Schwachstellen in Windows-Systemen

Basierend auf dem Artikel aus der letzten Ausgabe des Maschinenbaus: «Der Penetration Test» wollen wir uns diesmal mit den möglichen Schwachstellen von Betriebssystemen und Web-Anwendungen auseinandersetzen. Sobald die vorgängig ausgeführten Untersuchungen Schwachstellen anzeigen, gilt es, diese genauer zu untersuchen.

Obwohl dem Betriebssystem von Microsoft oft schlechtes nachgesagt wird, kann es doch sehr sicher betrieben werden. Die Erfahrung zeigt, dass die meisten Schwachstellen durch installierte Dienste verursacht werden. Jeder installierte Dienst erhöht die Angriffsfläche eines Systems. Zudem gilt, dass das Patchen oft vernachlässigt und dadurch einem potenziellen Angreifer die Arbeit unnötig erleichtert wird. Wenn dann in einer Firewall zu viele Ports geöffnet sind, wird die Gefahr einer erfolgreichen Attacke erhöht. Wie bereits erwähnt, gibt der Portscan die ersten Informationen auf mögliche Schwachstellen bekannt. Sind typische Windows-Ports offen, wie Kerberos (88), RPC und Netbios (135 bis 139), LDAP (389), SMB/CIFS (445), SQL Server (1433) und Terminal Services (3389) kann eine genauere Analyse weitere Informationen liefern. Hier lohnt sich der Einsatz eines umfassenden Scanners wie Nessus oder der GFI Languard Network Security Scanner. Sind Lücken in den eingesetzten Versionen bekannt, geben dies die beiden Programme an. Nun muss nur noch im Internet das entsprechende Angriffstool gefunden werden. Google hilft hier sicherlich am schnellsten weiter, jedoch auch spezialisierte Seiten wie PacketStorm liefern oft ein passendes Programm.

Kann ein Zugriff auf die Anmeldeseite des Betriebssystems aufgebaut werden, sollten zuerst die möglichen Ziele identifiziert werden. In der Regel werden Accounts nach einer gewissen Anzahl Fehlversuchen gesperrt. Davon ausgenommen ist jedoch immer der Administrator. Der Administrator ist aber oft kein ideales Angriffsziel, da er (hoffentlich) gut überwacht wird. Ein Tool, das bei der Suche nach einem geeigneten Angriffsziel weiterhelfen kann, ist enum. Es nützt die Möglichkeit von Windows aus, sich mittels einer NULL-Session auf einen Server zu verbinden. Damit ist es möglich, ohne Benutzernamen und Passwort einige Details abzufragen. Ist ein geeigneter Benutzer gefunden, erfolgt eine sogenannte Brute-Force-Attacke. Hier werden alle möglichen Kombinationen von Buchstaben, Zahlen und Sonderzeichen durchprobiert, bis das Passwort gefunden ist.

Immer wieder geschieht es, dass für die Serveradministration der Port 3389 (Terminal Services) auch durch die Firewall hindurch geöffnet wird. So kann zum Beispiel der externe Support jederzeit auf den Server zugreifen. Dieses grobfahrlässige Vorgehen öffnet einem Hacker einen optimalen Zugang zum System. Im Internet sind Tools verfügbar, die Attacken auf genau diesen Port durchführen können. Die Empfehlung ist daher eindeutig: Terminal Services dürfen nur via VPN erreichbar sein.

Eine weitere Sünde ist der direkte Zugriff auf den SQL-Server. Sobald der Port 1433 offen ist, welcher durch den SQL Server geöffnet wird, kann eine Attacke gefahren werden. Ist das Passwort nach einer Weile geknackt (abhängig von der Länge und der Komplexität), stehen alle Daten der Datenbank zur Verfügung. Da sich hier in der Regel wichtige oder gar vertrauliche Daten befinden, muss dieser Zugriff konsequent unterbunden sein.

Schwachstellen in Unixsystemen

Auch unter Unix sind Schwachstellen offen. Typische Ports, die als Angriffsziel interessant sein können, sind: ssh (22), telnet (23), finger (79), exec (512), login (513) und shell (514).

Ist finger geöffnet, kann als erstes abgefragt werden, welche Benutzer gerade am System angemeldet sind. Das Tool hydra kann anschliessend analog dem Windowssystem ein Angriff auf das Passwort gestartet werden.

RPC basierte Dienste sind unter Unix ein beliebtes Angriffsziel. Daher liefert das Tool rpcinfo genauere Informationen. Mit diesem Aufruf stehen alle notwendigen Informationen für einen weiteren Angriff zur Verfügung. Um welches System handelt es sich? Welche Applikationen sind installiert? Sind Schwachstellen dafür bekannt?

Schwachstellen in Web-Anwendungen

Die Betriebssysteme werden immer sicherer. Daher verlagern

Bild: Archiv

ZUM AUTOR

Dipl.-Ing, FH, CISSP, Andreas Wisler
IT-Redaktor Maschinenbau
Geschäftsführer GO OUT
Production GmbH
Schulstrasse 11
CH-8542 Wiesendangen

Telefon +41 (0)52 320 91 20
www.goout.ch
info@goout.ch

sich die Angriffe auf «leichtere» Ziele. Zu den beliebtesten gehören Web-Anwendungen. Immer mehr Applikationen bieten einen zusätzlichen Zugriff via HTTP (oder über den verschlüsselten Pfad HTTPS). Leider ist es aufgrund der Limitationen des zugrunde liegenden Protokolls auch für erfahrende Programmierer nicht einfach, sichere Web-Anwendungen zu entwickeln.

Alle Daten werden als ASCII Text übermittelt (in beide Richtungen). Ein Abfangen und Senden ist daher nicht besonders schwer. Damit eine Webseite und die übertragenen Daten einfach untersucht werden können, lohnt sich der Einsatz eines lokalen Proxys. Dieser zeigt den kompletten Datenfluss übersichtlich an. Auch können damit die Daten vor dem Versenden anzeigt und modifiziert werden. Dies ist vor allem dann interessant, wenn in Formularen so genannte Hidden-Felder übermittelt werden. Leider verwenden viele Administratoren diese Felder um einen Benutzer eindeutig wiederzuerkennen. Ohne Aufwand kann dann aber der Benutzer verändert werden. Für die schnelle Manipulation stehen sogar Plugins für den Webbrowser Firefox zur Verfügung.

Trotz vielen Fachberichten und Warnungen gibt es immer noch Formularfelder, welche den eingegebenen Inhalt ungeprüft an Datenbanken weiterreichen. SQL Injection heisst dieses Angriffsszenario, welches versucht, diese Anfragen zu manipulieren und wahre Ausdrücke zu generieren. Somit können auch ohne Kenntnis der Datenbank (vertrauliche) Daten ausgelesen werden.

Eine neue Gefahr ist Cross Site Scripting. Hier werden jedoch nicht Daten ausgelesen, sondern fremder Code in die echte Seite eingeschleust. Wiederum geschieht dies über schlecht ausgewertete Formularfelder. Ob die Seite dafür anfällig ist, lässt sich leicht mit <script> Alert (XSS Test) </script> testen. Öffnet sich ein zusätzliches Fenster, ist die Seite anfällig auf Cross Site Scripting. Das gefährliche daran ist, dass sich ein Benutzer auf der richtigen Seite befindet, jedoch einen «falschen» Inhalt angezeigt bekommt. Werden so vertrauliche Informationen eingegeben, gelangen diese an den Angreifer und nicht an die Webseite.

Fazit

Dies sind nur drei mögliche Angriffspunkte. Jedes weitere Gerät, welches auf die Angriffsversuche reagiert, kann ein lohnenswertes Ziel sein. Daher gilt, nur das anzubieten, was auch wirklich benötigt wird! Bevor eine Applikation in den produktiven Betrieb geht, muss eine umfassende Kontrolle, idealerweise durch externe Personen, welche nicht in das Projekt involviert sind, durchgeführt werden. Viele Schwachstellen werden so frühzeitig erkannt und können geschlossen werden, bevor es zu einem «Unfall» kommt.