [Ruef 1999, McClure et al. 2005]. (1) Er wird als erstes versuchen Informationen zur Umgebung zusammenzutragen, um dank dieser Auswertung die Angriffe möglichst zielgerichtet und effizient anstreben zu können. (2) Hat er potentielle Schwachstellen gefunden, wird er sie für seinen Vorteil ausnutzen wollen. (3) Ist ihm dies gelungen, wird er entweder nach weiteren Rechten Ausschau halten oder sich gleich eine Hintertür einrichten. (4) In einer letzten Phase geht es ihm darum die bisher hinterlassenen Spuren, die auf einen Angriff bzw. Einbruch hindeuten könnten, zu verwischen.

Auch in Fällen der Kompromittierung einer Citrix-Umgebung gilt: Ein einmal kompromittiertes System zu 100 % von gänzlichen Hintertüren zu befreien ist schier unmöglich – Eine Neuinstallation ist oftmals der kürzere und schnellere Weg, steht aber meist im Widerspruch zu den betrieblichen Aspekten.

Obschon in der Welt von Citrix gewisse Gefahren gegeben sind, stehen ihnen die Administratoren nicht gänzlich machtlos gegenüber. Durch das solide Absichern des Betriebssystems an sich, das möglichst zeitnahe Einspielen von Patches, das Installieren von zusätzlicher Sicherheits-Software (z.B. Personal Firewall), das Umsetzen von Group Policies und das Nutzen der Sicherheitsfunktionen von Citrix selbst kann ein gutes Mass an Sicherheit erlangt werden. Dies erfordert jedoch sehr viel Zeit und Wissen, das wohl nur die wenigsten Administratoren im alltäglichen Betrieb aufzubringen in der Lage sind.

„Durch den zentralisierten Ansatz von Citrix werden nicht nur Vorteile für sich gewonnen."

Gerade deshalb ist es umso wichtiger, in sicherheitskritischen Umgebungen nicht auf das Heranziehen von Spezialisten zu verzichten. Eine Begleitung während des Aufbaus und das kontinuierliche Überprüfen der Installation auf mögliche Schwachstellen hilft dabei, potentielle Mängel frühzeitig ausmachen und beheben zu können [Ruef 2003]. Citrix ist und bleib ein gewaltiges Monster, das man aber mit genügend Mut und Disziplin zu bändigen in der Lage ist.

Literaturverzeichnis

Dannbacher, André, Kienle, Fabian, Oktober 2002, Leitfaden für sichere Citrix-Systeme, http://www.computec.ch/download.php?view.344

Hoglund, Greg, McGraw, Gary, Februar 2004, Exploiting Software – How to break Code, Addison-Wesley Professional, ISBN 0201786958, http://www.amazon.de/exec/obidos/ASIN/0201786958/

McClure, Stuart, Scambray, Joel, Kurtz, Georg, November 2005, Das Anti-Hacker-Buch, Vmi Buch, ISBN 3826681673, http://www.amazon.de/exec/obidos/ASIN/3826681673/, englische Original-Ausgabe erschienen mit dem Titel „Hacking Exposed“

Microsoft, 20. Januar 2005, Für das FAT32-Dateisystem geltende Beschränkungen, Microsoft.de, http://support.microsoft.com/kb/184006/de

Microsoft, 17. März 2004, SO WIRD'S GEMACHT: Verwenden des Tools "Application Security", um den Zugriff auf Programme in den Terminaldiensten von Windows 2000 einzuschränken, Microsoft, http://support.microsoft.com/kb/320181/de

Ruef, Marc, 1999, Die Sicherheit von Windows, astalavista.com und computec.ch, http://www.computec.ch/download.php?view.283

Ruef, Marc, 22. Dezember 2001, Die psychosozialen Aspekte der Computerkriminalität, computec.ch, http://www.computec.ch/download.php?view.110

Ruef, Marc, 2002, Intrusion Prevention – Neue Ansätze der Computersicherheit, Computer Professional, Ausgabe 4-2002, Seiten 10-14, http://www.computec.ch/download.php?view.302

Ruef, Marc, 2003, Auditing mit Linux - Entdecken Sie die Schwachstellen Ihres Systems, computec.ch, http://www.computec.ch/download.php?view.528

Ruef, Marc, Februar 2004, Lehrgang Computersicherheit, Universität Luzern, Master of Advanced Studies eLearning und Wissensmanagement, http://www.computec.ch/download.php?view.481

Ruef, Marc, Rogge, Marko, Velten, Uwe, Gieseke, Wolfram, November 2002, Hacking Intern - Angriffe, Strategien, Abwehr, Data Becker, Düsseldorf, ISBN 381582284X, http://www.amazon.de/exec/obidos/ASIN/381582284X/

scip AG, 1. Juni 2005, ZoneLabs ZoneAlarm 5.x Vet engine bis 11.9.1 Vet Antivirus Engine VetE.dll OLE-Stream Pufferüberlauf, scip AG, http://www.scip.ch/cgi-bin/smss/showadvf.pl?id=1510

Windows About, 6. Februar 2006, Change Permissions for Task Scheduling, The New York Times Company, http://windows.about.com/library/tips/bltip696.htm

Der Autor

Marc Ruef
Security Consultant
+41 44 445 1812
mailto:maru@scip.ch

Marc Ruef arbeitet als Security Consultant bei der schweizer Firma scip AG und ist dort Leiter des Bereichs Security Auditing und Penetration Testing. Im Oktober 2002 ist im Data Becker Verlag sein zweites Buch mit dem Titel Hacking Intern (ISBN 381582284X) erschienen, dessen erste Auflage nach rund einem Jahr ausverkauft war. Im Februar 2004 erschien über den Hüthig Telekommunikation Verlag seine deutsche Übersetzung des englischen Klassikers Network Intrusion Detection von Stephen Northcutt und Judy Novak (ISBN 3826609743).

Neben einer Vielzahl von Fachpublikationen zur theoretischen Informatik und IT-Security (über 200 Stück, Stand Januar 2006) unterstützt er diverse internationale Projekte aus diesen Bereichen. Seit 1997 betreut er die Webseite computec.ch, die als grösstes Archiv deutschsprachiger Publikationen zum Thema Informationssicherheit gilt. Des Weiteren ist er Entwickler des Attack Tool Kit (ATK), einem open-source Exploiting Framework, das als Ergänzung zu Lösungen wie Nessus generische Penetration Tests einfacher und transparenter macht. Zudem ist er CoreHacker des open-source CMS e107 und ein Mitglied des OWASP Switzerland Chapter (Open Web Application Security Project). Nebenbei ist er an verschiedenen Hochschulen und Universitäten als Dozent für Computersicherheit tätig.

Der Herausgeber

scip AG
Technoparkstrasse 1
CH-8005 Zürich
+41 44 445 1818
mailto:info@scip.ch
http://www.scip.ch

scip AG ist eine unabhängige Aktiengesellschaft mit Sitz in Zürich. Seit der Gründung im September 2002 fokussiert sich die scip AG auf Dienstleistungen im Bereich IT-Security. Unsere Kernkompetenz liegt dabei in der Überprüfung der implementierten Sicherheitsmassnahmen mittels **Penetration Tests** und **Security Audits** und der Sicherstellung zur Nachvollziehbarkeit möglicher Eingriffsversuche und Attacken (**Log-Management** und **Forensische Analysen**). Vor dem Zusammenschluss unseres spezialisierten Teams waren die meisten Mitarbeiter mit der Implementierung von Sicherheitsinfrastrukturen beschäftigen. So verfügen wir über eine Reihe von Zertifizierungen (Solaris, Linux, Checkpoint, ISS, Cisco, Okena, Finjan, TrendMicro, Symantec etc.), welche den Grundstein für unsere Projekte bilden. Das Grundwissen vervollständigen unsere Mitarbeiter durch ihre ausgeprägten Programmierkenntnisse. Dieses Wissen äussert sich in selbst geschriebenen Routinen zur Ausnutzung gefundener Schwachstellen, dem Coding einer offenen Exploiting- und Scanning Software als auch der Programmierung eines eigenen Log-Management Frameworks. Den kleinsten Teil des Wissens über Penetration Test und Log-Management lernt man jedoch an Schulen – nur jahrelange Erfahrung kann ein lückenloses Aufdecken von Schwachstellen und die Nachvollziehbarkeit von Angriffsversuchen garantieren.