Meta-Characters

کاراکترهایی مانند * ، ; ، | و ` در برنامه های کاربردی و اسکریپت ها معنی مخصوصی دارند. این کاراکترها به هیچ عنوانی روی URL تاثیری ندارند اما ممکن است در پایان باعث شوند که نحوه اجرای برنامه کاربردی تغییر پیدا کند و معنی ورودی ها را برای برنامه کاربردی عوض کند و بعضی مواقع باعث ایجاد سوراخهای امنیتی برای اینگونه برنامه ها می شود!

بیشتر Meta-Character ها به صورتهای مختلف در سرورهای مختلف تفسیر می شوند. اگر بخواهیم به صورت واضح تری بیان کنیم **Meta-Character ها کاراکترهای قابل نمایش و یا غیر قابل نمایشی هستند که روی رفتار دستورات زبانهای برنامه نویسی اثر می گذارند** به خصوص در برنامه ها و یا Query های ساخته شده برای یک پایگاه داده. البته بسته به نوع سیستم عامل، زبان برنامه نویسی و نوع پایگاه داده معانی اینگونه کاراکترها تغییر پیدا می کند.

جدول ۱ انواع Meta-character ها و تفسیر آنها را بیان می کند.

جدول شماره ۱

نشانه	توضیحات
;	برای توضیحات اضافی در یک زبان استفده می شود.
|	کاراکتر Pipe برای دستورات اجرایی استفاده می شود.
&	استفاده در دستورات اجرایی
%20	به عنوان یک فضای خالی در URLها و دیگر نام ها به کار می رود.
%00	کاراکتر تهی[1] برای انتهای یک رشته و یا انتهای نام یک فایل به کار می رود.
%04	انتهای یک فایل برای آلوده کردن پایان فایل ها.
%0a	ایجاد خط جدید برای نوشتن دستورات جدید اجرایی. برای آلوده کردن میلها و محتوای فایل ها
%0d	ایجاد خط جدید برای نوشتن دستورات جدید اجرایی. برای آلوده کردن میلها و محتوای فایل ها
‘	در بعضی از Query های پایگاه داده استفاده می شود. و از کاراکترهایی می باشد که در یکی از زیباترین تکنیک های هک کردن وب به نام SQL Injection به کار می رود.
“	مانند کاراکتر ‘ می باشد.
-	در ترکیبات بعضی از Query های پایگاه داده استفاده می شود.
* %	در ترکیبات بعضی از Query های پایگاه داده استفاده می شود.
/	برای آلوده کردن مسیر ها و بعضی از Query های پایگاه داده استفاده می شود.
\	برای آلوده کردن مسیر ها و بعضی از Query های پایگاه داده استفاده می شود.
< >	برای نوشتن بعضی از زبانهای اسکریپتی داخل یک فایل درون یک سرور وب استفاده می شود!!
? $ @ : {([])}	بسته به نوع زبان برنامه نویسی معانی مختلفی دارند.

[1] -Nullbytes

مشخص کردن کاراکترهای مخصوص در یک رشته URL

سوالی که حالا به وجود می آید این است که چگونه می توان از کاراکترهایی مانند % ، ? ، و یا + بدون آنکه معنی مخصوصی بدهند استفاده کرد؟ برای مثال اگر بخواهیم پارامترهای book=pride&prejudice و shipping=snailmail را به وسیله URL به یک برنامه ارجاع دهیم به صورت زیر می شود:

http://mycheapbookshop.com/purchase.cgi?book=pride&predjudice&shipping=snailmail

که نتیجه آن یک URL دو پهلو می باشد زیرا دو نشانه & در آن قرار دارد و خیلی شبیه URL ای می باشد که می خواهد سه پارامتر را ارجاع دهد، book=pride ، prejudice= و shipping=snailmail !!

اگر می بخواهیم که نشانه & را به عنوان جزیی از ارزش یک پارامتر ارجاع دهیم می توانیم آن را به وسیله کاراکتر مخصوصی که در دو رقم هگزادسیمال می باشد ، بیان کرد که به آن شکل رمزگذاری اسکی می گویند که همیشه با پیشوند % می باشد مانند زیر:

– همه کاراکترهای رمز شده هگز	%XX (%00 - %FF)
– کاراکترهای کنترلی	%00 - %1F - %7F
– کاراکترهای بالای ۸ بیت	%80 - %FF
– کاراکتر جای خالی	%20 یا +
Carriage return –	%0d
Line feed –	%0a

در مثال قبل ، ارزش اسکی نشانه & ، ۳۸ می باشد که ۲۶ هگزا دسیمال می شود. بنابراین اگر می بخواهیم از کارکتر & به عنوان ارزش یک پارامتر استفاده کنیم باید از کد 26% استفاده کنیم.

http://mycheapbookshop.com/purchase.cgi?book=pride%26prejudice&shipping=snailmail

سواستفاده از روزهای URL

معماری پروتکل Http باعث شد که رمزگذاری URL ایجاد شود تا بعضی از کاراکترهای غیر الفبایی در رشته های URL به صورت کاراکترهای الفبایی در بیایند تا بتوان از آنها استفاده کرد. بعضی مواقع سرورهای وب به وسیله روشهای استفاده غیر متعارف از کاراکترهای رمز شده در رشته های URL فریب می خورند. دو تا از مهمترین آسیب پذیری های سرورهای وب که گزارش شده است در رمزگشایی URL ها می باشد !

آسیب پذیریهای Unicode

در اکتبر سال ۲۰۰۰ یک آسیب پذیری به نام "unicode bug" کشف شد که به موجب آن با استفاده از خطایی که در رمزگشایی کاراکتر / به وجود می آمد به یک کاربر اجازه می داد که از دایرکتوری

اسناد وب به دایرکتوری سیستم ویندوز پرش کرده و با استفاده از فراخوانی cmd.exe کنترل کل سیستم را به دست بگیرد! به عنوان مثال به URL زیر توجه کنید:

http://192.168.7.21/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+d:\

این URL باعث می شود که لیست کلیه فایلها و دایرکتوریها درون درایو D در مرورگر شما نمایش داده شود!

اما این حمله چگونه کار می کند؟ %c0%af یک Unicode غیر مجاز می باشد که در اصل کاراکتر / را نشان می دهد. این URL باعث می شود که سرور وب رشته %c0%af را به / تعبیر کند و باعث بارگذاری آن به فیلتر کننده سرور وب شود و موجب می شود که به دو دایرکتوری قبل از دایرکتوری /Scripts/ برود (یعنی به صورت معمول دایرکتوری ریشه!) و بعد از آن به /winnt/system32/cmd.exe دسترسی پیدا می کند! دایرکتوری /Scripts/ معمولا در موقعیت C:\inetpub\Scripts قرار دارد. در شرایط معمولی، سرورهای وب اجازه نمی دهند که یک URL به موقعیتی خارج از دایرکتوریهای اسناد وب دسترسی پیدا کند(در این مورد C:\inetpub) . بنابراین وقتی که سرور کاراکتر / را تشخیص می دهد خود به خود آن را در نظر نمی گیرد. اما /..%c0%af.. به / .. / .. تعبیر می شود و همین امر باعث می شود که سرور وب URL بالا را به صورت زیر ترجمه کند :

http://192.168.7.21/scripts/../../winnt/system32/cmd.exe?/c+dir+d:\

که در اصل یعنی اجرای خط فرمان cmd.exe به وسیله آن :

http://192.168.7.21/winnt/system32/cmd.exe

اما در اینجا حتما این سوال به ذهن شما رسیده است که چگونه %c0%af به / تعبیر می شود؟

در ابتدا باید این را بیان کنیم که چگونه یک Unicode نامشروع ساخته می شود.

کد اسکی کاراکتر / به هگزادسیمال **2F** می باشد که به صورت باینری **00101111** می شود. رمز کننده Unicode و یا واضح تر UTF-8 کاراکترهای بالاتر از **256** و یا آنهایی که از ۸ بیت بیشتر می شوند را مجاز تشخیص می دهد. راه صحیح نمایش **2F** در UTF-8 همان شکل **2F** می باشد. اگر چه نمایش **2F** را می توان به صورت چندین نوع نمایش UTF-8 نشان داد. کاراکتر / را می توان به صورت یک بایتی ، دوبایتی و یا حتی سه بایتی نشان داد که درشکلهای رمز شده UTF-8 در زیر نمایش داده شده اند:

Used "/"			**Binary**		**Decimal**	**Hex**
1 byte	0xxxxxxx		00101111		47	2F
2 byte	110xxxxx	10xxxxx	11000000	10101111	49327	C0 AF
3 byte	1110xxxx	10xxxxxx 10xxxxxx	11100000	10000000 10101111	14713007	E0 80 AF

که در اصل x ها همان الگوی کاراکتر رمز شده از راست به چپ می باشند. بنابراین نمایش کاراکتر / به صورت دو بایتی در UTF-8 همان **C0 AF** می باشد که در URL ما به صورت دو کارکتر رمز شده %c0%af نشان داده شده است! همان حمله حتی در استفاده از رمزگذاری UTF-8 سه بایتی کار می کند URL زیر نشان دهنده شکل نمایش URL این حمله می باشد:

http://192.168.7.21/scripts/..%e0%80%af../winnt/system32/cmd.exe?/c+dir+d:\

آسیب پذیری رمزگشایی دوباره[2] و یا رمزگشایی زاید[3]

وقتی که مایکروسافت در صدد به دست آوردن حیثیت دوباره خود در آسیب پذیری به وجود آمده ، بود ؛ آسیب پذیری دیگری در می سال ۲۰۰۱ کشف شد! که به **رمزگشایی دوباره** و **رمزگشایی اضافه** معروف شده بود. و از همین جا درگیریهای آنها با آسیب پذیریهای به وجود آمده شروع شد.

از آسیب پذیری رمزگشایی دوباره می توان به صورت URL ای به شکل زیر بهره برداری کرد:

http://192.168.7.21/scripts/..%25%32%66../winnt/system32/cmd.exe?/c+dir+d:\

کاراکتر / به جای رشته %25%32%66 جایگزین می شود. نمایش URL در رمزگشایی اول به صورت زیر می شود:

http://192.168.7.21/scripts/..%2F../winnt/system32/cmd.exe?/c+dir+d:\

%25 = “ % “
%32 = “ 2 “
%66 = “ F “

اگر URL فوق یک بار دیگر رمزگشایی شود به صورت زیر می شود:

http://192.168.7.21/scripts/../../winnt/system32/cmd.exe?/c+dir+d:\

در جدول زیر ترکیبات بیشتری از اینگونه رشته ها را می بینید که امکان ترجمه دیگری از خود را دارند:

Encoded Pattern	Hex Representation	ASCII Character
%25%32%63	%5c	“ \ “
%25%35f	%2f	“ / “
%252f	%2f	“ / “
%252F	%2F	“ / “
%255C	%5C	“ \ “

تهیه کننده:

امیر حسین شریفی ، websecuritymgz@websecuritymgz.com

٤ آبان ماه ۱۳۸۲

منبع:

Web Hacking , Stuart McClure,Saumil Shah , Shreeraj Shah -۱

[2] -Double-Decode
[3] -Superfluous