Session-IDs im FRITZ!Box Webinterface

Session-IDs und geändertes Login-Verfahren im FRITZ!Box Webinterface

Ab der FRITZ!Box Firmware-Version xx.04.74 wurde der vorhandene Kennwortschutz verbessert sowie ein weiteres Sicherheitsmerkmal für den Zugriff auf die Konfigurationsoberfläche der FRITZ!Box eingeführt, die Session-IDs.
Die Verwendung von Session-IDs bietet einen wirksamen Schutz vor sogenannten Cross-Site Request Forgery-Angriffen, bei denen ein Angreifer unberechtigt Daten in einer Webanwendung verändert. Das folgende Dokument beschäftigt sich mit der Verwendung von Session-IDs und richtet sich an Entwickler, die Tools für die FRITZ!Box programmieren.

Schutz vor Missbrauch

Neben der reinen Verwendung von Session-IDs läuft ein aktiver Schutz gegen mögliche Angriffsversuche. Versucht eine Anwendung ohne oder mit einer ungültigen Session-ID auf die FRITZ!Box zuzugreifen, werden alle aktiven Sitzungen aus Sicherheitsgründen beendet. Ein Zugriff auf die FRITZ!Box ist somit erst nach erneuter Anmeldung möglich.
Greift ein im Hintergrund laufendes Programm wie z. B. ein Anrufmonitor ohne gültige Session-ID, permanent auf die FRITZ!Box zu, beendet dieser Zugriff eine aktive Sitzung. In der Praxis äußert sich das darin, dass die FRITZ!Box regelmäßig ein erneutes Login fordert.
Alle Programme, die auf das FRITZ!Box Webinterface zugreifen, sollten daher Session-IDs unterstützen, da sie sonst nicht nur keinen Zugriff mehr erhalten, sondern wie oben beschrieben auch den normalen Zugriff auf die FRITZ!Box Oberfläche über den Browser beeinträchtigen können.

Session ID-Verfahren

Verwendung der Session-ID

Die Session-ID ist eine 64-Bit-Zahl, die durch 16 Hexziffern dargestellt wird. Sie wird beim Login vergeben und muss für die Dauer der Sitzung mitgeführt werden. Dabei sollte ein Programm zu jeder FRITZ!Box jeweils nur eine Session-ID verwenden, da die Anzahl der Sessions zu einer FRITZ!Box beschränkt ist.
Die Session-ID hat nach Vergabe eine Gültigkeit von 60 Minuten. Die Gültigkeitsdauer verlängert sich automatisch bei aktivem Zugriff auf die FRITZ!Box.
Die Session-ID 0 (0000000000000000) ist immer ungültig.
Die Verwendung der Session-IDs ist ein neues Feature ist, das per Firmware-Update installiert und weiter entwickelt wurde. Abhängig von der Firmware werden Session-IDs also gar nicht oder mit unterschiedliche Verfahren genutzt. Es sollten daher alle Varianten implementiert werden und parallel laufen.

Vergabe der Session-ID

Für die Vergabe der Session-ID wurde eine neue Einstiegsseite eingeführt.

- ab Firmware-Version xx.04.74: login_sid.xml
- ab FRITZ!OS 5.50: login_sid.lua

Auf dieser Seite kann über den Wert ‹iswriteaccess› das Loginverfahren abgefragt werden. Beträgt der Wert 0, muss ein Login mit Kennwort erfolgen. Der dabei angewandte neue Login-Mechanismus wird weiter unten beschrieben.
Beträgt der Wert 1, ist kein Login erforderlich und es wird bei der Anfrage mit ‹SID› gleich eine gültige Session-ID mitgeteilt. Sollte die Seite nicht aufgerufen werden können, handelt es sich um eine Firmware die keine Session-IDs unterstützt.

Übergabe der Session-ID

Die Übergabe der Session-ID erfolgt im Parameter sid (per GET oder POST).
Beispiel HTML-Formular:

```
‹form action="…/cgi-bin/webcm" …›
…
‹input type="hidden" name="sid" value="0000000000000001"›
```

Beispiel für direkten Link:

```
‹A HREF="…/cgi-bin/webcm?getpage=../html/seite.html&
 sid=0000000000000001‹/A›
```

Zugriff ohne Session-ID

Grundsätzlich können alle dynamisch generierten Seiten nur mit einer gültigen Session-ID aufgerufen werden. Auch das Lesen oder Schreiben von Web-Variablen erfordert eine Session-ID. Folgende Inhalte können ohne gültige Session-ID aufgerufen werden:

- Einstiegsseiten (z. B. Login-Seite)
- Statische Inhalte (z. B. Grafiken)

Beenden einer Sitzung

Eine Sitzung kann durch Löschen der Session-ID jederzeit auch vor Ablauf des Timeouts von 60 Minuten beendet werden. Dies geschieht durch POST der Werte

```
sid=‹Session-ID›
```

und

```
security:command/logout=‹dummy›,
```

wobei der Wert für ‹dummy› irrelevant ist.

Verändertes Login-Verfahren

Verwendet eine FRITZ!Box Session-IDs, erfolgt das Login nicht direkt über das Kennwort, sondern über einen Response-Wert, der aus dem Klartextkennwort und einer Challenge ermittelt wird.
Der Login-Vorgang sieht wie folgt aus:

```
‹form action="../cgi-bin/webcm/" …›
…
‹input type="password" name="login:command/response"
value="response"›
‹/form›
```

Im Vergleich dazu der Login-Vorgang, wenn die FRITZ!Box noch keine Session-IDs unterstützt:

```
‹form action="../cgi-bin/webcm/" …›
…
‹input type="password" name="login:command/password"
 value="geheim"›
‹/form›
```

Ermittlung des Response-Wertes

Beim neuen Login-Verfahren wird also das Klartextpasswort

```
login:command/password=‹klartextpassword›
```

ersetzt durch

```
login:command/response=‹response›
```

Der Response-Wert wird wie folgt gebildet:

```
‹response› = ‹challenge›-‹md5›
```

Der Wert ‹challenge› kann aus der Datei login_sid.xml bzw. login_sid.lua ausgelesen werden und ‹md5› ist der MD5 (32 Hexzeichen mit Kleinbuchstaben) von

```
‹challenge›-‹klartextpassword›
```

Der MD5-Hash wird über die Bytefolge der UTF-16LE-Codierung dieses Strings gebildet (ohne BOM und ohne abschließende 0-Bytes).
Aus Kompatibilitätsgründen muss für jedes Zeichen, dessen Unicode Codepoint › 255 ist, die Codierung des "."-Zeichens benutzt werden (0x2e 0x00 in UTF-16LE). Dies betrifft also alle Zeichen, die nicht in ISO-8859-1 dargestellt werden können, z. B. das Euro-Zeichen.
Abschließend ein Beispiel mit deutschem Umlaut:
Die Challenge

‹challenge› = "1234567z"

kombiniert mit dem Kennwort

‹klartextpassword› = "äbc"

ergibt den Wert

‹response› = "1234567z-9e224a41eeefa284df7bb0f26c2913e2"

Besonderheiten ab FRITZ!OS 5.50

Mit FRITZ!OS 5.50 wurden Benutzerrollen eingeführt. Eine Anmeldung erfolgt daher nicht nur mit einem Kennwort, sondern mit einer Kombination aus Benutzername und Kennwort.
Das Luaskript login_sid.lua verhält sich im Wesentlichen wie die alte Einstiegsseite login_sid.xml, bietet aber erweiterte Möglichkeiten.
Der Abruf erfolgt über die URL "fritz.box/login_sid.lua", die Parameterübergabe kann per GET oder per POST erfolgen. Je nach Parameter können über das Skript folgende Aktionen ausgeführt werden:

1) Prüfen, ob eine Anmeldung erforderlich ist und falls nicht, gleich eine gültige Session-ID erhalten
   Parameter: keine
2) Eine Anmeldung durchführen
   Parameter: "username" (Name des Benutzers oder leer), "response" (eine aus Passwort und Challenge erzeugte Response)
3) Prüfung, ob eine bestimmte Session-ID gültig ist
   Parameter: "sid" (die zu prüfende Session-ID)
4) Abmelden, d.h. eine Session-ID ungültig machen
   Parameter: "logout" (Wert beliebig), "sid" (eine gültige Session-ID)

Die Antwort-Xml-Datei hat im Prinzip immer folgendes Format:

```
‹SessionInfo›
   ‹SID›ff88e4d39354992f‹/SID›
   ‹Challenge›ab7190d6‹/Challenge›
   ‹BlockTime›128‹/BlockTime›
   ‹Rights›
      ‹Name›BoxAdmin‹/Name›
      ‹Access›2‹/Access›
      ‹Name›Phone‹/Name›
      ‹/Access›2‹/Access›
      ‹Name›NAS›‹/Name›
      ‹Access›2‹/Access›
   ‹/Rights›
‹/SessionInfo›
```

Die Anzahl und Reihenfolge der Werte im Bereich ‹Rights› ist variabel. Es werden nur die Rechte geliefert, die man hat.

Bedeutung der Werte:

- ‹SID›: Besteht dieser Wert nur aus Nullen, dann hat man keinerlei Rechte, ansonsten enthält er eine gültige Session-ID für den weiteren Zugriff auf die FRITZ!Box. Welche Zugriffsrechte man im einzelnen hat, steht im Bereich ‹Rights›.

- ‹Challenge›: Enthält eine Challenge, mit der man über das Challenge-Response Verfahren eine Anmeldung durchführen kann.
- ‹BlockTime›: Die Zeit in Sekunden, in der kein weiterer Anmeldeversuch zugelassen wird.
- ‹Rights›: Die einzelnen Rechte, die man mit der Session-ID hat.
  Mögliche Werte für die Rechte: 1 für nur lesenden und 2 für Lese- und Schreibzugriff.

Beispiel-Code für den Bezug einer Session-ID Firmware-Version xx.04.74

Verwendung von login_sid.xml

```
using System.Xml.Linq;
using System.Security.Cryptography;

public void Test () {
      string kennwort = "xxxxxxxx";

      // SessionID ermitteln
      string challenge = GetChallenge();
      string response = GetResponse(challenge, kennwort);
      string sid = GetSid(challenge, response);

      // Übersichtsseite der FRITZ!Box einlesen
      string seite =
      SeiteEinlesen(@"http://fritz.box/home/home.lua", sid);
      MessageBox.Show(seite);
}

public string SeiteEinlesen (string url, string sid) {
      Uri uri = new Uri(url + "?sid=" + sid);
      HttpWebRequest request = WebRequest.Create(uri) as HttpWebRequest;
      HttpWebResponse response = request.GetResponse() as HttpWebResponse;
      StreamReader reader = new StreamReader(response.GetResponseStream());
      string str = reader.ReadToEnd();
      return str;
}

public string GetChallenge () {
      XDocument doc = XDocument.Load(@"http://fritz.box/cgi-
           bin/webcm?getpage=../html/login_sid.xml");
      XElement info = doc.FirstNode as XElement;
      return info.Element("Challenge").Value;
}

public string GetResponse (string challenge, string kennwort) {
    return GetMD5Hash(challenge + "-" + kennwort);
}

public string GetSid (string challenge, string response) {
      HttpWebRequest request = WebRequest.Create(@"http://fritz.box/cgi-
         bin/webcm") as HttpWebRequest;
      request.Method = "POST";
      request.ContentType = "application/x-www-form-urlencoded";
      string parameter = String.Format(@"login:command/response={0}-
        {1}&getpage=../html/login_sid.xml", challenge, response);
      byte[] bytes = Encoding.ASCII.GetBytes(parameter);
      request.ContentLength = bytes.Length;
       Stream stream = request.GetRequestStream();
       stream.Write(bytes, 0, bytes.Length);
       stream.Close();
```

```
       HttpWebResponse wr = request.GetResponse() as
HttpWebResponse;
       StreamReader reader = new StreamReader(wr.GetResponseStream());
       string str = reader.ReadToEnd();

       XDocument doc = XDocument.Parse(str);
       XElement info = doc.FirstNode as XElement;
       return info.Element("SID").Value;
}

public string GetMD5Hash (string input) {
      MD5 md5Hasher = MD5.Create();
      byte[] data = d5Hasher.ComputeHash(Encoding.Unicode.GetBytes(input));
      StringBuilder sb = new StringBuilder();
      for (int i = 0; i < data.Length; i++) {
       sb.Append(data[ i ].ToString("x2"));
 }
 return sb.ToString();
}
```

Beispiel-Code für den Bezug einer Session-ID ab FRITZ!OS 5.50

Verwendung von login_sid.lua

```
using System.Net;
using System.Security.Cryptography;
using System.Xml.Linq;

public void Test () {
      string benutzername = "xxxxxxxx";
      string kennwort = "xxxxxxxx";

      // SessionID ermitteln
      string sid = GetSessionId(benutzername, kennwort);
      string seite = SeiteEinlesen(@"http://fritz.box/home/home.lua", sid);
      MessageBox.Show(seite);
}

public string SeiteEinlesen (string url, string sid) {
      Uri uri = new Uri(url + "?sid=" + sid);
      HttpWebRequest request = WebRequest.Create(uri) as HttpWebRequest;
      HttpWebResponse response = request.GetResponse() as HttpWebResponse;
      StreamReader reader = new StreamReader(response.GetResponseStream());
      string str = reader.ReadToEnd();
      return str;
}

public string GetSessionId (string benutzername, string kennwort) {
      XDocument doc = XDocument.Load(@"http://fritz.box/login_sid.lua");
      string sid = GetValue(doc, "SID");
      if (sid == "0000000000000000") {
            string challenge = GetValue(doc, "Challenge");
            string uri = @"http://fritz.box/login_sid.lua?username=" +
benutzername + @"&response=" + GetResponse(challenge, kennwort);
            doc = XDocument.Load(uri);
            sid = GetValue(doc, "SID");
      }
      return sid;
}

public string GetResponse (string challenge, string kennwort) {
      return challenge + "-" + GetMD5Hash(challenge + "-" + kennwort);
}
```

```
public string GetMD5Hash (string input) {
      MD5 md5Hasher = MD5.Create();
      byte[] data =
md5Hasher.ComputeHash(Encoding.Unicode.GetBytes(input));
      StringBuilder sb = new StringBuilder();
      for (int i = 0; i < data.Length; i++) {
            sb.Append(data[i].ToString("x2"));
      }
      return sb.ToString();
}

public string GetValue (XDocument doc, string name) {
      XElement info = doc.FirstNode as XElement;
      return info.Element(name).Value;
```