

21.01.10

Von: Julian Totzek-Hallhuber

## Web Application Security – der Logikfaktor

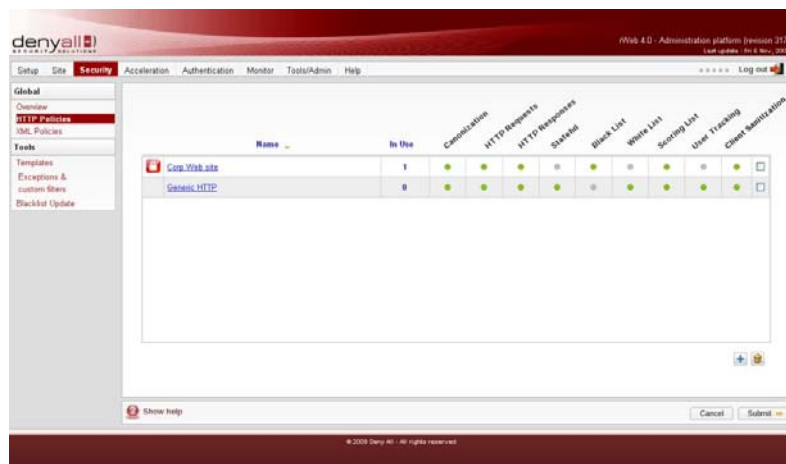


Webanwendungen werden heutzutage nicht nur durch klassische SQL Injections oder Cross Site Scripting angegriffen. Selbst wenn SQL Injections oder XSS in den meisten Fällen die deutlich komplexeren und schwierigeren Angriffe sind, können diese auf die Logik der Applikation mindestens genauso, wenn nicht sogar weitaus gefährlicher sein.

Somit muss auch die Logik einer Applikation entsprechend abgesichert werden, wenn vollständiger Schutz gewährleistet werden soll.

Ein Angriff auf die Logik einer Applikation erfordert in den seltensten Fällen tiefe technische Kenntnisse. Meist reicht einfaches Analysieren aus, um einen Fehler in der Logik zu entdecken.

Die Logik lässt sich oft schon durch Ändern eines einzigen Parameters oder der URL überlisten. Moderne WAF-Systeme müssen auch vor solchen Bedrohungen schützen können. In den meisten Fällen sind die Regeln einer WAF sehr statisch definiert und selbst im positiven Security Model (Whitelisting) können solche Bedrohungen nicht adressiert werden. Denn eine Whitelist definiert jede URL und jeden dazugehörigen Parameter, wenn der Parameter aber nur innerhalb dieses Rahmens geändert wird, kann auch eine Whitelist diesen Angriff nicht mehr abfangen.



 Bildupload

Doch nicht nur das Ändern von Parametern kann die Logik überlisten. Werden zu viele vermeintlich legitime Requests gesendet, kann die Logik ebenfalls zerstört werden. Es muss sich dabei nicht immer um die Logik der Applikation handeln, Angriffe auf die IT-Security Prozesse sind genauso möglich.

Im Folgenden finden Sie Beispiele und Anregungen, auf solche Bedrohungen zu reagieren bzw. schon auf sie vorbereitet zu sein.

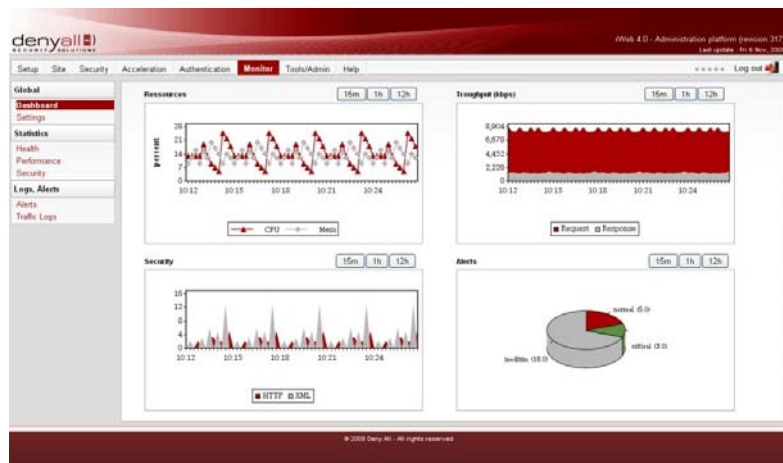
### E-Mail Web Access Account-locking

Bei dem dargestellten Angriff handelt es sich wohl um den einfachsten Angriff aus diesem Bereich. Er zielt nicht auf die Applikation selbst, sondern auf den von der IT-Security definierten Prozess. E-Mail Web Access-Systeme sind in fast allen Unternehmen zu finden. Mitarbeiter sollen auch von unterwegs die Möglichkeit haben, auf ihre E-Mails zugreifen zu können. Aber nicht nur Mitarbeiter, sondern auch jede andere Person kann auf dieses System zugreifen, sofern der Access nicht über andere Methoden wie z.B. SSL VPN geschützt ist.

Denn in fast allen Fällen sind beispielsweise Thresholds definiert, die nach einer bestimmten Anzahl von fehlerhaften Logins den Account sperren. Eine sicherlich nützliche Funktion, die aber auch Angreifer unterstützt. Kennt der Angreifer die Namenskonventionen des Unternehmens – z. B. vorname.nachname@company.de – kann er durch bewusste Falscheingabe des Passworts für diesen Benutzernamen den Account sperren, so dass der Mitarbeiter keinen Zugriff mehr auf seine E-Mails hat.

Dennoch handelt es sich bei Thresholds um eine nützlich Funktion, da so auch Brute Force-Angriffe auf die Passwörter der Accounts unterbunden werden können. **Doch wie kann beispielsweise gesichert werden, dass durch ein E-Mail Web Access Account-**

Locking dem Vorstand nicht gerade vor einem wichtigen Meeting, einer Pressekonferenz oder ähnlichem der Zugriff auf seine E-Mails gesperrt wird?

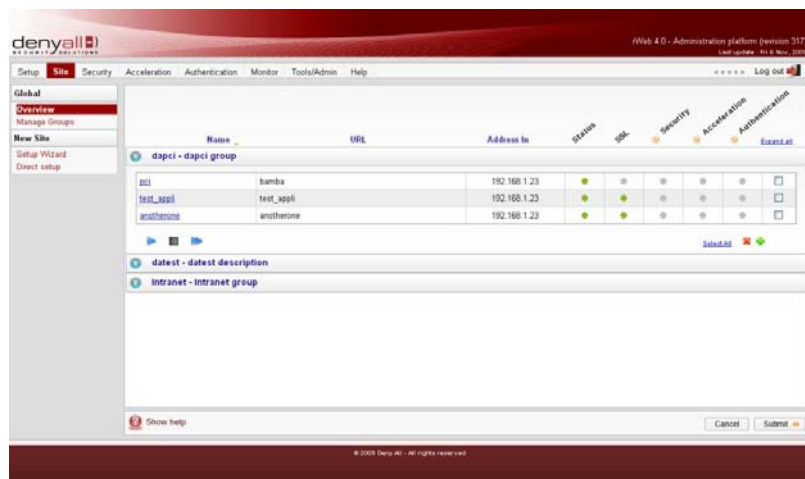


Bildupload

Die Antwort ist simpel, mit einer Web Application Firewall (WAF). Eine WAF muss auf solche Bedrohungen reagieren können. Am sinnvollsten ist an dieser Stelle ein zweistufiges Konzept zu implementieren, durch das sowohl Application Denial of Service als auch der eben beschriebene Angriff geblockt werden können. Einfache Application DoS Angriffe können so durch ein automatisiertes Erkennen von zu vielen Zugriffen von einer IP-Adresse sehr einfach und effektiv blockiert werden. Der oben beschriebene Angriff kann durch eine Modifikation bzw. Erweiterung eines solchen Regelsatzes verhindert werden.

## Einfaches Ändern eines Parameters

Auch die Adresszeile eines Browsers kann manipuliert werden. Oft werden wichtige und interessante Daten direkt in der Adresszeile des Browsers übertragen. Wenn man sich ein Webinterface vorstellt, in dem ein externer Partner des Unternehmens alle seine Kunden anlegt und verwaltet, steht ihm nach dem Login eine Übersicht mit allen für ihn bestimmten Adressen zur Verfügung. Der Link auf eine solche Adresse würde z.B. einen Request wie den folgenden auslösen „http://www.company.com/myaccount.php?kdnr=123456“. Durch einfaches Ändern des angehängten Parameters auf „kdnr=123457“ kann so jedoch eventuell auf eine Adresse zugegriffen werden, die nicht zum Datenstamm gehört. Selbst eine Whitelist, also das positives Sicherheitsmodell, kann hier nicht schützen. Denn in der Whitelist ist definiert, dass im Parameter „kdnr“ eine 6-stellige positive Zahl übertragen werden darf und dies ist der Fall.



Bildupload

Meist handelt es sich hierbei um die fehlende oder fehlerhaft programmierte Logik in der Applikation. Es wird nicht ausreichend geprüft, welcher Nutzer welche Daten aufrufen darf. Eine moderne Web-Application Firewall muss auch solche Angriffe erkennen und absichern können. Dafür müssen die Daten mit dem Backend austauschbar sein oder sie müssen die Response-Daten des Backends einlesen und verarbeiten können. Ähnlich wie beim zuvor beschriebenen Sicherheitsverfahren werden eingegebene Daten und Antworten des Backends im Benutzerkontext auf der WAF analysiert.

Diese Art von Absicherung kann nicht für jeden einzelnen Parameter definiert werden, da der Aufwand viel zu groß wäre. Oft finden Pentester aber genau diese logischen Schwachstellen in Applikationen. Handelt es sich dann um eine Standardanwendung wie z.B. Microsoft, Oracle oder SAP, wird ein Patch sicher Wochen oder sogar Monate auf sich warten lassen. Große Unternehmen haben in diesem Punkt Vorschriften, wie und wann ein Patch veröffentlicht wird. Auch bei kritischen Schwachstellen kann ein Patch sehr lange bis zur Fertigstellung benötigen. Selbst wenn es eine eigene Applikationsentwicklungsabteilung im Unternehmen gibt, kann es dauern, bis diese Schwachstellen behoben werden. Für genau diese Fälle kann eine WAF mit oben beschriebener Funktionalität entsprechend schnell konfiguriert werden und logische Schwachstellen absichern.

Date	Hour	Type	Sub-Type	Source	Target	Details
11/00/2009	22:07:00	Security	SQL	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	21:09:00	Configuration	Backup & Restore	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	21:08:00	Configuration	Backup & Restore	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	21:07:00	Configuration	Backup & Restore	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	20:08:00	Backend	Performances	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	20:07:00	Backend	Performances	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	19:07:00	Resources	Memory	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	17:07:00	Security	SQL	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	17:06:00	Acceleration	Load-Balancing	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine
11/00/2009	10:07:00	Backend	Availability	192.168.0.101	10.1.1.8	Attack blocked by Canonicalization engine

## Bildupload

Unternehmen sollten sich hier auf Experten berufen und von deren Know-how profitieren, um genau diese Schwachstellen absichern zu können.

Julian Totzek-Hallhuber, Technical Consultant Deny All

Foto/Schmuckbild: Quelle-Fotolia.com

Copyright All-About-Security.de / www.all-about-security.de Alle Rechte vorbehalten  
 Vervielfältigung nur mit Genehmigung von All-About-Security.de