Anmeldung am FRITZ!Box Webinterface

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

Die Anmeldung an einer FRITZ!Box kann grundsätzlich auf drei Arten erfolgen:

- Mit Benutzernamen und Kennwort
- Nur mit Kennwort (Auslieferungszustand)
- Ohne Kennwort (nicht empfohlen)

Seit FRITZ!OS 5.50 ist in allen drei Fällen zusätzlich eine Session-ID erforderlich.

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 erhalten, sondern wie oben beschrieben auch den normalen Zugriff auf die FRITZ!Box Oberfläche über den Browser beeinträchtigen können.

Die Einstiegsseite

Informationen zum genutzten Login-Verfahren sowie die Vergabe der Session-ID erfolgt ab FRITZ!OS 5.50 über die Einstiegsseite

```
http://fritz.box/login_sid.lua
```

Hinweis: Sollte die Seite nicht aufgerufen werden können, handelt es sich um eine alte Firmware, die keine Session-IDs unterstützt.

Je nach übergebenen Parameter können 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 Kennwort 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", "sid" (eine gültige Session-ID)

Die Antwort-XML-Datei hat immer den gleichen Aufbau:

```
<SessionInfo>
	<SID>8b4994376ab804ca</SID>
	<Challenge>a0fa42bb</Challenge>
	<BlockTime>0</BlockTime>
	<Rights>
		<Name>NAS</Name>
		<Access>2</Access>
		<Name>App</Name>
		<Access>2</Access>
		<Name>HomeAuto</Name>
		<Access>2</Access>
		<Name>BoxAdmin</Name>
		<Access>2</Access>
		<Name>Phone</Name>
		<Access>2</Access>
	</Rights>
</SessionInfo>
```

Die Anzahl und Reihenfolge der Werte im Bereich ist variabel. Es werden nur die Rechte geliefert, die die aktuelle Session-ID auch tatsächlich hat.

Bedeutung der Werte:

- <SID>: Besteht dieser Wert nur aus Nullen, bestehen keinerlei Rechte. Eine Anmeldung ist erforderlich.
  Ansonsten enthält er eine gültige Session-ID für den weiteren Zugriff auf die FRITZ!Box. Die aktuellen Zugriffsrechte stehen im Bereich <Rights>.
- <Challenge>: Enthält eine Challenge, mit der über das Challenge-Response-Verfahren eine Anmeldung durchgeführt werden kann.
- <BlockTime>: Zeit in Sekunden, in der kein weiterer Anmeldeversuch zugelassen wird.
- <Rights>: Die einzelnen Rechte, die die aktuelle Session-ID hat. Mögliche Werte sind 1 (nur lesen) und 2 (Lese- und Schreibzugriff).

Anmeldung

Ist die "Anmeldung ohne Kennwort" in der FRITZ!Box eingestellt, wird beim Aufruf der Einstiegsseite bereits eine gültige Session-ID übergeben. Sie können in diesem Fall direkt zum Abschnitt "Verwenden der Session-ID" springen.

Ist eine Anmeldung erforderlich, muss zunächst im Rahmen des Logins eine gültige Session-ID erzeugt werden. Aus Sicherheitsgründen erfolgt das Login nicht mit dem Klartextkennwort, sondern über ein Challenge-Response-Verfahren.

Ermittlung des Response-Wertes

Der Response-Wert wird aus dem Klartextkennwort und einer Challenge wir folgt ermittelt:

```
<response>=<challenge>-<md5>
```

<challenge> wird aus der Einstiegsseite login_sid.lua ausgelesen.

<md5> ist der MD5 (<challenge>-<klartextpassowort>) in 32 Hexzeichen mit Kleinbuchstaben

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.

Beispiel mit deutschem Umlaut:

Die Challenge

```
<challenge> = "1234567z"
```

kombiniert mit dem Kennwort

```
<klartextpasswort> = "äbc"
```

ergibt den Wert

```
<response> = "1234567z-9e224a41eeefa284df7bb0f26c2913e2"
```

Der Login-Vorgang sieht in diesem Fall wie folgt aus:

```
http://fritz.box/login_sid.lua?
   response=1234567z-9e224a41eeefa284df7bb0f26c2913e2
```

Die Antwort-XML beinhaltet nun die gültige SID.

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 20 Minuten. Die Gültigkeitsdauer verlängert sich automatisch bei aktivem Zugriff auf die FRITZ!Box.

Die Session-ID 0 (0000000000000000) ist immer ungültig.

Die Übergabe der Session-ID erfolgt im Parameter "sid".

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 20 Minuten beendet werden.

Dies geschieht durch Aufruf der Login-Seite mit dem Parameter "logout":

```
http://fritz.box/login_sid.lua?logout=1&sid=<sid>
```

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

Beispiel-Code .NET

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

namespace FritzLogin
{
  class LoginData
  {
    public LoginData(string scheme, string host, String userName, string
password, string path)
    {
      Scheme = scheme;
      Host = host;
      UserName = userName;
      Password = password;
      Path = path;
      Url = string.Format("{0}://{1}/{2}", Scheme, Host, Path);
    }

    public string Scheme { get; set; }
    public string Host { get; set; }
    public string UserName { get; set; }
    public string Password { get; set; }
    public string Path { get; set; }
    public string Url { get; private set; }
  }

  class Program
  {
    static void Main(string[] args)
    {
      var loginData = new LoginData("http", "fritz.box", "username",
"password", "login_sid.lua");
      Test(loginData); Console.ReadKey();
    }

    static public void Test(LoginData loginData)
    {
      Console.WriteLine("Url: " + loginData.Url);
      // determine sessionID
      string sid = GetSessionId(loginData);
      Console.WriteLine("SID: " + sid);
      string page = ReadPage(loginData.Url, sid);
      Console.WriteLine("Page content:" + System.Environment.NewLine +
page.ToString());
    }
```

```
    static public string ReadPage(string url, string sid)
    {
      url = String.Format("{0}?sid={1}", url, sid);
      HttpWebRequest request = WebRequest.Create(url) as HttpWebRequest;
      HttpWebResponse response = request.GetResponse() as HttpWebResponse;
      StreamReader reader = new StreamReader(response.GetResponseStream());
      return reader.ReadToEnd();
    }

    static public string GetSessionId(LoginData loginData)
    {
      XDocument doc = XDocument.Load(loginData.Url);
      // Brute-Force-Protection.
      string szBlocktime = GetValue(doc, "BlockTime");
      int blockTime = Int32.Parse(szBlocktime);
      if (blockTime > 0)
      {
        Console.WriteLine("waiting {0}sec. ...", blockTime);
        System.Threading.Thread.Sleep(blockTime * 1000);
      }
      string sid = GetValue(doc, "SID");
      if (sid == "0000000000000000")
      {
        string challenge = GetValue(doc, "Challenge");
        string url = String.Format("{0}?username={1}&response={2}",
loginData.Url, loginData.UserName, GetResponse(challenge,
loginData.Password));
        doc = XDocument.Load(url);
        sid = GetValue(doc, "SID");
      }
      return sid;
    }

    static public string GetResponse(string challenge, string password)
    {
      return String.Format("{0}-{1}", challenge, GetMD5Hash(challenge + "-"
+ password));
    }

    static public string GetMD5Hash(string input)
    {
      MD5 md5Hasher = MD5.Create();
      // UTF-8 > UTF-16LE
      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();
    }

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

Beispiel-Code PHP

```
<?php
```

```
$fritz_host = 'fritz.box';
$fritz_url = 'http://' . $fritz_host . '/login_sid.lua';
$fritz_user = 'username';     //optional
$fritz_pwd = 'password';
$with_debug_output = false;


// Get Challenge-String
$tmp = simplexml_load_string(file_get_contents($fritz_url));
if ($tmp->BlockTime > 0) {
      sleep($tmp->BlockTime);
      $tmp = simplexml_load_string(file_get_contents($fritz_url));
}
if ($with_debug_output) {
      print_r('Got challenge: ' . $tmp->asXML() . PHP_EOL);
}
$challenge = $tmp->Challenge;

// Get SID
$challenge_str = $challenge . '-' . $fritz_pwd;
$md_str = md5(iconv("UTF-8", "UTF-16LE",  $challenge_str));
$response = $challenge . '-' . $md_str;
$tmp = simplexml_load_string(file_get_contents($fritz_url . '?user=' .
$fritz_user . '&response=' . $response));
if ($with_debug_output) {
      print_r(' Got session ID: ' . $tmp->asXML() . PHP_EOL);
}
$sid = $tmp->SID;

// Logout
$tmp = simplexml_load_string(file_get_contents($fritz_url . '?logout=1&sid='
. $sid));
if ($with_debug_output) {
      print_r(' Logout: ' . $tmp->asXML() . PHP_EOL);
}
?>
```