

Session Identifikatoren

Einführung

Wenn man sich bei einer Web Anwendung, wie Online Shop oder e-Banking anmeldet, wird auf dem Server eine Session aufgebaut. Darin werden aktuelle Benutzerinformationen wie Inhalte eines Warenkorbes oder den Fortschritt einer Zahlungserfassung abgespeichert. Der Server verknüpft diese Daten mit dem angemeldeten Benutzer. Damit dieser vom Server wieder erkannt wird ist ein Session Identifikator (SessionID) notwendig, welcher bei jeder Anfrage vom Client mitgeschickt wird. Dies wird über die folgenden Technologien bewerkstelligt:

- Cookies
- Hidden Form Fields
- URL Argumente
- SSL Client Zertifikate

Session Identifikatoren bilden ein attraktives Ziel für Hacker. Die SessionID ist dabei, während der Benutzung der Anwendung, die elektronische Identitätskarte des Benutzers. Ein Hacker könnte mit der SessionID die Identität des Benutzers stehlen und missbrauchen.

Diebstahl der SessionID

Bisher waren drei Attackenarten auf SessionIDs allgemein bekannt:

- Interception
Belauschen des Netzwerks oder mittels direkter Attacke z.B. via Cross Site Scripting
- Prediction
Erraten der SessionID

- Brute Force
Durchprobieren von Zeichenkombinationen

Hier möchten wir auf eine weitere Möglichkeit aufmerksam machen:

- Fixation
Vorgängiges Festlegen der SessionID

Session Management

Der allgemeingültige Ablauf beim Session Management von Web Anwendungen beginnt beim Aufruf der Startseite resp. beim Eingeben des Benutzerpasswortes. Der Server generiert zu diesem Zeitpunkt eine SessionID und sendet sie dem Benutzer. Dabei wird sichergestellt, dass die SessionID bei jeder nachfolgenden Anfrage vom Browser des Benutzers zum Server mitgesendet wird. Während ein Benutzer seine Einkäufe im Web Shop tätigt oder seine Zahlungen im e-Banking erfasst, wird nun immer die SessionID im Hintergrund mitgesendet, und so dem Server bei jedem Klick mitgeteilt wer sich auf der Benutzerseite befindet. Die Session endet mit dem Abmelden – der Server löscht die entsprechende Verknüpfung zwischen Benutzer und SessionID und löst somit die Session auf.

Session Fixation

Einführung

Heutige Sicherheitsmassnahmen konzentrieren sich, wie zuvor aufgezeigt auf das Verhindern des Diebstahls einer aktiven und gültigen SessionID. Dabei wird aber ein Sachverhalt vollends ausser Acht gelassen: Die Möglichkeit, dass ein Angreifer beim Server eine SessionID anfordert, diese dann dem Opfer einschleust und wartet, bis das Opfer sich bei der Anwendung anmeldet und somit eine gültige Session aktiviert. Hacker betreiben in diesem Zusammenhang „Social Engineering“ um den Benutzer zu täuschen (z.B. mit einem gefälschten Email). Diese Art von Attacken wird „Session Fixation“ genannt, weil eben die SessionID vorgängig festgelegt

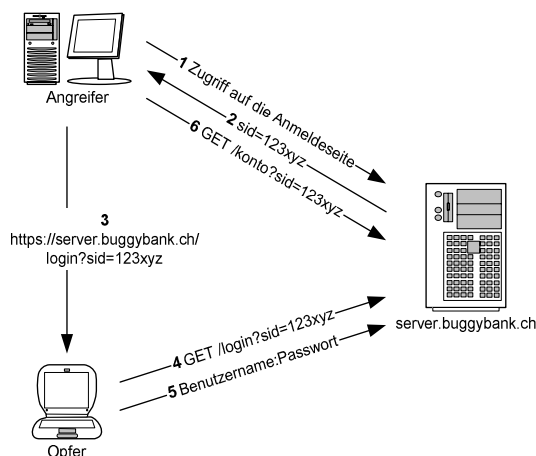
wird, anstatt diese beim Anmelden zufällig zu erzeugen.

Ablauf der Attacke

(Übertragung der SessionID via URL)

Beim folgenden Beispiel handelt es sich um eine e-Banking Anwendung. Ein Benutzer kann sich anmelden um online Zahlungen zu erfassen resp. seine Konten zu pflegen. Die SessionID wird mittels URL Argument übertragen (sid).

Zu Beginn surft der Angreifer auf die Anmeldeseite der e-Banking Anwendung (1). Der Server generiert darauf eine SessionID (2) und sendet diese dem Angreifer zurück. Dieser wiederum sendet ein Email mit dem Link (3) an das Opfer und versucht dabei dieses zu animieren die Anwendung zu benutzen. Nun klickt das Opfer auf den Link (4) im Mail und gibt auf der darauf folgenden Anmeldeseite sein Passwort ein (5). Der e-Banking Server verknüpft nun die SessionID mit dem Benutzer – dem Opfer, welches die Anwendung wie gewohnt benutzen kann. Da aber die SessionID vorher vom Hacker festgelegt worden ist, ist diesem das „goldene Ei“ bekannt und er kann die Anwendung ebenfalls benutzen (6).



Session Fixation Beispiel

Realität

In vergangenen Sicherheitsanalysen konnte Compass Security die im Beispiel beschriebene Funktionsweise beobachten. Häufiger als URL Argumente werden jedoch Cookies als Transportmedium für SessionIDs eingesetzt. Das Prinzip der Attacke bleibt gleich. Dem Angreifer stellt sich aber bei der Einschleusung der (fixierten) SessionID auf das Opfer ein wesentlich grösseres Problem. Cookies können nicht einfach per Mail gesetzt werden. Dennoch stellt sich ein grosses Gefahrenpotenzial dar. Es wird angenommen, dass viele Web Anwendungen betroffen sind. Folgende Schwachstellen müssten zusätzlich vorhanden sein um ein Cookie beim Opfer einschleusen zu können:

- ❑ Ein vorhandener Cross Site Scripting Bug auf einem Webserver in derselben Domäne (z.B. `www.buggybank.ch`). Siehe dazu den Artikel „Cross Site Scripting“ auf der Compass Homepage.
- ❑ Ein verwundbarer Webserver in derselben Domäne (`buggybank.ch`), bei welchem Webseiten verändert werden können. Damit könnten Cookies mittels eines Meta-Tags gesetzt werden.
- ❑ Einfügen des Cookies in der Antwort des Servers (Set-Cookie). Z.b mittels eines kompromittierten Proxies.
- ❑ Fehlerhafte Server Software. Einige Produkte (z.B. JRun) akzeptieren eine wahllose SessionID als URL Argument und senden diese als Cookie zurück.
- ❑ Ein gehackter DNS Server könnte den Benutzer auf den Server des Hackers umleiten, welcher die fixierte SessionID ausliefert.
- ❑ Eine Schwachstelle im eingesetzten Web Browser



Session Fixation Schwachstelle in Web Anwendungen

by Christoph Schnidrig
christoph.schnidrig@csnc.ch

Gegenmassnahmen

Es soll ausdrücklich erwähnt sein, dass der Webserver nicht daran schuld ist, wenn eine Anwendung anfällig auf Session Fixation ist. Jedoch hat der Webserver resp. Applikationsserver dafür zu sorgen, dass die generierten SessionIDs wirklich zufällig sind d.h. nicht durch Erraten oder simples Durchprobieren herausgefunden werden können. Folgende Punkte müssen beim Entwickeln von Web Anwendungen beachtet werden:

- ❑ Alle vom Benutzer gewählten SessionIDs müssen abgewiesen werden. Es muss nach jedem Anmelden eine neue SessionID vom Server erzeugt werden.
- ❑ SessionIDs dürfen erst nach erfolgreichem Anmelden dem Benutzer gesendet werden.
- ❑ Der Benutzer sollte die Möglichkeit haben sich abzumelden. Dabei muss das Cookie im Browser überschrieben und die Session auf dem Server gelöscht werden.
- ❑ Harte Timeouts sollen den Benutzer zwingen von Zeit zu Zeit neu anzumelden. Dies soll das Verweilen in einer gestohlenen Session verhindern.
- ❑ Werden Cookies als Transportmedium eingesetzt muss darauf geachtet werden, dass die Einstellungen des Cookies angepasst werden (Non-Persistent, Secure, Pfad)

Referenzen

Tools

- ❑ @stake's Webproxy
<http://www.atstake.com/webproxy/>
- ❑ Curl
<http://curl.haxx.se/>

Dokumentationen

- ❑ Session Fixation Vulnerability in Web based Applications
<http://www.acros.si/papers.html>

- ❑ Cross Site Scripting
<http://www.csnc.ch/downloads/docs/techdocs/>

Über den Autor

Nach der Informatik TS arbeitete Christoph Schnidrig 3 Jahre als System Engineer bei Comline AG. Anfangs 2001 wechselte er zu Compass Security und nahm die Tätigkeit als Security Analyst auf. Ende 2001 schloss er ein Nachdiplomstudium in Wirtschaft ab.

Compass Security AG

Compass Security Network Computing AG konzentrierte sich seit der Gründung im Februar 1999 durch Walter Sprenger und Ivan Buetler auf Sicherheitsanalysen. Seit dem wurden viele Sicherheitsassessments in der Schweiz wie auch im Ausland durchgeführt.

Nach der Gründung konzentrierte man sich vor allem auf Standard Penetration Tests, welche über das Internet durchgeführt wurden. Dabei setzte man unter anderen automatisierende Tools (ISS, CyberCop, Satan, etc) ein. Währenddessen lernte man die Grenzen dieser Tools kennen. Vor allem bei komplexen Umgebungen konnte man sich nicht auf deren Aussage verlassen.

Darauf begann Compass mit Sicherheitsanalysen von e-Business Applikationen. Dabei wurden die meisten Test manuell durchgeführt. Es interessierten Fragen wie: Kann User A die Daten von User B sehen?

Compass versucht die eingesetzten Assessment Methoden ständig zu verbessern. Schwerpunkt wird auch auf die Schulung sicherheitsrelevanter Sachverhalte gelegt. Aus diesem Grund werden regelmässig Kurse angeboten. Dieses Jahr findet das erste Mal ein Application Security Kurs statt, welcher unter anderem auf die in diesem Artikel behandelten Probleme eingeht.

Weiteres siehe: <http://www.csnc.ch>