CGI 技术及其安全性研究

道 焰 （广东工业大学电子信息系 510643）
朱世伟 （华中理工大学计算机系 430074）

摘要：本文论述了 CGI 技术的标准规范、实用范围，并给出了一 CGI 的应用实例，最后就如何编写安全 CGI 程序进行了讨论。

关键词：CGI WWW 标准规范 安全

一、CGI 标准规范

公共网关界面 CGI 是 WWW 服务器在调用外部进程或外部可执行程序（CGI 原本）时的参数规范协议，它规定了一组标准的环境变量和参数格式。CGI 原本根据 WWW 服务器设置的环境变量和传递参数采取相应的动作后生成响应结果，通过协议规定的格式返回 WWW 服务器或浏览器，CGI 原本和 WWW 之间的数据通信主要包括以下四个方面：

1. **环境变量**。在调用 CGI 程序前，服务器在特定的环境变量中放入信息，CGI 程序可以从这些环境中读取所需的内容。主要的环境变量如下：

SERVER-SOFTWARE

运行网关程序的软件名及版本号。格式为：名字/版本

SERVER-NAME

当用 URL（Uniform Resource Location）引用网关程序时，需指定的服务器主机名、域名服务器别名或 IP 地址。

GATEWAY-INERFACE

服务器执行的 CGI 修订版本号。格式为：CGI/修订版本号

SERVER-PROTOCOL

查询信息时遵循的信息传输协议及版本。格式为：协议名/版本

SERVER-PORT

查询信息被送往的端口号。

REQUEST-METHOD

询问方式，对于 HTTP 有"GET"，"HEAD"，"POST"等几种方式。

PATH-INFO

由客户给出的附加路径信息。它放在访问 CGI 原本时的实际路径的尾部。如果该信息来源于 URL，那么在它送往 CGI 原本前就能被服务器解译出。例如在服务器上有一个 CGI 程序，/cgi－bin/foobar，它能处理常驻在服务器文档根目录（Documentroot）里的文件。当需要告诉 foobar 去处理哪个文件时，由包含在 URL 尾端的外部路径，通过环境变量 PATH-INFO，foobar 将知道文档相对于文档根目录的位置，或者通过 SERVER 为你产生环境变量 PATH-TRANSACTED 知道文档的绝对路径。

QUERY-STRING

该变量定义为 URL 中的第一个后的信息串。该信息串可以使用 ISINDEX 加进，也可以使用 HTML 语言元素 FORM（利用 GET 法）加入，这一串通常是一信息查询串。

SCRIPT-NAME

当用 URL 引用 CGI 原本时所指的路径及 CGI 原本名。

REMOTE-HOST

发出查询的主机名，如果服务器找不到该变量则找 REMOTE-ADDR 变量。

REMOTE-ADDR

发出查询的远程主机的 IP 地址。

CONTENT-LENGTH

客户给出询问内容的长度。

2. CGI **命令行参数**。命令行仅仅用在 ISINDEX 查询中，它不能用于 HTML form 或其他未定义的查询类型。服务器通过查找询问信息（环境变量）中是否有非编码字符"＝"来决定是否调用命令行，如果发现一个"＝"，则不调用命令行。

3. CGI **原本输入**。CGI 有两种方法获取来自客户机的数据，一种是 GET，一种是 POST。如果查询方法为 GET，则应该从环境变量 QUERY-STRING 中读取数据。如果是 POST 或 PUT 方式，则应该从标准输入 STDIN 中读取，此时数据将不以 EOF 为结束标志，而是从 CONTENT-LENGTH 中取出数据长度。

4. CGI **原本输出**。CGI 原本输出有以下两种方法：

(1)数据返回给服务器。CGI 程序按照 HTTP 协议要求，将 CGI 头域和信息实体从标准输出送给服务器，服务器除了计算信息实体的长度外，还要加入一般性头域，然后形成给客户机的应答，连同 CGI 输出的信息实体一起返回给客户机。

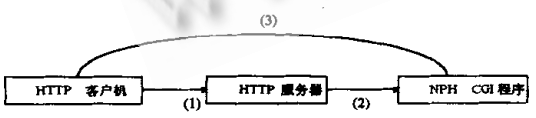
①表示客户机与 HTTP 服务器连接，并提出请求。

②表示服务器一些环境变量赋值后，运行 CGI 原本。

③CGI 原本完成处理后将结果返回服务器。

④HTTP 将应答返回给客户机。

(2)数据返回给客户机。文件名以 Nph－为前缀的 CGI 程序称为 NPH 程序(Non－Parset Header)。它从标准输出中直接将数据送给客户机，而不需要从中做任何处理工作，是一种非常特殊的 CGI 程序。对于普通的 CGI 程序，服务器只有收到 CGI 程序的全部执行完毕后的所有输出，才能响应客户机，针对客户机一次请求有且仅有一次应答，而 NPH 程序却可以控制对用户应答的次数，且不必等到程序执行完毕之后，这样就极大地增强了 CGI 程序的功能，同时 NPH 程序直接将数据返回给客户机，不需要服务器作中介，提高了服务器和整个系统的效率。



①表示客户机与服务器连接后，提出请求。

②表示服务器为一些环境变量赋值后，运行 NPH 程序，将全部输出的控制权交给 NPH 程序。

③表示 NPH 程序完成处理后直接将结果返回给客户机。

二、CGI 技术的应用

CGI 程序可以进行各种处理，例如可用来建立查找程序和可点图像，还可以作为数据库的界面或转到其他协议的网关等等。随着 WWW＋数据库已成为当前网络发展的流行趋势，CGI 技术的应用范围也越来越广泛。

例：一综合数据查询，可用于查找各种数据。

·创建一 HTML 文本，用来指定 CGI 提交方式即激活 CGI 程序，主体部分如下：

```
< FORM ACTION = " cgi - bin/search. exe?" METHOD = "POST" >
    < INPUT TYPE = "text" NAME = "Match" >
    < INPUT TYPE = "submit" VALUE = "确认" >
    < INPUT TYPE = "reset" VALUE = "清除" >
    </FORM>
```

·编写 CGI 程序，编译成可执行文件(如 search. exe)。程序的主体部分如下：

```
sub CGI-MAIN()
Dim sel As String
Dim Test As String
Dim FNum, Fd As Integer
sel = LCase $ (Mid $ (CGI-LogicalPath. 2))
StartDocument(sel)
FNum = FreeFile
Fd = 0
If Len(CGI-FormTuples(2). Value) = 0 Then
    Send("搜寻失败!")
    Exit Sub
End If
Open "num. dat" For Input As
Do While Num EOF(FNum)
If Input > 0 Then
  If Fd = 0 Then
    Send("< H3 >")
    Send("你需要的结果是" &CGI-FormTuples(0). Value)
    Send("</H3>")
    Send("< HR >")
  End If
  Send("< H4 >< I > = >" &TestString & "</H4>")
```

```
    Fd = Fd + 1
  End If
  Loop
  If Fd = 0 Then
    Send("<H3>")
    Send("你需要的结果是" &CGI-FormTuples(0).
Value)
    Send("CHR")
    Send("抱歉! 找不到此数据")
    Send("</H3>")
  End If
  Close (FNum)
  End Sub
```

三、编写安全的 CGI 程序

因为 CGI 程序是可执行的,这就意味着全世界的人均可以在你的系统上运行这个程序,一般 CGI 程序极容易成为黑客袭击的目标,这些黑客(hacker)企图通过袭击 CGI 程序以获得非法权限,因此当你在编写 CGI 程序时必须了解一些安全性问题。

1. 一般将 CGI 原本放置于服务器的 cgi - bin 这一特殊目录下,通过加密的方法仅仅允许 Web 管理员安装这些原本,这样就避免了黑客利用 CGI 原本的潜在漏洞侵入系统的根文件目录。

2. 注意赋值语句,例如 PERL 和 Bourne 外壳语言提供了一个 eval 命令,这个命令允许你构造一个串,同时允许解释器能执行该串。这样潜在着一定的危险。请看下列 Bourne 外壳语句:

```
eval 'echo $QUERY-STRING | awk 'BEGIN {RS
="&"}{printf "QS - %s \ n, $1}'' 一个精明的黑客
```

可能将该查询串变成一变量集的集合命令,这样只需送一以;开头的查询串就能偷袭该原本。

3. 不要完全相信客户,一个有良好行为的客户在他的查询串中应避免使用那些在外壳语言中有特殊意义的字符以免原本错误的解释了这些字符。而一些别有用心的客户可能用这些特殊的字符破坏你的原本而获取非法权限。

4. 由于黑客常常通过更改环境变量指使 CGI 原本去执行他所想执行的文件,所以你在调用外部程序时除了要验明用户身份外,还应用绝对路径来指定程序,而不是依赖环境变量。

5. 千万不要通过非验明身份的远程用户输入的外壳命令。这些命令包括 C 语言中的 popen()和 system(),各种外壳系统中 exec 和 eval 命令。如下例,用户发送一个指明在 fill - out form 中的地址:

```
$ mail-to = &get-name-form-input; read the address
from form
open(MAIL,"|/usr/lib/sendmail $ mail-to");
print MAIL "To: $ mailto \ nFrom: mo \ n \ nHi
there! \ n";
close MAIL;
```

问题出在 open()的调用上,编者假设赋给变量 $ mail-to 的永远是一有效的地址,但是如果黑客送入如下的地址:

```
nobody@ nowhere. com; mail badquys @ hell. orq </
etc/passwd;
```

那么 open()将执行下面的命令:

```
/usr/lib/sendmail nobody @ nowhere. com; mail
badquys@hell.orq</etc/passwd;
```

无意中将系统的 PASSWORD 文件泄漏给了远程用户。

5. 当你在网上下载其他站点的 CGI 原本到你的 Web 服务器上时,你应该从以下几方面仔细地检查这些 CGI 原本以确保你系统的安全。

·该程序是否很长,一般程序越长问题越多。

·它在主机上是否有读、写文件的操作,读程序有可能破坏你已经建立起来的种种限制,将系统信息泄漏给黑客,而写程序有可能改变甚而破坏你的文件目录。

·它与你系统上的其他程序有联系吗? 许多 CGI 程序为了响应 form 中的输入建立与 sendmail 程序的联系发出 e - mail,这样做存在一些安全漏洞。

·程序中是否有 suid(set - use - id)权限? 一般这样作是极危险的,除非有充足的理由。

·程序的作者是否确认用户输入,从这一点可以看出原作者是否考虑了安全问题。

参考文献

[1] HTML 用户使用指南 [美] N. 兰达尔 科学出版社

[2] 计算机世界报 计算机世界报出版社 (10 - 13)

[3] INTERNET 最新实用技术及其应用 孙辫华 沈正华 柳纯录 北京大学出版社

(来稿时间:1997 年 6 月)